

EON Script 中文使用手冊



瑋特擬真科技股份有限公司
Vimtek Virtual Reality Tech Co.,Ltd.

www.vimtek.com.tw

Tel:07-3958202 Fax:07-3958203

版權所有：瑋特擬真科技股份有限公司

Script 節點介紹.....	4
程式腳本撰寫入門.....	5
在 Script 節點中新增屬性欄位.....	10
開始建立程式腳本.....	17
使用 Script 編輯器.....	19
如何引用欄位內的數值.....	24
EON 的程式編輯語法.....	32
EON 的控制語法列表.....	33
MakeSFVec2f 控制語法.....	35
MakeSFVec3f 控制語法.....	36
MakeSFColor 控制語法.....	37
Find 控制語法.....	38
Find Node 控制語法.....	40
GetNodeName 控制語法.....	41
GetNodePath Method 控制語法.....	42
GetScriptTimeout 控制語法.....	43
SetScriptTimeout 控制語法.....	44
MessageBox 控制語法.....	45
SaveSnapShot 控制語法.....	46
GetParentNode 控制語法.....	47
GetFieldCount 控制語法.....	48
GetField 控制語法.....	49
GetFieldByName 控制語法.....	50
Value 讀取方式.....	51
GetName 控制語法.....	52
GetType 控制語法.....	53

目錄

GetMFCount 控制語法.....	55
SetMFCount 控制語法.....	56
GetMFElement 控制語法.....	57
SetMFElement 控制語法.....	58
AddMFElement 控制語法.....	60
RemoveMFElement 控制語法.....	62
Initialize 初始化.....	64
EventsProcessed 事件處理.....	65
Shutdown 模擬視窗關閉語法.....	66

Script 節點介紹

Script 節點是 EON 所有節點中，功能及用途最多的節點。此節點允許在 EON 中，使用 VBScript 或 JScript 撰寫出符合特殊需求的節點。VBScript 及 JScript 在安裝 EON 程式的同時便已包含在其中。在 EON 中使用 Script 撰寫出來的節點，只會受到開發者的創意而有所限制。而在 Script 節點內所撰寫的程式，允許使用者創造出 EON 基本功能節點所無法設定的功能，並且透過 Script 節點撰寫的程式，不但可以簡化 EON 功能連結設定縮短製作的時間，並且可以發揮更加強大的功能。

由於 Script 節點的自由度相當的大，因此使用者將可撰寫出許多在流程視窗中，所無法設定的功能(例如判斷、陣列等...)。且使用 Script 程式腳本來執行一些重複的功能或動作時，將可縮減節點功能與流程視窗的設定時間，加快整個專案的製作進度。

本操作手冊將會說明在 EON 中，如何建立 Script 節點的屬性欄位及程式腳本的撰寫方式、程式編輯視窗的使用方式，及複數資料欄位引用與編修，本手冊中將可給予初學者各種不同的指導方針，例如程式撰寫流程、VB 與 Java 程式語法的差異，以及除錯與疑難排解的提示等相關訊息。

如果想使用 Script 節點，必須具備 VB 語法或 Java 語法的基礎。如果對 VB 或 Java 語法不熟悉，或者想要學習較深入的程式，可以上網搜尋 JScript.chm 或 VBScript.chm 關鍵字，查詢程式語法的撰寫技術。

本章節將會透過一個簡易的實作範例，介紹如何開始使用 script 節點撰寫程式腳本。

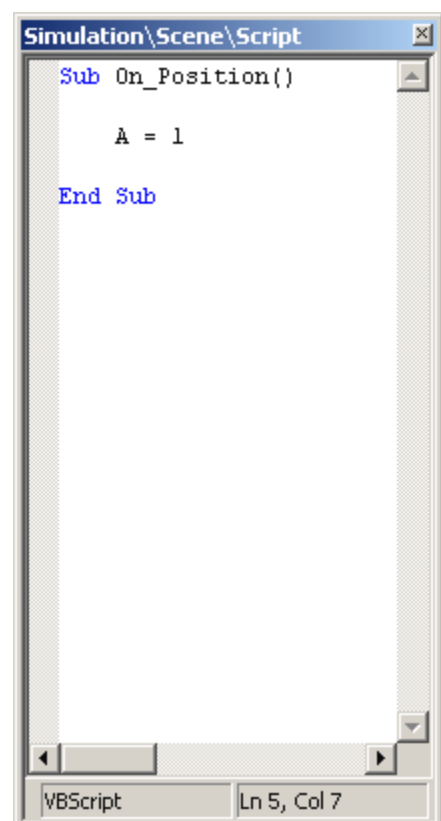
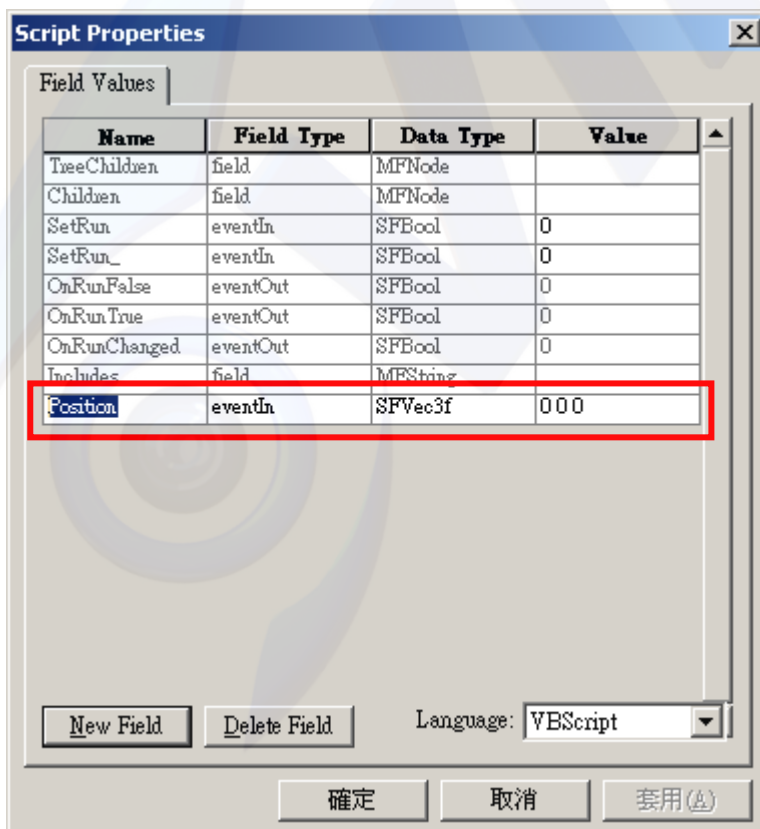
何謂程式腳本？

所謂的程式腳本，簡單的來說可分成以下的三個步驟：

1. 啟動程式腳本
2. 運算或處理變數的數值
3. 輸出新的欄位數值

1. 啟動程式腳本

正常的程式腳本執行流程，script 節點必需要有一個屬性欄位，用來接收其它節點輸出的事件訊號，作為觸發程式腳本執行的開關。因此開始撰寫程式腳本前，必需在 Script 節點屬性設定視窗中，新增一個欄位作為觸發腳本運行的事件訊號。並且撰寫程式腳本時，必需在程式腳本的開頭以 **On_新增欄位名稱()** 的語法撰寫，之後，節點收事件訊號時，才知道要執行哪一段程式，例如：以 VB 的語法在 Script 節點中新增 **Position** 的欄位時，程式腳本的開端必需寫成語法為 **Sub On_Position()**。只要是節點接到的事件訊號有變化時，節點便可執行相關的程式腳本。



2. 運算或處理變數的數值

接收到觸發腳本運行的事件訊號後，我們可使用熟悉的程式語法，去撰寫出需要進行運算或處理的程式腳本。以 VB 語法為例：

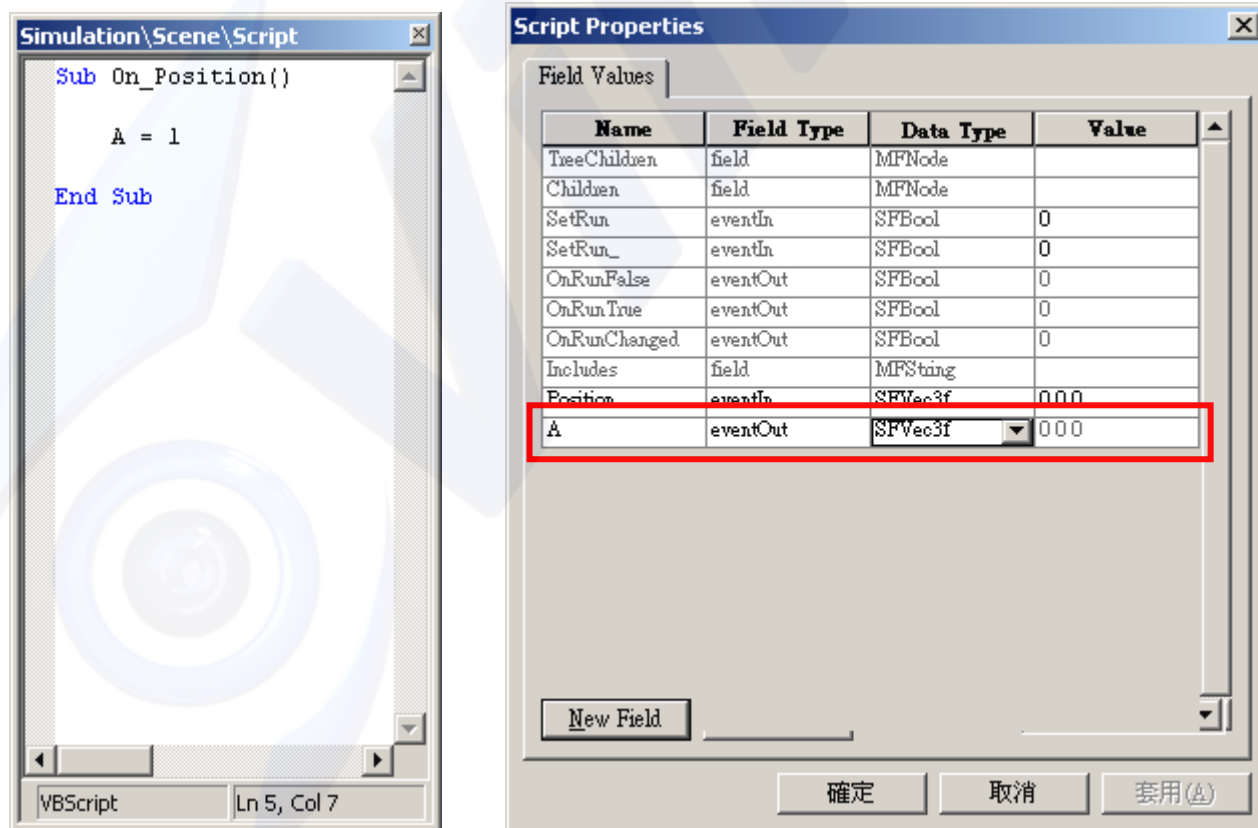
```
NrClicks = NrClicks + 1
```

```
if NrClicks > 10 then mystring = "Please stop clicking here."
```

3. 輸出新的欄位數值

經過程式腳本**運算或處理**後，通常會產生一些新的數值，因此爲了將這些新的數值傳給其它的節點，必需在 Script 節點新增屬性欄位，用來傳送這些新的數值。而這些新欄位名稱必須與程式腳本裡的新變數名相同，欄位類型則可選擇 **outevent** 或 **exposedfield**，如此設定 Script 節點才能將新的數據，透過屬性欄位傳遞給其他的節點。

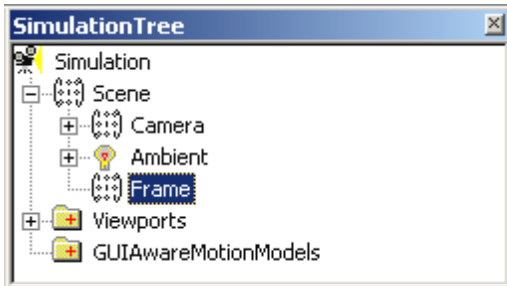
以 VB 語法為例：程式腳本中，有一個變數 **A**，而變數 **A** 的數值等於 **1**。爲了將這個數值傳送給其它的節點，因此在 Script 的屬性設定視窗中，我們新增了一個名稱爲 **A** 的欄位，並將 Fields Type(欄位類型)設爲 **outevent**、Data Type(資料類型)設爲 **SFVec3f**。(欄位類型與資料類型的說明將會在下個章節提到。)



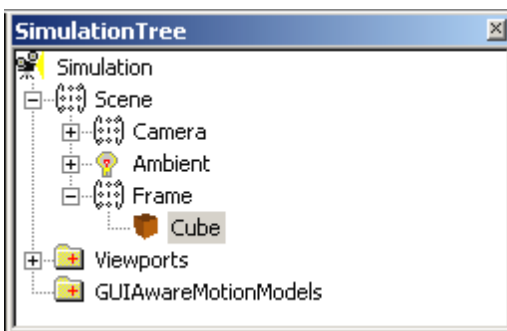
程式腳本撰寫實作練習

以下範例將會引導進行一個快速的程式腳本撰寫實作練習。本練習內容是設定按鍵 [1] 按下時，模擬視窗內的物件將會消失不見。而按下按鍵 [2] 時，消失的物件會再出現。

