

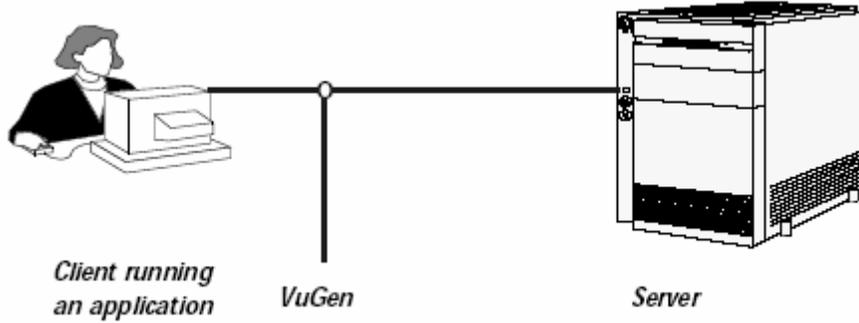
如何做關聯 (Correlation)

目 錄

1. 爲什麼要做關聯 (CORRELATION) ?	3
2. 何謂關聯?	3
3. 哪些錯誤代表著我應該做關聯?	4
4. 要如何做關聯?	5
4.1 關聯 (CORRELATION) 函數.....	5
4.2 如何找出要關聯 (CORRELATION) 資料.....	5
4.2.1 自動關聯.....	5
4.2.2 手動關聯.....	9

1. 爲什麼要做關聯 (correlation) ？

當錄製腳本時，VuGen 會攔截 client 端（瀏覽器）與 server 端（網站伺服器）之間的對話，並且通通記錄下來，產生腳本。在 VuGen 的 Recording Log 中，您可以找到瀏覽器與伺服器之間所有的對話，包含通訊內容、日期、時間、瀏覽器的請求、伺服器的回應內容等等。腳本和 Recording Log 最大的差別在於，腳本只記錄了 client 端要對 server 端所說的話，而 Recording Log 則是完整紀錄二者之間的對話。



當執行腳本時，您可以把 VuGen 想像成是一個演員，它偽裝成瀏覽器，然後根據腳本，把當初真的瀏覽器所說過的話，再對網站伺服器重新說一遍，VuGen 企圖騙過服務器，讓服務器以爲它就是當初的瀏覽器，然後把網站內容傳送給 VuGen。

所以紀錄在腳本中要跟伺服器所說的話，完全與當初錄製時所說的一樣，是寫死的 (hard-coded)。這樣的作法在遇到有些比較聰明的伺服器時，還是會失效。這時就需要透過「關聯 (correlation)」的做法來讓 VuGen 可以再次成功地騙過伺服器。

2. 何謂關聯？

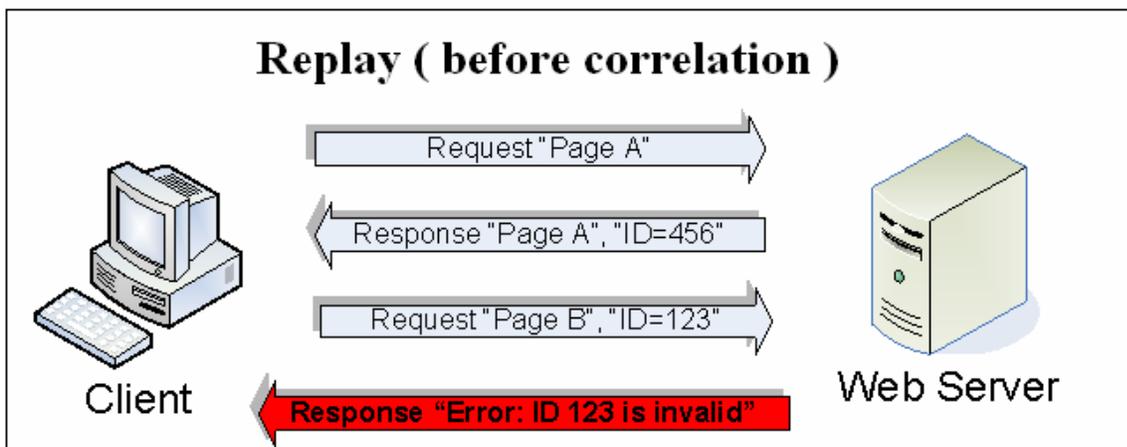
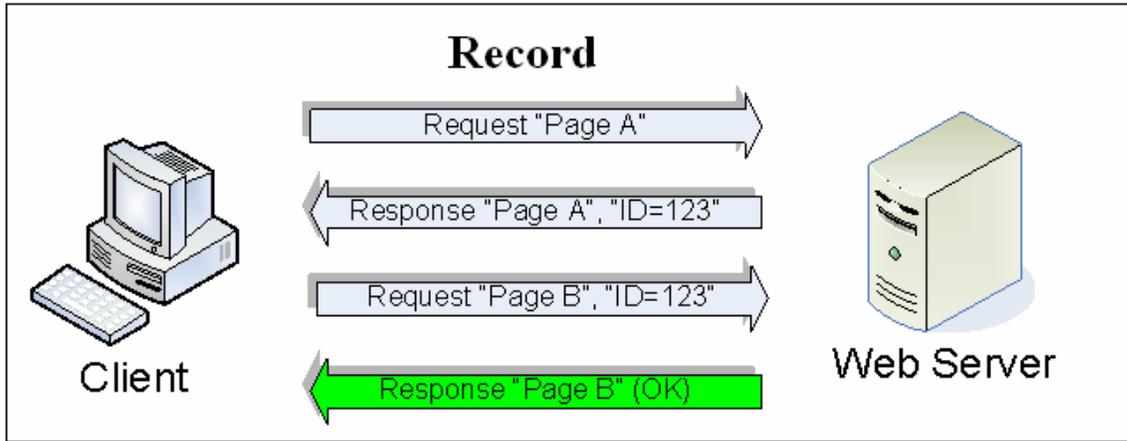
所謂的關聯 (correlation) 就是把腳本中某些寫死的 (hard-coded) 資料，轉變成是擷取自伺服器所送的、動態的、每次都不同的資料。