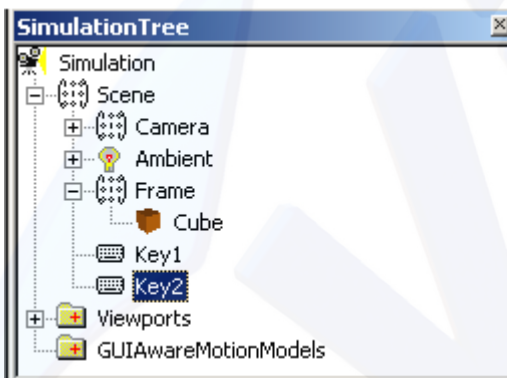
1. 新增一個 Frame 節點到 Scene 底下。



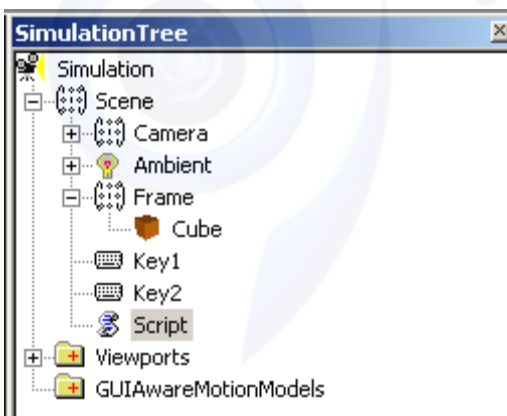
2. 再新增的 Frame 節點底下新增一個 Cube 原型



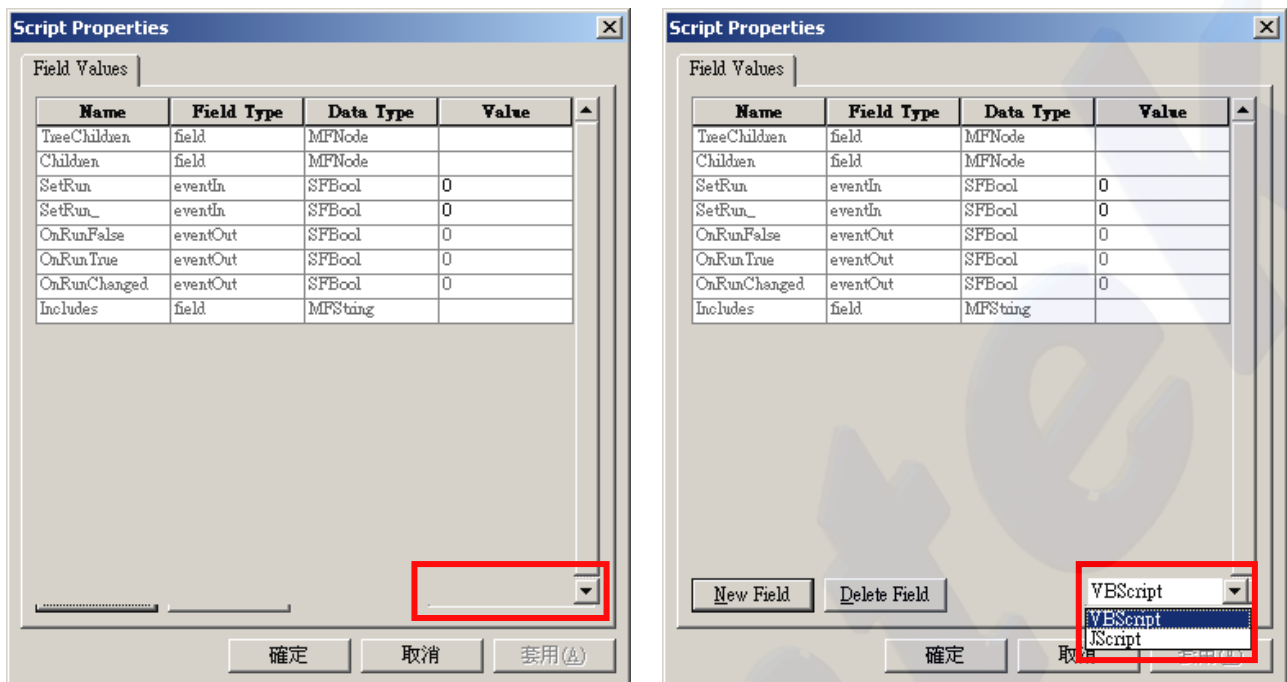
3. 新增二個 keyboard sensors 節點到 Scene 底下，新增節點後將名稱改為 Key1 與 Key2，並設定按鍵 [1]、[2] 作為節點感應的按鍵。



4. 拖曳一個 Script 節點到 Scene 節點底下，並於新增節點後開啓節點的屬性設定視窗。

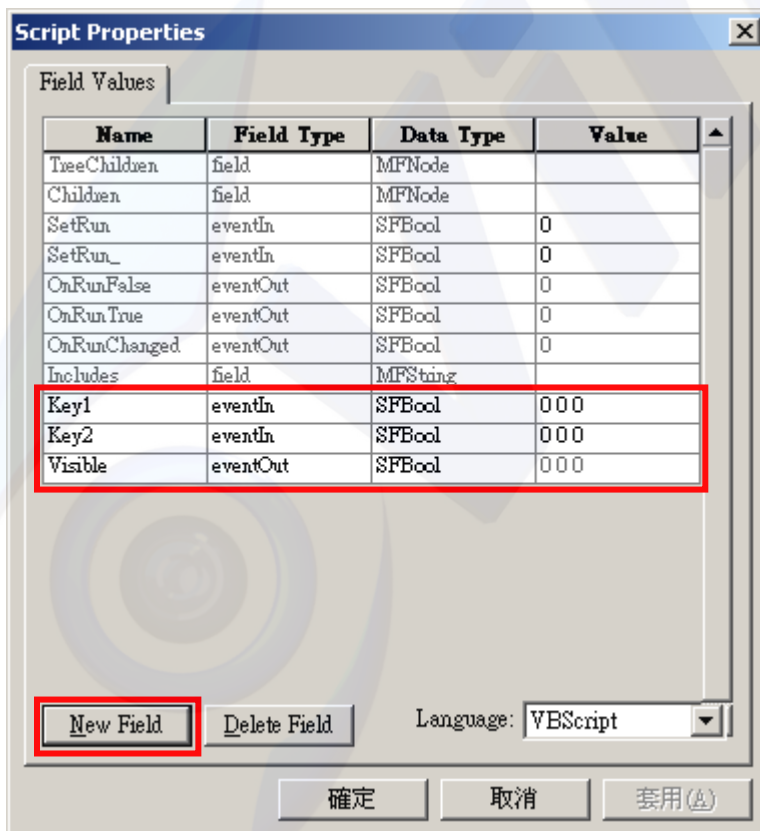


5. 開啟 Script 節點的屬性設定視窗後，先點選視窗右下程式語法的下拉式選單，將程式語法改為 VBScript。(Script 節點預設語法為 JavaScript)

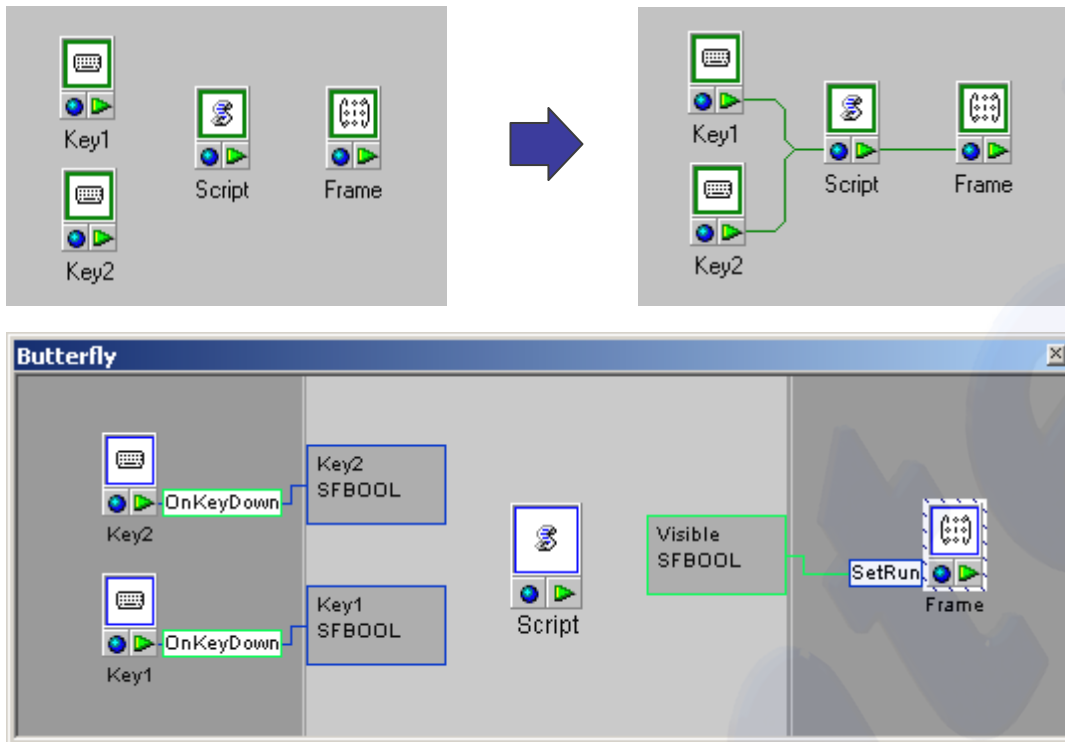


※由於我們使用的系統為中文編碼，因此視窗下方的選單有被蓋掉的情況，但仍可正常點選。

6. 依照下圖的設定，點取左下方的 New Field(新增屬性欄)按鈕，新增三個屬性欄位。



7. 將二個 keyboard sensors 節點與 Frame、Script 節點拖曳到流程設定視窗中，並依以下的訊號連結方式，將此四個節點連結起來。



8. 先點選 Scrip 節點後，再點擊視窗上方的程式撰寫視窗工具鈕  開啓程式編輯視窗。並將視窗內的程式碼，改爲以下的程式：

Sub On_Key1()

If Key1 Then Visible = False

End Sub

Sub On_Key2()

If Key2 Then Visible = True

End Sub

```
// Edit Scriptnode here.
// Add subroutines for each in-event or exposed field
// that you define in the script node's property page.

function initialize()
{
    // Called when simulation starts
    // Add initialization code
}

function On_Key1()
{
    // TODO: Add your event handler code here
}

function On_Key2()
{
    // TODO:
}

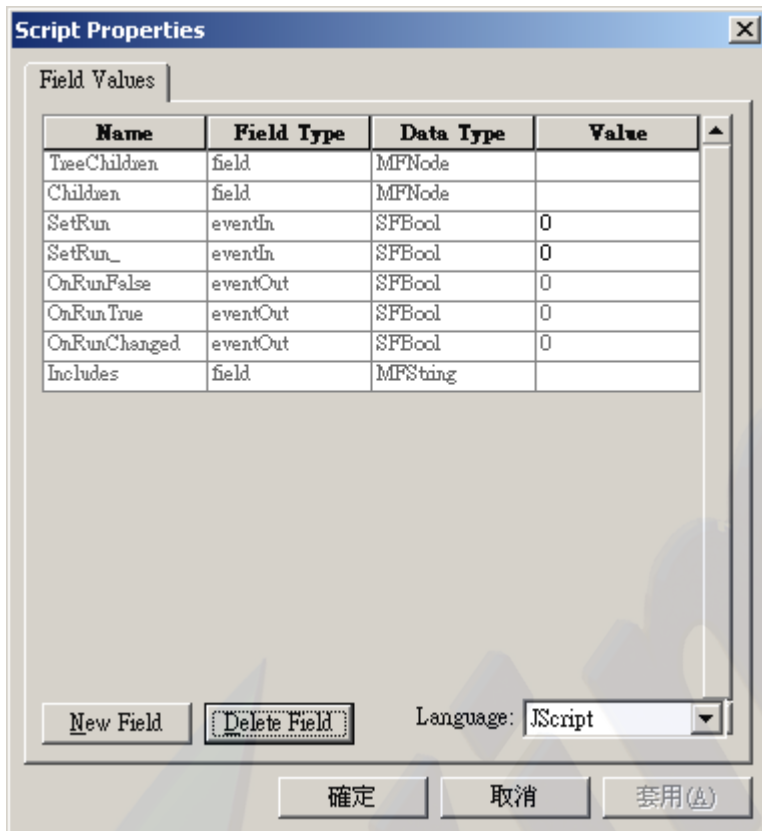
Sub On_Key1()
    If Key1 Then Visible = False
End Sub

Sub On_Key2()
    If Key2 Then Visible = True
End Sub
```

9. 修改好程式碼後，即可開啓模擬視窗測試。

在 Script 節點中新增屬性欄位

建立一個新的 Script 節點後，首先要進行的第一個步驟，在屬性設定視窗中新增一個屬性欄位。但我們在撰寫程式腳本時，依照腳本內容的不同，有時我們也會在撰寫的途中，新增一些相關的屬性欄位。因此在本章節將會向你介紹在撰寫程式時，如何設定節點屬性視窗中的 **Name**、**Field type**、**Data type** 與 **Value** 這四個屬性欄位，與欄位的性質。



※每個 EON 的節點裡，都有隱藏的預設欄位，而這些欄位分別是 TreeChildren、Children、SetRun、SetRun_、OnRunFalse、OnRunTrue、OnRunChanged。同樣的 Script 節點也有這些預設欄位，而這些欄位的 Name、Field Types、Data Types 都是無法刪除或更動的。

四個屬性欄位的性質說明

Names(名稱)欄

此欄位可以讓使用者自行定義屬性欄的名稱。欄位內可以任意輸入字元，但是有以下幾點需要注意的：**欄位的第一個字元必需為英文字母，不可為阿拉伯數字；欄位雖然可以輸入中文字元，但 EON 的程式會無法辨別，因此中文欄位將無法觸發程式腳本。**

新增屬性欄位時，系統會自動將新增的欄位命名為 **NewField**。如果一次新增數個欄位時，新增第二個之後欄位的名稱，會自動依序定義成 **NewField1**、**NewField2**...**NewFieldN**，同樣的這些新增欄位的名稱字可以自行更改。

Field Type(欄位類型)欄

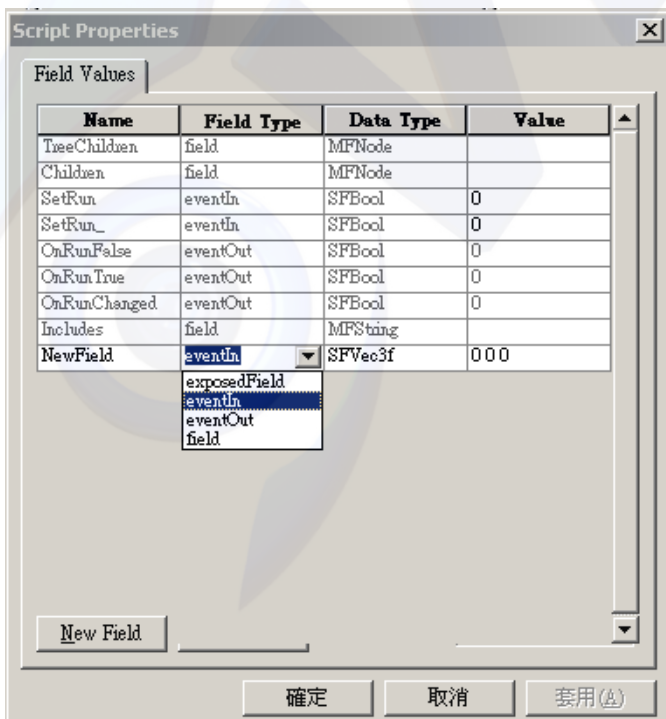
此欄位共有以下四種型態屬性，使用者可以依照不同的腳本需求設定。

ExposedField：欄位的類型設為 **ExposedField**，則表示此欄位具有可以接收和傳送欄位事件的功能。

EventIn：欄位的類型設為 **EventIn**，則表示此欄位只能接收其它節點傳來的事件，無法將欄位內的事件傳遞出去。

EventOut：欄位的類型設為 **EventOut**，則表示此欄位只能將欄位內的事件傳遞出去，且無法從其它的節點接收事件。

Field：欄位的類型設為 **Field**，則此一欄位內的數值，只是用來儲存或顯示腳本運行時的欄位事件值。無法從別的節點接收事件，也無法將此欄位內的事件輸出。



Data Type(資料類型)欄

EON 的 Script 平台接受所有標準的資料型態。只要在新增節點欄位時，在 Data Type 這一行欄位的下拉選單中，選擇表單裡的型態名稱即可改變它的資料型態。

Script 節點的資料型態請參照下表。在 EON 的 Script 節點中，資料型態可分為 SF 開頭的單一資料型態的欄位，以及開頭為 MF 含有多個 SF 型態資料的多重資料欄位。以 SFBooly 布林值的資料型態為例，一個欄位可以儲存一個 true 或 false 的變數數值，而使 MFBool 則可以儲存多個一連串的 true 或 false 的變數數值。

型態名稱	欄位型態	性質描述	數值範圍
整數	SFInt32 MFInt32	此欄位內的數值為十進位的整數型態的數值。	[-2, 147, 483, 648, 2, 147, 483, 647]
浮點數	SFFloat MFFloat	此欄位為的數值為帶有小數點的浮點數。	[-3.4E-38, 3.4E 38]
影像	SFImage MFImage	此欄為可以應用在材質貼圖上的彩色影像圖檔檔案連結。	
文字	SFString MFString	此欄位為文字字串性質的參數。	N/A
2 維浮點數	SFVec2f MFVec2f	此欄位為二維座標性質的參數。	[-3.4E-38, 3.4E 38]
3 維浮點數	SFVec3f MFVec3f	此欄位為三維座標性質的參數。	[-3.4E-38, 3.4E 38]
布林值	SFBool MFBool	此欄位為布林性質的參數。 可以顯示 TRUE 或 FALSE。	TRUE/FALSE
節點連結	SFNode MFNode	EON Script 特有的數值型態，可以使 Script 節點直接抓取或套用數值在節點上。	
色彩參數	SFColor MFColor	此欄位為 RGB(紅綠藍)3 原色的浮點數值參數。	0-1
時間參數	SFTime MFTime	此欄位為以秒為單位的倍精準型浮點參數。計時的範圍為一天總秒數。	
旋轉參數	SFRotation MFRotation	4 個浮點參數：此欄位的前三個浮點數為圍著三個座標軸旋轉的角度，第四個浮點數為旋轉時間。在執行時(x、y、z)三個參數的變量將會一致。	

※撰寫 Script 程式時，SFImage(單一影像)類型參數，是絕對不會使用到的。而且在 EON 的所有節點裡，也沒有任何節點有這種類型的參數欄位。

新增 Script 節點的屬性欄位時，如何設定適當的欄位屬性是非常重要的的一件事。要使 Script 節點在 Routes：Simulation(流程與模擬)視窗中與其它節點連接起來，必須將 Datatype 欄位的類型，設定成與接收或輸出事件訊號欄位相同的類型，因為在 Routes：Simulation(流程與模擬)視窗中不同 Datatype 類型的數值訊號是無法相互連接的。

※補充說明：新增一個新的 Datatype 類型為 SFNode 欄位後，在 Script 節點底下同時也會多出一個與欄位名稱相同的黃色的資料夾。如果新增 Datatype 類型為 MFNode 的欄位，則在 Script 節點底下同樣的，也會多一個跟欄位名稱相同的黃色的資料夾，不過資料夾的圖示上，會多出一個紅色的+號，表示可以一次放入數個節點連結。

由於 VBScript 的程式語法，不支援動態排序陣列數值的功能。因此在撰寫程式時，我們可以使用 MF 類型的欄位來儲存陣列內的數值，然後再透過程式的運算來排序這些欄位內的值，這也算是一個變通的方式。

Values(數值)欄

在 Script 屬性欄內的資料或數值，必須符合欄位 Datatype(資料類型)。因為在撰寫一段程式時，是不可能強迫變數去接收不同類型的輸入訊號或數值。然而我們可以在 Field Type(資料型態)為 Field 或 ExposedField 時，在 Values 欄位裡輸入不同型態的數值，並將其設為初始值，但這並非正常的作法，而且也沒有任何意義。

為了將要數值輸入到 Script 節點的屬性設定視窗或屬性列的欄位裡，必需知道以下所列的一些數值輸入技巧。例如：

要輸入像 SFVec3f 或者 SFCOLOR 這種有數個數值類型的欄位時，需要利用空格將不同的數值分開。例如：要輸入 (3, -10, 1.5) 立體座標參數，則在欄位中輸入 **3 -10 1.5**。此外，如果是要在像 MFVec3f 或者 MFCOLOR 這種 MF(多重資料資料)型態的欄位時，同樣也是使用空格將所有的參數隔開。例如：要輸入 (3, -10, 1.5)、(2, 0, 2)、(1, 10, 0.5) 這三組立體座標參數時，則在欄位中輸入 **3 -10 1.5 2 0 2 1 10 0.5**。

關於字串類型的資料欄位，在字串的前後加上雙引號 " 加以註記，例如："您好"。如要在 **MFString** 這種多行文字型態的欄位裡輸入資料時，則可在不同行的文字字串的前後各加上一個雙引號 " 加以註記，並且在各字串間以空格隔開。例如："你好嗎" "我很好"。

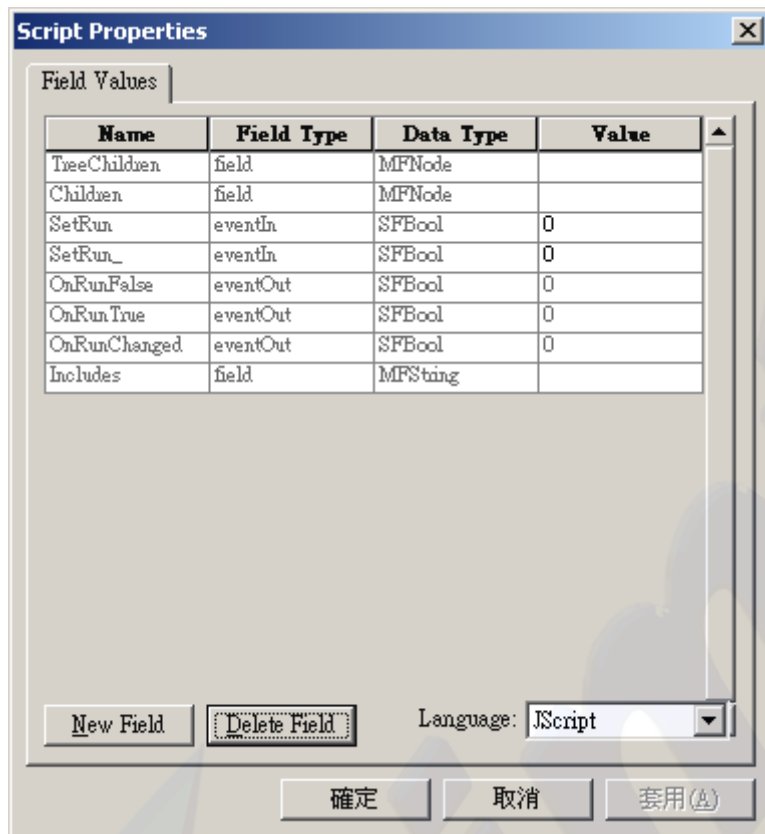
而布林數值類型的資料欄位，則是可以使用數字 **0** 表示 **False** 值，數字 **1** 表示 **True** 值，如要在 **MFBool** 這種多個布林數值的多重資料資料欄位時，則與多重資料座標參數一樣也是使用空格將所有的參數隔開。例如：要輸入 **True**、**True**、**False**、**False**、**True**、**False**、**True** 這樣一系列布林值時，則在欄位內輸入 1 1 0 0 1 0 1。

而 **SFNode** 類型的參數，則是以輸入字串的方式，將節點在模擬樹視窗中的路徑位置輸入到欄位中。例如：要在欄位中輸入 **Camera** 節點的路徑，則在欄位中輸入"Scene\Camera"。如要在 **MFNode** 欄位內輸入多個節點的路徑時，也跟多行文字一樣在不同的節點路徑加上一個雙引號 " 加以註記，並且在各字串間以空格隔開。例如："Scene\Frame" "Scene\Frame1"。

模擬視窗執行時，可開啓 **Script** 節點的屬性設定視窗，變更欄位內的數值。或直接修改屬性列視窗中欄位內的數值。並且在欄位內的數值修改後，**Script** 節點也會將改變後的新數值，帶入到程式裡進行運算。但是有部分欄位的數值，在模擬視窗關閉後，會回復到模擬視窗尚未執行前的狀態。

新增屬性欄位

當開啓一個新 Script 節點的屬性設定視窗時，可以看在屬性欄位裡有 TreeChildren、Children、SetRun、SetRun_、OnRunFalse、OnRunTrue、跟 OnRunChanged 這幾個灰色的預設屬性設定欄位，而這些欄位的名稱、欄位類型與資料類型皆無法變更與修改。如下圖：



新增一個新的屬性欄

在開啓節點的屬性設定視窗後，可以點選視窗中的 **New Field** 按鈕，或者按下鍵盤上的 **[Alt] + N** 鍵，以新增一行新的屬性設定欄位。新增欄位名稱將會被自動命名為 **NewField**，而第二新增的欄位會被命名為 **NewField1**，以此類推。而這些新增的屬性欄位必須依照程式腳本裡的變數重新命名。只需先選要變更名稱的欄位，再點擊一次滑鼠左鍵或按下 **[F2]** 功能鍵，即可重新命名該欄位。

在變更屬性欄位的名稱後，你可以點選 **Field Type** 欄位的下拉式選單，並依照程式腳本的需求，選擇適當的欄位類型。同樣的，此步驟也適用於 **Data Type** 屬性的選擇。最後在於 **Value** 欄位中，輸入一個預設的參數值。

※提示：如果想整理屬性設定視窗內的欄位，可以選取要調整的欄位後，用 **Ctrl**+ **C** 的方式複製選取的欄位，貼到 **Excel** 或記事本上，然後在 **Excel** 或記事本裡調整欄位的順序。待調整好欄位的順序後，複製調整好的欄位資料，並且用 **Ctrl**+ **V** 的方式貼上複製的資料，即可將屬性列欄位重新排列。有條列的屬性欄位，將有助於在撰寫程式時，可以更加明確的找到所需要引用的變數。

編輯欄位

如果需要改變一個欄位的 **Name** (名字)、**Field Type** (欄位類型) 或者 **Data Ttype** (資料類型) 時，不必刪除整個屬性欄位或是再建立一個新屬性欄位。只需直接修改欄位的欄位的 **Name** (名字)、**Field Type** (欄位類型) 或者 **Data Ttype** (資料類型) 即可。如果發現有相同名稱的屬性欄位時，那麼請變更其中一個欄位的名稱，因為節點是不允許有相同名稱的欄位，而且節點會自動除去排序在後名稱相同的欄位。

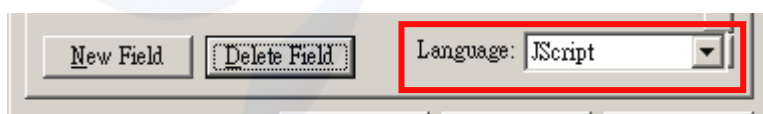
此外有一點是必須注意的，只要修改了 **Name** (名字)、**Field Type** (欄位類型) 或者 **Data Ttype** (資料類型) 這些欄位中的任一項設定後，會造成原本在 **Routes：Simulation**(流程與模擬) 視窗中，與變更欄位相關的連結設定全部消失。

刪除屬性欄位

如果要刪除屬性欄位，只需先用遊標選取要刪除的欄位，然後再點選 **Delete Field** 鈕，即可將選取的欄位刪除。

變更程式語法

目前 **Script** 節點預設的程式語法為 **Jscript**，且節點只支援 **VBScript** 和 **Jscript** 這二種語法。如果想要切換節點的語法，那麼你可以在屬性設定視窗的右下方的下拉式選單中，選擇想要使用的語法。



※提示：由於節點預設的語法為 **Jscript**，因此在程式編輯視窗中，有一些 **Jscript** 格式的註解，與數值初始化的語法，如果是使用 **VBScript** 語法，請將程式編輯視窗中的內容刪除。

開始建立程式腳本

程式腳本是由一組程式碼所建立起來的，通常使用 **Procedures**、**Functions** 與 **handlers** 這些名字來稱呼它。在 **VBScript** 的語法裡，程式腳本是由關鍵字「**Sub**」，作為程式腳本的開頭，並且以關鍵字「**End Sub**」結束。而在 **Jscript** 的語法裡，則是使用「**functions**」這個關鍵字作為程式腳本的開頭。以下為此二種語法的撰寫範例：

VBScript

```
sub On_Myfield()  
    statements  
end sub
```

JScript

```
function On_Myfield()  
{  
    statements  
}
```

Myfield：表示 Script 節點欄位的名稱。

Statements：表示撰寫的程式

以欄位類型為 **EventIn** 與 **ExposedField** 的欄位來說，把欄位的名稱，以 **On_ fieldname()** 的語法，寫在程式腳本的開頭時，則表示只要該欄位內的數值有任何的變動，就會觸發節點執行其底下的程式碼。即使該欄位接收到的數值相同，也會觸發節點執行程式。

開啓程式編輯視窗前，只要新增任何的欄位，EON 都會在程式編輯視窗中，同時新增一個該欄位程式腳本的開頭與結尾。如下圖：

Includes	field	MFSting	
NewField	exposedField	SFVec3f	000
NewField1	eventIn	SFVec3f	000

```
function On_NewField()  
{  
    // TODO: Add your event handler code here  
}
```

也可以自行建立一個程式腳本的開頭與結尾，不過要使用正確的語法格式，在程式腳本開頭的關鍵字之後，加入「**On_欄位名稱()**」的固定語法，如此當節點的欄位接收到訊號時，才知道要執行那一段程式腳本。

※進階提示：使用者可以為每個程式腳本，寫具有資料類型和輸出事件格式的副程式，這些副程式所產生出來的值，也可以透過節點的欄位輸出給其它的節點，但是 EON 是絕不會自動創這些副程式，必需自己撰寫。

節點預設欄位的值也可以在程式腳本中使用

撰寫程式腳本的內容時，Script 節點預設的接收欄位數值如 **SetRun**、**SetRun_**，撰寫時，通常會被用來作為觸發開、關功能程式的腳本裡。這意味著當 Script 節點被關閉時，可能不會有任何的訊號被送出。但是在實際操作下，會發現其它欄位在接收到訊號時，仍可正常運作執行程式，並不會受到節點本身 **SetRun**、**SetRun_** 的影響，而中斷或關閉程式的運行。

如果要使節點在接收到訊號時停止程式腳本的執行，建議建使用名稱為 **IsActive**、欄位類型為 **exposedfield** 資料類型為 **SFBool** 的欄位。**IsActive** 欄位也可以撰寫在初始狀態的程式碼裡，設定 Script 節點開或關的狀態。例如：

```
sub On_SetRun()
    IsActive = SetRun
    IsNotActive = not SetRun
end sub
sub On_SetRun_()
    SetRun = not SetRun_
end sub
sub On_IsActive()
    SetRun = IsActive
end sub
sub On_IsNotActive()
    SetRun = not IsNotActive
end sub
sub On_PosIn()
    if IsNotActive then exit sub
    p = PosIn
    if p(1)>-5 then p(1) = -5
    PosOut = p
end sub
```

※提示：到當你接收 **SetRun** 和 **SetRun_** 的輸入型訊號時，**OnRunTrue**、**OnRunFalse** 和 **OnRunChanged** 等輸出型訊號的參數值，將會自動改變訊號的數值。因此在 **SetRun** 和 **SetRun_** 的程式腳本中，**OnRunTrue = true**、**OnRunTrue= true**、**OnRunFals= true** 這些欄位值的變化程式可以略過不撰寫。

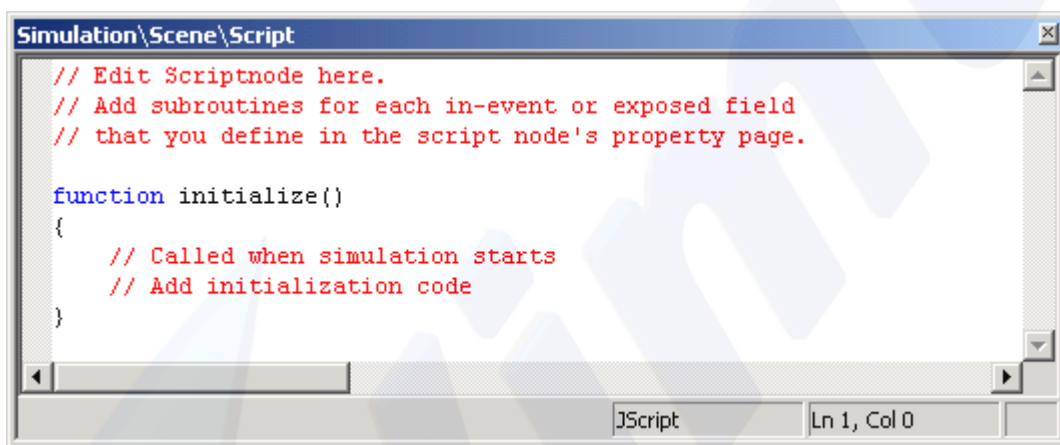
使用 Script 編輯器

開啓 Script 編輯器

當一個 Script 節點被新增到模擬樹視窗後，節點的內容是一片空白，因此節點要執行程式腳本，必須透過一個類似記事本的編輯器，撰寫節點將要執行程式腳本。需先選取要編輯 Script 節點，再透過以下列任一種方式，即可開啓 Script 編輯器，進行程式腳本撰寫。

- 點選視窗上方工具列中的  圖示
- 按下鍵盤上的 Ctrl + E
- 按下鍵盤上的 Alt + 6
- 或點選視窗上方的 Window 功能列表選擇 Window -> Script -> Open Editor

當第一次打開 Script 編輯器時，可以看到在編輯器裡的前幾行，有寫上一些提示文字。而且節點欄位初始化的腳本也會裡面。如下圖：



當模擬視窗開啓時，可以開啓 Script 編輯器，並且修改裡面的程式腳本。修改後的腳本並不會在已開啓的模擬視窗中執行，必需重新開啓模擬視窗，才會執行修改後的程式腳本。

如果要編輯數個不同 Script 節點的程式腳本時，可以在同一時間裡開啓各個 Script 節點的 Script 編輯器進行編輯。如果要切換各個不同的 Script 編輯器，可以直點選視窗下方的編輯器分頁，或者點選視窗上方的 Window 功能列表，選擇要編輯的 Script 編輯器進行程式腳本編輯。

在 Script 編輯器撰寫程式

文字縮排

爲了使程式腳本易於閱讀與編修，因此建議在撰寫程式時，使用文字縮排的功能將各行程式分段排版，在每一行程式碼前，使用鍵盤上的 **空白鍵** 或 **Tab** 鍵，進行文字縮排的功能。若想取消縮排，只需按下 **倒退鍵(Backspace)** 或 **Shift** + **Tab** 鍵，即可取消縮排。

如果想一次在多行程式碼間加入縮排，一次選取要進行縮排的程式碼，然後再按下 **Tab** 鍵即可。如果要一次取消多行程式的縮排，一樣的先選取要取消縮排的程式碼，然後按下 **Shift** + **Tab** 鍵即可取消。

快捷鍵

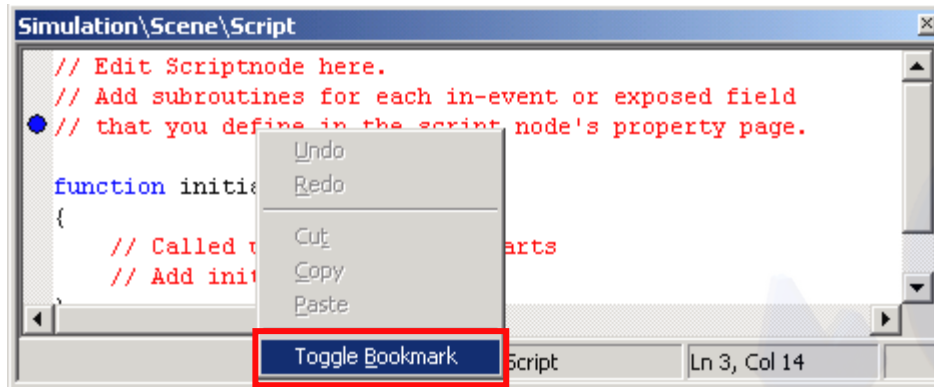
Script 編輯器的快捷鍵與一般文字編輯器所使用的快捷鍵相同，並且具有 **Undo(回復)**與 **Redo(重做)**的功能，以及快速搜尋及替代的功能。

快捷鍵列表

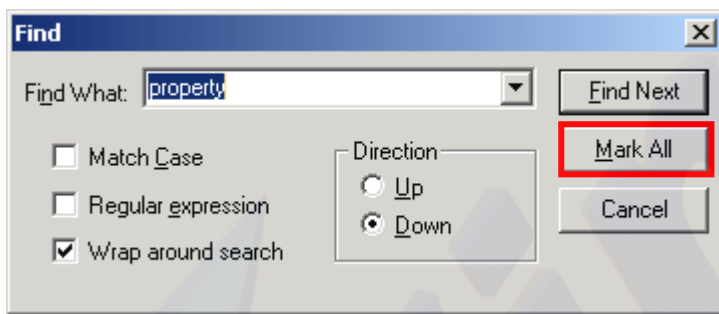
Function	Shortcut
剪下	Ctrl + X
複製	Ctrl + C
貼上	Ctrl + V
回復	Ctrl + Z
重做	Ctrl + Shift + Z
全選	Ctrl + A
搜尋	Ctrl + F
尋找下一個	F3
尋找與替代	Ctrl + H
區段選取	Ctrl + 滑鼠選取
選擇一行	點選單行左側邊框
選擇一段文字	連點文字區段
多行選擇	Shift + 點選
區段選擇	拖曳滑鼠到最後的文字

建立書籤

撰寫程式的過程中，使用者可以建立一個書籤標記，可以快速的找到相關的程式碼位置。只要先使用滑鼠選取要建立書籤的那一行程式，再點滑鼠右鍵開啓功能視窗，選擇 **Toggle Bookmark** 選項，即可在編輯器的左側建立一個藍色圓點的書籤。

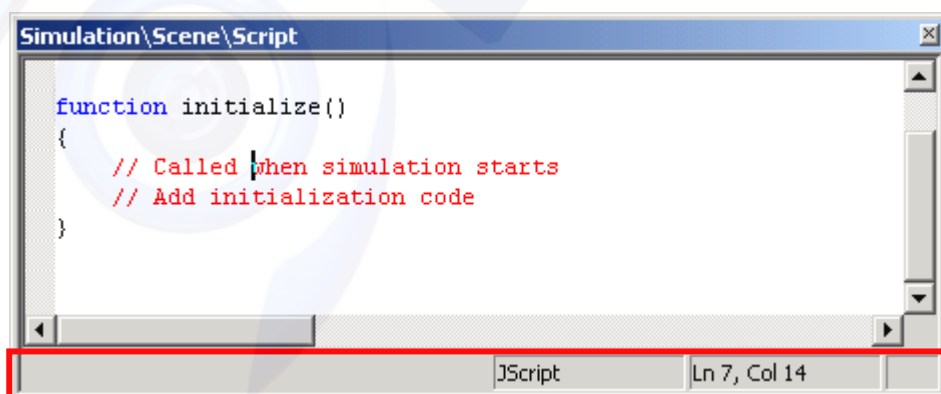


搜尋視窗中也可以建立書籤，只要在輸入文字欄裡輸入關鍵字，然後點選搜尋視窗右側的 **Mark All** 鈕，Script 編輯器就會將含有關鍵字的那一行程式，自動建立書籤標記。



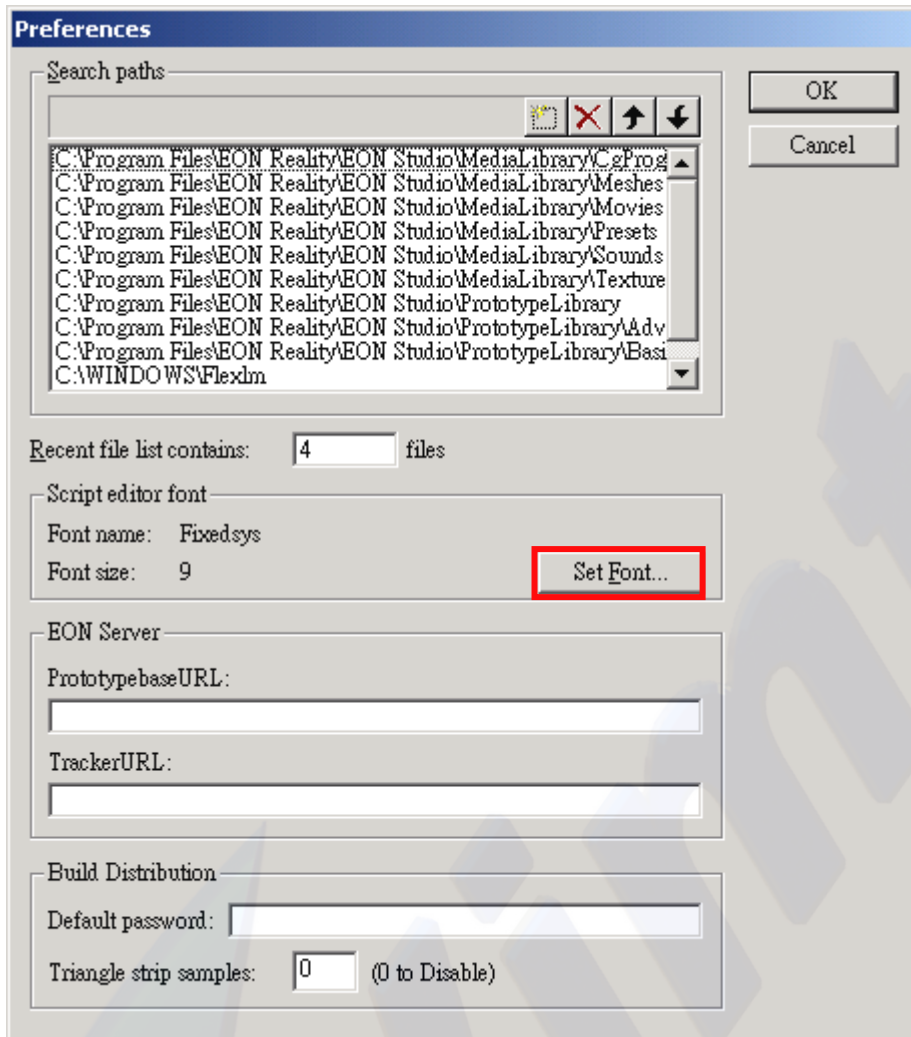
文字遊標點位置

Script 編輯器視窗的下方，有一條可以顯示文字游標目前所在的行數。由於 EON 在執行節點的腳本發生錯誤時，會跳出一個錯誤視窗顯示那一行的程式碼發生錯誤，因此利用這個功能可以輔助快速找到要修改程式的位置。



改變 Script 編輯器的字體

當要改變 Script 編輯裡的字體時，只需點選視窗上方的 Options -> Preferences 開啓偏好設定選單，再點選 Script editor font 欄位的 Set Font 鈕，即可開啓字型設定視窗。



關鍵字變色效果

在 Script 編輯器撰寫程式腳本時，可以發現在輸入一些關鍵字後，這些文字會自動以不同顏色標示。若想要另外新增一些關鍵字，可以在 Wondos 底下找到 vbscript.ini 與 jscript.ini 這二個組態設定檔案。請以記事本開啓這些檔案，在[Keywords]底下新增關鍵字與色彩類型。以下是 vbscript.ini 檔案裡部分內容的截取。

```
[Colors]
Declaration=0,0,255
Flow=0,0,255
Conversion=0,0,255
Math=0,0,255
StringManipulation=0,0,255
Comparison=0,0,255
[Keywords]
Abs=Math
And=Comparison
Array=Declaration
Asc=StringManipulation
AscB=StringManipulation
AscW=StringManipulation
Atn=Math
Case=Flow
CBool=Conversion
CByte=Conversion
```

如何引用欄位內的數值

在 Script 編輯器中所引用的欄位數值，其數值的型態取決於欄位創立時，Data Type 欄所設定的屬性。而 Data Type 欄位的數值可概略分為單一資料型態(SF)與多重資料型態(MF)的數值，單一型態欄位的數值一次只存取單一個數值，例如 SFBool 欄位一次僅能存取一個布林值。而複合型的欄位則可存取一組數值，例如 SFVec3f 欄位可以存取一組 XYZ 三個座標的參數。

而程式腳本引用欄位數值的方式，並非靠 Field Type 欄的 field、inevent、exposedfield 或 outevent 的屬性來設定，Field Type 欄的屬性只能設定該欄位的值是以何種方式傳遞。程式腳本是直接將欄位名稱視為變數，將欄位值帶入到程式腳本執行。

單一型態 Single Field (SF) 類型欄位

有一些欄位的型態例如：SFBool、SFVec3f、SFFloat 在一開始時就是單一資料型態的欄位。而 Script 編輯器內的程式腳本，引用這些欄位內的參數的方式，取決於這些欄位內的數值型態，是否為單一資料型態的數值(SF)，或是多重資料型態的數值(MF)。此二種型態欄位數值引用的方式有些許的不同。

例如一個在 EON 中標示的 3d position(座標)欄位，是由一組 x、y 和 z 三個座標所組成的，因此這個欄位的資料型態雖然是單一資料型態的數值，內容則為多重資料型的數值，所以要引用此欄位內數值的方式，就與引用 SFBool 型態的資料有所不同。

引用單一數值欄位

以下是在 Script 編輯器內，引用單一數值欄位內的參數的一般語法：

Myfieldname.value

Myfieldname 為 Scrip 節點新增的欄位名稱

.value 表示欄位的數值，但這個部分的語法在 VB 的程式語法中，引用單一數值欄位時，特別要求要使用這個語法表示。用此語法表示變數值時，VB 會將此欄位的名稱視為變數名。

範例 1

此範例中使用的語法為 VBScript scrip 的一般寫法，此語法是新增 3 個名稱為 TotalScore、NrTries 與 Score 變數並進行演算。

```
Dim TotalScore, NrTries, Score
TotalScore = 15
NrTries = 4
Score = TotalScore / NrTries
Msgbox "Your Score is " & Score
```

範例 2

以下的步驟是示範在 Script 中如何引用欄位中的變數進行演算，首先在 Script 節點的屬性欄內建立三個可以引用變數的欄位。

TotalScore of datatype SFInt32 is an exposedfield.

NrTries of datatype SFInt32 is an exposedfield.

Score of datatype SFFloat is an outevent.

然後在 Script 編輯器中輸入以下的語法，即可進行與範例 1 相同的運算，但不同的是此程式的變數值，是直接引用節點屬性欄內的數值。

```
Score = TotalScore / NrTries
```

範例 3

請注意以下的程式語法，可以發現這段語法與範例 2 的語法相同，同樣的都是直接引用節點屬性欄內的數值，但以下的寫法是最正統也最完整的寫法

```
Score.value = TotalScore.value / NrTries.value
```

範例 4

當有節點的欄位屬性為 SFVec3f，SFVec2f，SFColor 和 SFRotation 這種複合參數的欄位時，或者使用 Jscript 語法撰寫程式腳本時，在引用欄位值時就必須在變數名稱後面特別加註.value 的程式語法

例如：要引用以下二個節點裡的單一型態欄位的值時：

```
NrClicks inevent SFInt32
```

```
Force exposedfield SFFloat
```

使用 VBScript 撰寫時，可以省略用.value 的語法不用：

```
Force = NrClicks
```

但使用 Jscript 撰寫時，必須在變數名稱後面加上.value 的語法

```
Force.value = NrClicks.value
```

範例 5

要引用的節點欄是以下二個複合數值型態的欄位時：

```
Position inevent SFVec3f
```

```
NewPos exposedfield SFVec3f
```

不論是 VBScript 或 Jscript 都不接受以下的語法：

```
NewPos = Position
```

如果使用上述的語法，會造成 VBScript 執行時，產生類型不符的錯誤，這是因為 Position 的資料型態為複合數值型的欄位，因此必需使用以下的語法：

```
NewPos.value = Position.value
```

或者在 VBScript 語法，你可以以下面的方式撰寫：

```
NewPos = Position.value
```

VBScript 語法也接收以下的方式：

```
p = Position
```

```
NewPos = p
```

這是因為變數 P 已經直接被指定為複合型的陣列參數。

如果要在 Jscript 模式下，引用複合型欄位內的數值，則必需使用陣列的語法撰寫：

```
p = Position.value.toArray()  
NewPos = eon.MakeSFVec3f(p[0], p[1], p[2])
```

由於 Jscript 的語法，無法將複合型欄位內的數值轉換成變數，因此必需透過陣列的方式，先將欄位內的值轉換成各獨立的變數，再將各個變數轉成 EON 可接收的陣列型變數。

※**eon.MakeSFVec3f** 是 EON Script 內建的語法，並非 Jscript 的語法，而 **MakeSFVec3f** 的說明請參照後的章節。

範例 6

如果在 Jscript 的語法裡，去除在變數名稱後面特別加注 **.value** 語法時，則表示此變數代表的是欄位本身，不是引用欄位內的數值。

例如以下的範例，是在 Jscript 的語法裡，引用一個名稱為 **Distance** 欄位的數值：

```
d = Distance  
B = d.value
```

上述的這二行程式在 VBScript 的語法裡會產生錯誤，因為 d 的值已經是 **Distance** 這個變數，並非 **Distance** 欄位內的值。

範例 7

當需要在編輯器裡，宣告一個和節點欄位名稱相同的變數。如此設定將會節點欄位設定的數值失效，令欄位變動的數值看起來好像不存在一樣。

例如以下的範例，宣告一個名稱為 **NrClicks** 的變數，並將其值設定為-5。然後再設定引用 **NrClicks** 欄位的變化，去趨動一段程式腳本，但是在這個範例裡，**NrClicks** 變數的值將永遠為-5，而非屬性設定欄內的數值。

```
Dim NrClicks  
NrClicks = -5  
Sub On_NrClicks()  
MsgboxNrClicks  
End sub
```

如果在上述例子裡，把程式碼 **NrClicks** 改成 **NrClicks.value** 的話，模擬視窗運行時會產生”需要語法”的錯誤，因為 **NrClicks** 是變數，因此無法取代 **NrClicks** 這個欄位語法的。

單一欄位陣列型參數欄位

資料類型為 SFVec2f、SFVec3f、SFColor 與 SFRotation.的欄位，皆為單一欄位陣列型參數的欄位，要引用這些欄位的數值時，必需先將欄位內的各個陣列數值，分發到一個陣列變數裡。

例如以下程式碼，是使用 VBScript 將一組座標陣列值，分發到一個陣列變數裡。

p = Position

由於 VBScript 可以接受不用宣告變數的型態，就直接將一組陣列數值帶入到變數裡。因此可以用上述的語法，將變數 p 設成一組 SFVec3f 型態的座標值。

而使用 Jscript 的語法時，則必需先將欄位裡的陣列參數分離，然後再使用陣列的語法將各個數值帶入。

p = Position.value.toArray()

此外，使用 Jscript 的語法時，必需要特別注意變數名稱的大小寫，因為 Jscript 對於變數的大小寫比較敏感，因此必須正確的拼寫變數。例如欄位的名稱為 Position，那麼變數名稱就不能取作 position、POSITION 或 POSition，必需正確的寫為 Position，否則將無法引用該欄位的數值。

範例 8

VBScript:

```
sub On_position()  
    p=position  
    p(0)=p(0)+1  
    p(1)=p(1)+1  
    p(2)=p(2)+1  
    newPosition=p  
end sub
```

上述的程式腳本裡，只要 position 的欄位更新其數值，則腳一裡的陣列程序就會開始執行，並將新的陣列值，傳遞給其它任何有相關連結的欄位。

以下的範例裡，將使用 Jscript 的語法與上面的 VBScript 範例比較。

範例 9

Jscript:

```
function On_position()
{
    var p = position.Value.toArray();
    p[0] = p[0] + 1;
    p[1] = p[1] + 1;
    p[2] = p[2] + 1;
    newPosition.Value = eon.MakeSFVec3f(p[0], p[1], p[2])
}
```

多重資料資料類型的欄位(MF)

多重資料資料類型的欄位(MF)，是包含多個單一資料欄位(SF)數值的欄位。欄位內的數值資料看起來像是一組陣列值，但是卻又於 VBScript 或者 JScript 裡的正常陣列模式有點不同。像是 **MFFloat**、**MFNode**、**MFInt32** 等，資料型態名稱以 **MF** 開頭欄位的數值，就是可以包含 0~100 以上的單一欄位資料。由於這些多重資料資料裡的數值結構較為複雜，因此我們必需透過一些適當的語法才能正確的引用我們想要的數值。

範例 10

以下的程式碼是從一個名稱爲 **Times** 資料類型爲 **MFFloat** 的欄位裡，將第一組數值帶給變數 **t** 的範例。

```
t = Times.GetMFElement(0)
```

範例 11

以下的 VBScript 程式碼，是將 5 組數值，分別帶入到從一個具有名稱爲 **Times** 資料類型爲 **MFFloat** 的欄位裡的範例。

```
For i = 0 to Times.GetMFCount() - 1
t = Times.GetMFElement(i) - 5
Times.SetMFElement i, t
Next
```

以下爲 Jscript 的寫法

```
for (i = 0; i < Times.GetMFCount(); i++)
{
    t = Times.GetMFElement(i)-5
    Times.SetMFElement(i, t)
}
```

範例 12：由 MFVec3f 類型的欄位中，抓取 x, y, z 的座標值

在這個 VBScript 的程式碼中，抓取 MFVec3f 欄位中的第 4 組 x 的值。

```
p = Positions.GetMFElement(3)
x = p(0)
```

以下爲 Jscript 的寫法

```
p = Positions.GetMFElement(3).toArray()
x = p[0]
```

TreeChildren 欄位

每個節點裡，都會有一個名稱爲 **TreeChildren** 的欄位，此欄位是 MFNode 類型的多重資料型資料欄位，主要的功能是記錄節點底下所放置的子節點。如果要引用欄位裡第一個子節點，就可以使用以下的語法：

TreeChildren.GetMFElement(0)

如果要引用第 5 個子節點，就必需使用以下的語法：

TreeChildren.GetMFElement(4)

傳送數值

當每次要傳送多重資料資料類型的欄位(MF)裡的數值時，就必需要從欄位中抓取數值，再將數值傳遞給單一資料類型的欄位(SF)，再從流程視窗中連結相關的欄位值。如果將所要接收訊號的欄位，也設成相同型態的多重資料資料類型，就可以直接連結，略過抓取數值的步驟。

※上述的方式不適用於 MFNode 類型的欄位。

範例 13：新增數值到 MFString 類型的欄位裡

以下的範例是將 **This is my text** 這個字串，新增到 Text 這個欄位裡的語法。

```
Text.SetMFCount(0)
```

```
Text.AddMFElement( "This is my text" )
```

將上述的程式碼寫好後，可以直接使用流程視窗，將此欄位與 TextBox 的 Text 欄位作連結。

如果不想在流程視窗中進行連結，可以在程式碼中加入以下的語法，也可以達到相同的效果。

```
Eon.FindNode("TextBox").GetFieldByName("Text").value = Text.value
```

※提示：當傳送一組文字給 TextBox 節點接收後，必需要將節點隱藏後再顯示，否則節點將無法顯示新接收到的文字。

使用 Script 撰寫程式腳本時，有三種類型的 EON 語法類型可以引用，三種類型分別為：

1. **EON Base Object** (EON 基本控制語法)：可供程式腳本控制 EON 基本屬性。
2. **EON Node Object**(EON 節點控制語法)：可供程式腳本控制模擬樹底下的節點。
3. **EON Field Object**(EON 欄位控制語法)：可供程式腳本控制節點的屬性欄位。

這些控制方法之所以被稱為語法，是因為在撰寫程式腳本時，真的將這些控制方法當成語法來使用。例如：在 **VBScript** 的 **FileSystemObject(檔案系統語法)**裡就有 **Object(語法)**與 **Err Object(錯誤語法)**；**Jscript** 也有稱為 **Array Object(陣列物體)**、**Math Object(數學運算語法)**和 **FileSystemObject (檔案系統語法)**。而在使用 Script 撰寫程式腳本時，這三種類型的 EON 語法，也會被視同於這些語法的語法一，樣進行處理及引用。

每個控制語法皆有很多的控制方式與屬性設定的語法，大部份控制語法的語法中會使用到括號 ()。VBScript 並不會特別要求在語法中使用括號，除非控制這個語法的語法有特別要求。而 JScript 不論控制語法的語法有無特別的要求，在每個語法控制的語法裡都必需使用括號標示 ()，通常控制語法的語法並非都需要用到括號，除非是設定這個語法屬性欄位的數值。

請注意在使用控制 EON 語法的語法時，是不需要特別注意語法名稱的大小寫拼法。通常 Jscript 對於這一點總是會特別要求，例如在 Jscript 中引用 parseInt 這個控制語法的語法時，就不能將語法的拼法寫成 parseint 或 ParseINT。但使用控制 EON 語法的語法，就不會發生這樣的情況，只要語法的名稱拼對就可以正確執行。例如 EON 的欄位語法語法 getFieldbyname，你可以拼成 getFieldByName 或 GETFieldByName 等，都可以正常執行。

若將原本使用 VBScript 撰寫的程式腳本，轉換成 Jscript 時，如此設定可以節省改寫時間。

例如： **VBScript:**

```
dim ballframe, ballpos
set ballframe = eon.FindNode( "Ball" & nr )
set ballpos = ballframe.GetFieldByName( "Position" )
```

JScript:

```
var ballframe, ballpos
ballframe = eon.FindNode( "Ball" + n r)
ballpos = ballframe.GetFieldByName( "Position" )
```

※提示：在 VBScript 裡我們必需使用 Set 這個關鍵字，將一個變數設定成語法，而 Jscript 則不需要。

EON 的控制語法列表

以下的列表是專屬於 Eon Script 的 Eon 控制語法，以及這些語法的特性與屬性簡單描述。

EON Base Objects (EON 基本控制語法)	
MakeSFVec2f	此語法可以產生一組 2 維陣列的變數以應用在 SFVec2f 類型的欄位上。
MakeSFVec3f	此語法可以產生一組 3 維陣列的變數以應用在 SFVec3f 類型的欄位上。
MakeSFColor	此語法可以產生一組紅、綠、藍 3 色混色比例變數以應用在 SFCOLOR 類型的欄位上。
SetScriptTimeout	此語法可以設定程式腳本執行無反應到出現停止執行的對話視窗出現前的時間值。
GetScriptTimeout	此語法可以抓取程式腳本執行無反應到出現停止執行的對話視窗出現前的時間值。
Find	此語法可以提供程式腳本列出符合以名稱或其它搜尋規則條件的 EON 節點語法與數量。
FindNode	此語法可以使程式腳本以名稱搜尋模擬樹中相同名稱的 EON 節點語法。
GetNodeName	此語法可以提供程式腳本抓取節點的名稱，並將其轉換成字串。
GetNodePath	此語法可以提供程式腳本抓取完整的節點路徑，並將其轉換成字串。
MessageBox	此語法可以提供程式腳本顯示一個自行設定抬頭與內文的文字資訊視窗。
SaveSnapShot	此語法可以提供程式腳本將模擬視窗的內容儲存一個影像檔案。
EON Node Objects (EON 節點控制語法)	
GetParentNode	此語法可以提供程式腳本抓取指定節點的上層父系節點。
GetFieldCount	此語法可以提供程式腳本擷取指定節點屬性欄的總值。
GetField	此語法可以提供程式腳本依指定的欄位編號抓取指定欄位內的數值資料。
GetFieldByName	此語法可以提供程式腳本依指定的欄位名稱抓取指定欄位內的數值資料。

EON Field Objects (EON 欄位控制語法)	
Value property	此語法可以提供程式腳本抓取欄位內的數值。
GetName	此語法可以提供程式腳本抓取欄位的名稱。
GetType	此語法可以提供程式腳本抓取欄位的 datatype(數值類型)。
GetMFCount	此語法可以提供程式腳本從指定的多重資料型態欄位(MF)屬性欄擷取已儲存的參數值總筆數。
SetMFCount	此語法可以提供程式腳本設定指定的多重資料型態欄位(MF)屬性欄裡放儲存的參數值總筆數。
GetMFElement	此語法可以提供程式腳本擷取多重資料型態欄位(MF)中的某一筆參數值。
SetMFElement	此語法可以提供程式腳本修改多重資料型態欄位(MF)中的某一筆參數值。
AddMFElement	此語法可以提供程式腳本新增一筆參數值到指定的多重資料型態欄位(MF)屬性欄裡。
RemoveMFElement	此語法可以提供程式腳本從指定的多重資料型態欄位(MF)屬性欄裡移除一筆參數值。
EON 特殊事件	
Initialize	此語法可以提供程式腳本在模擬視窗開啓時進行欄位參數的初始化設定。
EventsProcessed	此語法可以提供程式腳本在偵測到欄位有變化時，將相關的參數傳到其它的欄位。
Shutdown	此語法可以提供程式腳本在模擬視窗開啓時關閉模擬視窗。

MakeSFVec2f 控制語法

語法說明：

此控制語法可以產生一組 SFVec2f 類型的變數，SFVec2f 是 EON 特有的 2 維陣列型態變數。

應用語法：

eon.MakeSFVec2f(exp1, exp2)

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 eon 開始。

exp1 與 exp2：此二個語法表示 2 維陣列的二個數值，可以將變數帶入到語法裡取帶數值。

範例：

TextBoxPos.value = eon.MakeSFVec2f(x, y)

上述語法是建立一個 SFVec2f 的參數，並將變數 X 與 Y 的值帶入到 MakeSFVec2f 的陣述式中，最後將 TextBoxPos 欄位的值以新建立的 SFVec2f 參數取代。

MakeSFVec3f 控制語法

語法說明：

此控制語法可以產生一組 SFVec3f 類型變數，SFVec2f 是 EON 特有的 3 維陣列型態變數。

應用語法：

eon.MakeSFVec3f(exp1, exp2, exp3)

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

exp1、**exp2** 與 **exp3**：此三個語法表示 3 維陣列的三個數值，可以將變數帶入到語法裡取帶數值。

範例：

Position.value = eon.MakeSFVec3f(x, y, z)

上述語法是建立一個 SFVec3f 的參數，並將變數 X、Y 與 Z 的值帶入到 **MakeSFVec3f** 的陣述式中，最後將 **Position** 欄位的值以新建立的 SFVec3f 參數取代。

MakeSFColor 控制語法

語法說明：

此控制語法可以產生一組 SFCOLOR 類型的變數，SFCOLOR 是 EON 特有的色彩混色比例值變數，可以提供 red(紅)、green(綠)、blue(藍)三種色彩參數混合的比例(數值範圍由 0~1)。

應用語法：

eon.MakeSFCOLOR (red, green, blue)

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 eon 開始。

red、green 與 blue：此三個語法表示 red(紅)、green(綠)、blue(藍) 三種色彩參數混合的比例(數值範圍由 0~1)。也可以將變數帶入到語法裡取帶數值。

範例：

NewColor.value = eon.MakeSFCOLOR(r, g, b)

上述語法是建立一個 SFCOLOR 的參數，並將變數 r、g 與 b 的值帶入到 MakeSFcolor 的陣述式中，最後將 **NewColor** 欄位的值以新建立的 SFCOLOR 參數取代。

Find 控制語法

語法說明：

此語法可以提供程式腳本列出符合以名稱或其它搜尋規則條件的 EON 節點語法與數量。

應用語法：

eon.Find(node_exp , root , maxdepth)

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

node_exp：此語法表示要搜尋的節點語法，可以直接輸入要尋找節點的名稱，或是以一個字串變數代替。

root：此語法可選擇是否要輸入，表示程式腳本由設定的路徑開始尋找。如果不輸入的話，表示由最上層的 Simulation 節點開始搜尋。

Maxdepth：此語法可選擇是否要輸入，表示程式腳本可搜尋的路徑層次深度。輸入數值 0 表示搜尋的範圍只限由 **root** 參數指定路徑那一層架構。輸入數值 1 則表示搜尋的範圍包含 **root** 參數指定路徑那一層架構，以及其下一層子節點架構。如果不輸入的話，表示不限制搜尋的深度搜尋所的層次。

注意：1、由於程式只接收字串，因此必需以”節點名”的方式，將節點名稱標示為字串，否則輸入的節點名將會被判斷成變數。

語法撰寫範例：

eon.Find(name)

eon.Find(name, root)

eon.Find(name, root, maxdepth)

如果程式腳本有搜尋到任何的語法時，需使用 **Count** 的方式來顯示找到的數量。例如：

Set Fn = eon.find("Script")

AF = Fn.count

如果要引用搜尋到的語法時，需使用 **item(i)**的方式來引用找到的語法，其中 **i** 表示第幾個語法。例如：

Set Fn = eon.find("Script")

Set AF = Fn.item(0)

實際範例：

VBScript:

```
set mynodes = eon.find("Frame")
if mynodes.Count > 0 then
set firstnode = mynodes.item(0)
firstnode.getfieldbyname("SetRun").value = false
end if
```

JScript:

```
mynodes = eon.find("Frame")
if (mynodes.Count > 0)
{
firstnode = mynodes.item(0)
firstnode.getfieldbyname("SetRun").value = false
}
```

上述語法是建立一個名稱爲 **mynodes** 的語法，然後在模擬樹中找尋所有名稱爲**Frame**的節點，並判斷最後搜尋的結果是否大於 0 個。如果條件成立則觸發找到的第一個 **Frame** 節點。

Find Node 控制語法

語法說明：

此語法可以使程式腳本以名稱搜尋模擬樹中相同名稱的 EON 節點語法。

應用語法：

`eon.FindNode([parentnode1!][parentnodeN!]...nodename)`

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

parentnode1!：此語法可選擇是否要輸入，表示要搜尋的上層路徑，可以直接輸入要尋找的上層節點名稱，或是以一個字串變數代替，而每一個不同階層的節點，必需使用！分隔。如果不輸入則表示由最上層的 **Simulation** 節點開始搜尋。

nodename：此語法表示要搜尋的節點的名稱，你可以直接輸入要尋找節點的名稱，或是以一個字串變數代替。

注意：1、由於程式只接收字串，因此必需以“上層節點名！節點名”的方式，將節點名稱標示為字串，否則輸入的節點名將會被判斷成變數。

範例：

VBScript example:

```
dim aNode
set aNode = eon.FindNode("parentnode1!UniqueScript")
```

JScript example:

```
var aNode
aNode = eon.FindNode( "parentnode1!UniqueScript" )
```

上述語法是建立一個名稱為 **aNode** 的語法，然後在模擬樹中找尋位於 **parentnode1** 底下所有名稱為 **UniqueScript** 的節點，並將 **aNode** 設成找尋到的節點。

GetNodeName 控制語法

語法說明：

此語法可以提供程式腳本抓取節點的名稱，並將其轉換成字串。

應用語法：

```
eon.GetNodeName(node)
```

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

node：此語法為要抓取名稱的 EON 節點語法。

範例：

```
childnode = this.GetFieldByName("TreeChildren").GetMFEElement(3)  
scriptname = eon.GetNodeName(childnode)
```

上述語法是先將 **childnode** 設為抓取節點 **TreeChildren** 欄位裡的第四節點連結，然後再將 **scriptname** 這個變數設為抓取到的節點名稱。

※注意：1、由於程式只接收字串，因此必需以”上層節點名！節點名”的方式，將節點名稱標示為字串，否則輸入的節點名將會被判斷成變數。

GetNodePath Method 控制語法

語法說明：

此語法可以提供程式腳本抓取完整的節點路徑，並將其轉換成字串。

應用語法：

`eon. GetNodePath (node)`

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

node：此語法為要抓取名稱的 EON 節點語法。

範例：

`nodepath = eon.GetNodePath(this)`

上述語法是將 **nodepath** 這個變數設為抓取到的 Script 節點完整路徑。

GetScriptTimeout 控制語法

語法說明：

此語法可以抓取程式腳本執行無反應到出現停止執行的對話視窗出現前的時間值。

應用語法：

eon.GetScriptTimeout()

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

範例：

t = eon.GetScriptTimeout()

上述語法是將 **t** 這個變數設為抓取到的無反應時間。

SetScriptTimeout 控制語法

語法說明：

此語法可以設定程式腳本執行無反應到出現停止執行的對話視窗出現前的時間值。

應用語法：

eon.SetScriptTimeout(exp1)

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

exp1：此語法表示設定的時間，此語法必需為整數，其設定單位為秒。如果此語法設定值為 0 時，則可關閉執行逾時的對話視窗。

範例：

```
tid = eon.SetScriptTimeout(0)
msgbox "Correct. Proceed to Question 5."
eon.SetScriptTimeout(tid)
```

上述語法是直接關閉執行逾時的對話視窗。並直接顯示另一個資訊對話視窗。

MessageBox 控制語法

語法說明：

此語法可以提供程式腳本顯示一個自行設定抬頭與內文的文字資訊視窗。

應用語法：

eon. MessageBox(message, title)

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 eon 開始。

message：此語法為資訊視窗顯示的文字內容的字串。如果字串的長度太長，可以另外建立一個字串變數，並且將這個變數帶入到此語法中。

title：此語法為資訊視窗上方顯示抬頭的字串。

注意：如果文字內容太長，VBScript 語法可以使用「Chr(i)」，Jscript 則可以使用「\n」的換行語法，將文字轉換成多行文字。

註記：

此語法在 VBScript 中使用時，由於 VBScript 不需要使用括號「()」的來標示參數值，因此其語法為：

eon.MessageBox message, title

而在 Jscript 中使用時，其語法則是因為 Jscript 一定要使用括號「()」的來標示參數值，因此其語法為：

eon. MessageBox(message, title)

範例：

VBScript

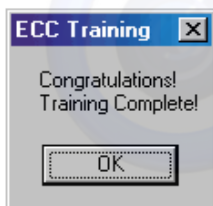
```
msg = "Congratulations!" & Chr(13) & "Training Complete!"
```

```
eon.MessageBox msg, "ECC Training"
```

JScript:

```
msg = "Congratulations!\nTraining Complete!"
```

```
eon.MessageBox(msg, "ECC Training")
```



SaveSnapShot 控制語法

語法說明：

此語法可以提供程式腳本將模擬視窗的內容儲存一個影像檔案。

應用語法：

`eon.SaveSnapShot(filename, format, param)`

eon：此語法是使用 EON 基本語法的操控方式，因此在程式的開頭必需以 **eon** 開始。

filename：此語法為輸入儲存的檔案位置及其路徑的字串。

format：此語法為輸入存檔格式的代號。(0 = BMP, 1 = PPM, 2 = PNG, 3 = JPEG).

param：此語法為保留功能，其值只能設為 0。

註記：

- 1.目前儲存的格式尚不支援 BMP 和 JPEG，在之後的版本將會修改此一問題。
- 2.如果輸入檔案路徑的資料夾不存在，將無法儲存影像
- 3.在 Jscript 裡輸入存檔路徑時，資料夾與資料夾間必需使用雙斜線「//」分隔。而 VBScript 則只需單斜線「/」即可。

- 4.此語法在 VBScript 中使用時，由於 VBScript 不需要使用括號「()」的來標示參數值，因此其語法為：

eon. SaveSnapShot filename, format, param

- 5.而在 Jscript 中使用時，其語法則是因為 Jscript 一定要使用括號「()」的來標示參數值，因此其語法為：

eon. SaveSnapShot (filename, format, param)

範例：

VBScript:

`eon.SaveSnapShot "C:/myimage.png", 2, 0`

`eon.SaveSnapShot "C:/_PROJECTS/Testing/myimage2.png", 2, 0`

JScript:

`eon.SaveSnapShot("C://myimage.png", 2, 0)`

`eon.SaveSnapShot("C://_projects//testing//myimage.png", 2, 0)`

上述的語法為擷取目前模擬視窗中的畫面，並將其儲存在 C:/與 C:/_PROJECTS/Testing/這二個地方，圖檔名稱皆為 myimage.png。但 C 底下並沒有 C:/_PROJECTS/Testing 這個資料夾，因此第二個存檔指令將無法執行。

GetParentNode 控制語法

語法說明：

此語法可以提供程式腳本抓取指定節點的上層父系節點。

應用語法：

node.GetParentNode()

node：此語法為 EON 的節點語法。由於此語法是使用 EON 節點的操控語法，因此在程式的開頭必需要標示要使用此語法的節點。

注記：

set parent = boxnode.value.GetParentNode()

上述的 boxnode 是 SFNode 類型欄位的名稱。

使用此控制語法，可以使程式腳本抓取目標節點上層的父亲節點，並且透過腳本的設計，作進一步的控制。

範例：

VBScript:

```
sub On_CurrentTime()  
dim h  
h = CurrentTime.value * 36  
set parent = this.GetParentNode()  
parent.GetFieldByName("Orientation").value = eon.MakeSFVec3f(h, 0, 0)  
end sub
```

JScript:

```
function On_CurrentTime()  
{  
var parent, h  
h = CurrentTime.value * 36  
parent = thisNode.value.GetParentNode()  
parent.GetFieldByName( "Orientation" ).value = eon.MakeSFVec3f(h, 0, 0)  
}
```

上述的二個程式腳本，是在 Script 節點中新增一個 CurrentTime eventIn SFFloat 的欄位，並從 TimeSensor 接收 CurrentTime 訊號作為觸發腳本的運訊號。然後新增 parent 與 h 這二個變數，設定 h 變數等於 CurrentTime 欄位值乘於 36，並且指定 parent 變數等於 Script 節點上層的 Frame 節點，最後再將 h 變數的值帶入到 Script 節點上層的 Frame 節點的 h 軸旋轉角度。

GetFieldCount 控制語法

語法說明：

此語法可以提供程式腳本擷取指定節點屬性欄的總值。

應用語法：

node.GetFieldCount()

node：此語法為 EON 的節點語法。此語法是使用 EON 節點的操控語法，因此在程式的開頭必需要標示要使用此語法的節點。

註記：

節點欄位的總數量總是會少 7 項，這是因為節點預設的欄位(SetRun、SetRun_等…)不列入計算的範圍。

範例：

```
if mynode.value.GetFieldCount() = 19 then  
A = 19  
end if
```

上述語法是新增一個 **mynode** exposedField SFNode 的欄位，然後判斷欄位內的指定的節點屬性欄是否等於 19 個。如果條件成立變數 A 的值等於 19。

GetField 控制語法

語法說明：

此語法可以提供程式腳本依指定的欄位編號抓取指定欄位內的數值資料。

應用語法：

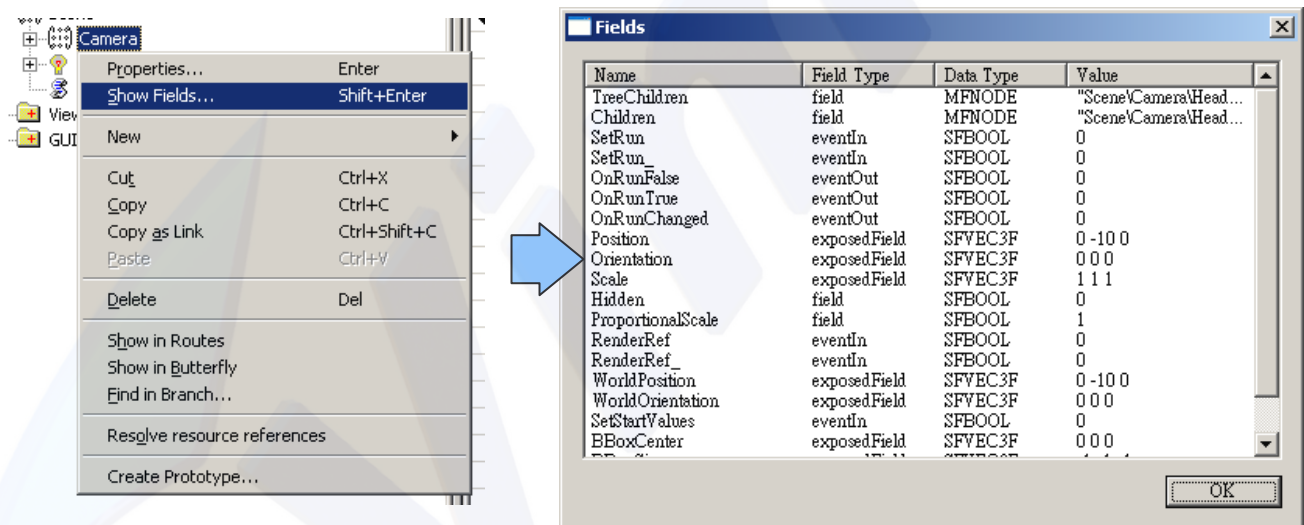
`node.GetField(index)`

node：此語法為 EON 的節點語法。是使用 EON 節點的操控語法，因此在程式的開頭必須需要標示要使用此語法的節點。

index：此語法為節點屬性列欄位的編號。

註記：

每個節點的屬性列的內容都不同，因此要找尋節點屬性列欄位的編號時，只需在節點上點一下滑鼠右鍵，選擇 **Show Fields** 選項，即可開啓節點的屬性欄位列表。節點屬性欄位的編號是由 0 開始，因此如果要控制第三個 **SetRun** 欄位時，輸入的欄位編號就為 2。



範例：

```
set SR = eon.findnode("Rotate").GetField(2)
SR.value = true
```

上述語法是在模擬樹中找尋 **Rotate** 節點的第三項 **SetRun** 屬性欄，並將此欄位的值設為 **true**。

GetFieldByName 控制語法

語法說明：

此語法可以提供程式腳本依指定的欄位名稱抓取指定欄位內的數值資料。

應用語法：

node.GetFieldByName(name)

node：此語法為 EON 的節點語法。是使用 EON 節點的操控語法，因此在程式的開頭必須需要標示要使用此語法的節點。

index：此語法為要尋找節點的名稱。

註記：

如果輸入的節點名稱錯誤，則會產生程式腳本執行逾時的清況。

範例：

```
eon.findnode("Viewport").GetFieldByName("FieldOfView").value = 80
```

上述語法是尋找 Viewport 節點的 FieldOfView 欄位，並將其值改為 80。

Value 讀取方式

語法說明：

此方法為程式腳本讀取欄位內的數值。

應用語法：

field.value

field：為屬性欄位的名稱。

註記：

如果要在程式腳本裡直引用 Script 節點內的數值時，必需在欄位名稱後面加注 .value 這個語法，否則在執行腳本時 Script 節點無法判斷語法是欄位值或是變數，因此而產生型態不明的錯誤視窗。

範例：

r = Radius.value

上述語法是將 r 這個變數的值，設定成等於 Radius 欄位的值。

GetName 控制語法

語法說明：

此語法可以提供程式腳本抓取欄位的名稱。

應用語法：

field.GetName()

field：為屬性欄位的名稱。

範例：

VBScript:

```
If f.GetName() = "Width" Then f.Value = 100
```

JScript:

```
if (f.GetName() == "Width") f.Value = 100
```

上述語法是判斷 f 這個變數所代表的欄位名稱是否為 **Width**，如果條件成立則變數 f 所代表的欄位值等於 100。

GetType 控制語法

語法說明：

此語法可以提供程式腳本抓取欄位的 **datatype**(數值類型)。

應用語法：

field.GetType ()

field：為屬性欄位的名稱。

註記：

使用此方式所抓取到的欄位類型將會以數值顯示，以下為各 **datatype**(數值類型)的列表。

數值類型	代碼
SFBool	0
SFColor	1
SFFloat	2
SFImage	3
SFInt32	4
SFNode	5
SFRotation	6
SFString	7
SFTime	8
SFVec3f	9
SFVec2f	10
MFBool	11
MFColor	12
MFFloat	13
MFImage	14
MFInt32	15
MFNode	16
MFRotation	17
MFString	18
MFTime	19
MFVec3f	20
MFVec2f	21

範例：

VBScript:

if f.GetType() = 0 then f.value = true

上述語法是判斷 f 這個變數所代表的欄位類型是否為 SFBool，如果條件成立則變數 f 所代表的欄位值等於 true。

GetMFCount 控制語法

語法說明：

此語法可以提供程式腳本從指定的多重資料型態欄位(MF)屬性欄擷取已儲存的參數值總筆數。

應用語法：

`field.GetMFCount ( )`

field：為屬性欄位的名稱。

範例：

VBScript:

```
if Times.GetMFCount() = 0 then exit sub
for i = 0 to Times.GetMFCount()-1
total = total + Times.GetMFElement(i)
next
Average = total/Times.GetMFCount()
```

JScript:

```
if (Times.GetMFCount() == 0) return;
var total = 0
for (var i = 0; i < Times.GetMFCount(); i++)
{
total = total + Times.GetMFElement(i)
}
Average.value = total/Times.GetMFCount()
```

上述語法是判斷 Times 這個欄位裡是否有多重資料的資料存在，如果有資料則 total 變數的值等於 Times 欄位裡的多重資料資料數值總合，Average 等於這些資料數值的平均值。

SetMFCount 控制語法

語法說明：

此語法可以提供程式腳本設定指定的多重資料型態欄位(MF)屬性欄裡放儲存的參數值總筆數。並自動新增或刪除不符設定值的欄位數。

應用語法：

`field.SetMFCount (count )`

field：為屬性欄位的名稱。

Count：此語法為設定多重資料欄位裡的資料總筆數。

範例：

VBScript:

```
sub On_Position()
Text.SetMFCount(0)
p = Position.value
Text.AddMFEElement("X = " & Round(p(0), 3))
Text.AddMFEElement("Y = " & Round(p(1), 3))
Text.AddMFEElement("Z = " & Round(p(2), 3))
eon.findNode("TextBox").GetFieldByName("Text").value = Text.value
end sub
```

JScript:

```
function On_Position()
{
Text.SetMFCount(0)
var p = Position.value.toArray()
Text.AddMFEElement("X = " + Math.round(p[0]*1000)/1000)
Text.AddMFEElement("Y = " + Math.round(p[1]*1000)/1000)
Text.AddMFEElement("Z = " + Math.round(p[2]*1000)/1000)
eon.findNode("TextBox").GetFieldByName("Text").value = Text.value
}
```

上述語法是先將 Text 這個多重資料欄位的資料總筆數設為只有 1 筆，然後將新增三筆亂數產生的文字資料到 Text 這個多重資料欄位裡，再將欄位的值傳送給 TextBox 節點接收。

語法說明：

此語法可以提供程式腳本擷取多重資料型態欄位(MF)中的某一筆參數值。

應用語法：

field.GetMFElement(number)

field：為屬性欄位的名稱。

number：為指定要擷取多重資料欄位資料的編號。

註記：

以下為單一資料欄位與多重資料欄位抓取欄位值的語法比較。

b = singlefield.value (單一資料欄位)

b = multifield.GetMFElement(0) (多重資料欄位)

切記，如果使用上述語法所抓取的數值為 SFNode 型態時，在 VBScript 的語法中，必需使用關鍵字 Set 建立一個空白的語法變數，再將建立的語法變數設定成抓取 SFNode 型態數值，或者在 Script 裡新增一個 SFNode 型態的欄位，將此欄位的值設定成抓取到的數值。

如果抓取的多重欄位編號不存在任何數值時，將會出現數值超過範圍的錯誤視窗。例如：使用 GetMFCount()的方式取得多重資料欄位裡有 4 筆資料，而多重資料欄位的編號排序是由 0 開始。所以使用 GetMFElement(4)這個語法抓取最後一筆資料時，將會產生數值超過範圍的錯誤。因此在使用 GetMFElement 語法抓取資料時，一定要將擷取的欄位順序減一才不會有數值超過範圍的情況發生。

範例：

```
set SceneNode = eon.findnode("Scene")
set SceneTreeChildrenField = Scenenode.GetField(0)
set CameraNode = SceneTreeChildrenField.GetMFElement(0)
```

上述語法是搜尋 Scene 這個節點的 ChildrenField 欄位，並將 CameraNode 這個語法變數設成 ChildrenField 這個多重欄位裡的第一個數值。

SetMFElement 控制語法

語法說明：

此語法可以提供程式腳本修改多重資料型態欄位(MF)中的某一筆參數值。

應用語法：

field.SetMFElement(number, value)

field：為屬性欄位的名稱。

number：為指定要修改多重資料欄位資料的編號。

value：為符合欄位類型的修改或變更數值。如果輸入的數值類型與欄位的數值類型不符，例如輸入的數值類型為 **SFInt32**，而欄位的類型為 **SFBool**，則在程式執行時將會跳出類型不符的錯誤視窗。

註記：

以下為單一資料欄位與多重資料欄位修改欄位值的語法比較。

singlefield.value = b

multiplefield.SetMFElement(0, b)

要修改的多重欄位編號不存在任何的數值時，將會出現數值超過範圍的錯誤視窗。例如：使用 **GetMFCount()** 的方式取得多重資料欄位裡有 4 筆資料，而多重資料欄位的編號排序是由 0 開始。所以使用 **SetMFElement(4, Value)** 這個語法修改最後一筆資料時，將會產生數值超過範圍的錯誤。因此在使用 **SetMFElement** 語法抓取資料時，一定要將擷取的欄位順序減一才不會有數值超過範圍的情況發生。

此語法在 VBScript 中使用時，由於 VBScript 不需要使用括號「**()**」的來標示參數值，因此其語法為：

con.multiplefield.SetMFElement 0, b

而在 Jscript 中使用時，其語法則是因為 Jscript 一定要使用括號「**()**」的來標示參數值，因此其語法為：

con. multiplefield.SetMFElement(0, b)

範例：

VBScript:

```
dim Margins  
set Margins = eon.findnode("TextBox").GetFieldByName("Margins")  
Margins.SetMFElement 2, 30
```

JScript:

```
var Margins  
Margins = eon.findnode("TextBox").GetFieldByName("Margins")  
Margins.SetMFElement(2, 30)
```

上述的範例是定義一個名稱爲 Margins 的語法變數，並將其設爲 TextBox 節點中 Margins 這個多重欄位的值。然後將 Margins 第三筆數值改爲 30

AddMFEElement 控制語法

語法說明：

此語法可以提供程式腳本新增一筆參數值到指定的多重資料型態欄位(MF)屬性欄裡。

應用語法：

field.AddMFEElement(number, value)

field：為屬性欄位的名稱。

number：為指定要修改多重資料欄位資料的編號。

value：為符合欄位類型的數值。如果輸入的數值類型與欄位的數值類型不符例如輸入的數值類型為 **SFInt32**，而欄位的類型為 **SFBool**，則在程式執行時將會跳出類型不符的錯誤視窗。

註記：

理論上可以給不受限制的為多重資料欄位新增一筆新的數值。但在實際執行時，過多筆數的資料會因為硬體與系統的不同，產生不同程度的筆數限制。

任何一筆新增的資料，其編號將會自動被排序到最後一筆，且多重資料欄的資料總筆數會自動加一。

範例：

```
VBScript:
sub On_Position()
Text.SetMFCount(0)
p = Position.value
Text.AddMFEElement("X = " & Round(p(0), 3))
Text.AddMFEElement("Y = " & Round(p(1), 3))
Text.AddMFEElement("Z = " & Round(p(2), 3))
con.findNode("TextBox").GetFieldByName("Text").value = Text.value
end sub
```

JScript:

```
function On_Position()
{
    Text.SetMFCount(0)
    var p = Position.value.toArray()
    Text.AddMFElement("X = " + Math.round(p[0]*1000)/1000)
    Text.AddMFElement("Y = " + Math.round(p[1]*1000)/1000)
    Text.AddMFElement("Z = " + Math.round(p[2]*1000)/1000)
    eon.findNode("TextBox").GetFieldByName("Text").value = Text.value
}
```

上述語法是先將 Text 這個多重資料欄位的資料總筆數設為只有 1 筆，然後將新增三筆亂數產生的文字資料到 Text 這個多重資料欄位裡，再將欄位的值傳送給 TextBox 節點接收。

RemoveMFElement 控制語法

語法說明：

此語法可以提供程式腳本從指定的多重資料型態欄位(MF)屬性欄裡移除一筆參數值。

應用語法：

field.RemoveMFElement (number)

field：為屬性欄位的名稱。

number：為指定要移除的多重資料欄位資料編號。由於多重資料欄位的編號排序是由 0 開始。所以使用 **RemoveMFElement** 這個語法移除資料時，一定要將擷取的欄位順序減一，才不會有數值超過範圍或刪除錯誤的情況發生。

註記：

當移除一筆多重資料欄位內的數值資料後，欄位內的資料順序編號將會自動的重新排列，排序在移除資料後面的編號將會自動往前遞補。例如有一個內含 8 筆數值資料的多重資料欄位，使用 **RemoveMFElement(4)** 的語法刪除欄位裡的第 5 筆資料後，原本編號為第 6 筆的資料，將會自動向前遞補成第 5 筆資料，第 7 筆遞補成第 6 筆。

如果要移除多重資料欄位裡不存在的一筆資料時，例如欄位裡只有 3 筆資料，但程式腳本卻要移除第 4 筆資料，Script 將會忽視此一行程式，並且不會有任何的錯誤視窗出現。

範例：

JScript:

```
ct = CurrentTime.value
lt = LastTime.value
t = ct-lt
Times.AddMFElement(t)
while (Times.GetMFCount()>5)
{
    Times.RemoveMFElement(0) //remove first element
}
for (var i=0; i<5; i++)
{
    t = t + Number(Times.GetMFElement(i))
}
Average.value = t/Times.GetMFCount()
FrameRate.value = 1/(Average.value)
LastTime.value = ct
```

上述的語法是由 CurrentTime 這個欄位接收數值，並將接收到的數值新增到 Times 欄位裡，如果 Times 欄位內所記錄的數值筆數超過 5 筆，則將第一筆數值刪除。最後再計算 Times 欄位之前記錄的 5 筆數值的平均值。

Initialize 初始化

語法說明：

此語法可以提供程式腳本在模擬視窗開啓時，進行欄位內數值或變數值的初始化設定。

範例：

VBScript:

```
dim count
sub initialize()
    count = 0
end sub
```

JScript:

```
var count
function initialize()
{
    count = 0
}
```

※由於 Jscript 對於大小寫比較敏感，因此在使用大小寫的語法時，你必須使用 initialize 這個正確的拼法，使用 Initialize 或 INITIALIZE 等其它的拼法都是無效的。

上述的語法是在模擬視窗開啓時，將 count 這個變數的值設為 0。

EventsProcessed 事件處理

語法說明：

EventsProcessed 是一個相當特別的語法，當這個語法所包含的程式碼裡，只有要一個變數值有所改變時，其相關對應的流程連結、欄位數值及 Script 程式，都會隨著改變。

此語法的用途是用來降低程式腳本在執行時，使用額外的資源頻率，使程式腳本可以更快的執行。有些時候 Script 節點的接收數值的屬性欄位，會比輸出數值的欄位多，而每當節點接收到一個新的數值時，就表示要執行某一個或數個程式腳本，並且輸出新的數值，而這些程式腳本的都是各自獨立的個體，因此當新的數值改變後，只是欄位的數值改變，並非重新執行運算，如果要得到新的運算結果，只能重新執行程式腳本，因此使用 EventsProcessed 將可解決此一問題。

範例：

VBScript:

```
sub EventsProcessed()  
    p1=position1  
    p2=position2  
    distance = Sqr((p1(0)-p2(0))^2+(p1(1)-p2(1))^2+(p1(2)-p2(2))^2)  
end sub
```

JScript:

```
function eventsProcessed()  
{  
    p1=position1.value.toArray()  
    p2=position2.value.toArray()  
    var dsqr = Math.pow(p1[0]-p2[0],2)+Math.pow(p1[1]-p2[1], 2)+Math.pow(p1[2]-p2[2], 2)  
    distance.value = Math.sqrt(dsqr)  
}
```

※由於 Jscript 對於大小寫比較敏感，因此在使用大小寫的語法時，你必須使用 eventsProcessed() 這個正確的拼法，使用 EventsProcessed 或 EVENTSPROCESSED 等其它的拼法都是無效的。

上述的語法是即時計算 position1 與 position 2 這二個座標點間的距離，並將 distance 這個變數的值設為計算後的結果。

Shutdown 模擬視窗關閉語法

語法說明：

Shutdown 是一個相當特別的語法，這個語法當關閉模擬視窗後所執行的副程式。此語法可以在模擬視窗關閉後，將最後的數據儲存起來，以便下次再開啓時以最後數據顯示。這是適用於透過連結傳送意見的輸出事件，因為模擬視窗可能已經關閉並且無法接收任何訊號。

範例：

VBScript:

```
sub shutdown()  
    eon.msgbox "Thanks for viewing my 3d model. Welcome back soon!", "GOOD BEY"  
end sub
```

JScript:

```
function shutdown()  
{  
    eon.msgbox ("Thanks for viewing my 3d model. Welcome back soon!", "GOOD BEY")  
}
```

※由於 Jscript 對於大小寫比較敏感，因此在使用大小寫的語法時，你必須使用 shutdown 這個正確的拼法，使用 Shutdown 或 SHUTDOWN 等其它的拼法都是無效的。

上述的語法是在模擬視窗關閉後，顯示一個文字說明視窗。