舉一個常見的例子，剛剛提到有些比較聰明的伺服器，這些伺服器在每個瀏覽器第一次跟它要資料時，都會在資料中夾帶一個唯一的辨識碼，接下來就會利用這個辨識碼來辨識跟它要資料的是不是同一個瀏覽器。一般稱這個辨識碼爲 Session ID。對於每個新的交易，伺服器都會產生新的 Session ID 給瀏覽器。這也就是爲什麼執行腳本會失敗的原因，因爲 VuGen 還是用舊的 Session ID 向伺服器要資料，伺服器會發現這個 Session ID 是失效的或是它根本不認識這個 Session ID，當然就不會傳送正確的網頁資料給 VuGen 了。

下面的圖示說明了這樣的情形：

當錄製腳本時，瀏覽器送出網頁 A 的請求，伺服器將網頁 A 的內容傳送給瀏覽器，並且夾帶了一個 ID=123 的資料，當瀏覽器再送出網頁 B 的請求時，這時就要用到 ID=123 的資料，伺服器才會認爲這是合法的請求，並且把網頁 B 的內容送回給瀏覽器。

在執行腳本時會發生什麼狀況？瀏覽器再送出網頁 B 的請求時，用的還是當初錄製的 ID=123 的資料，而不是用伺服器新給的 ID=456，整個腳本的執行就會失敗。



要對付這種伺服器，我們必須想辦法找出這個 Session ID 到底是什麼、位於何處，然後把它擷取下來，放到某個參數中，並且取代掉腳本中有用到 Session ID 的部份，這樣就可以成功騙過伺服器，正確地完成整個交易了。

3. 哪些錯誤代表著我應該做關聯？

假如腳本需要關聯 (correlation)，在還沒做之前是不會執行通過的，也就是說會有錯誤訊息發生。不過，很不幸地，並沒有任何特定的錯誤訊息是和關聯 (correlation) 有關係的。會出現什麼錯誤訊息，與系統實做的錯誤處理機制有關。錯誤訊息有可能會提醒您要重新登入，但是也有可能直接就顯示 HTTP 404 的錯誤訊息。

4. 要如何做關聯？

4.1 關聯 (correlation) 函數

關聯 (correlation) 會用到下列的函數：

函數	描述
web_reg_save_param	這是最新版，也是最常用來做關聯 (correlation) 的函數。 語法： web_reg_save_param ("Parameter Name", <List of Attributes>, LAST);
web_create_html_param	這二個函數主要是保留作為向前相容的目的的。建議使用 web_reg_save_param 函數。
web_create_html_param_ex	

詳細用法請參考使用手冊。在 VuG

en 中點選 **【Help】 > 【Function reference】 > 【Contexts】 > 【Web and Wireless Vuser Functions】 > 【Correlation Functions】**。

4.2 如何找出要關聯 (correlation) 資料

簡單的說，每一次執行時都會變動的值，就有可能需要做關聯 (correlation)。

VuGen 提供二種方式幫助您找出需要做關聯 (correlation) 的值：

1. 自動關聯
2. 手動關聯

4.2.1 自動關聯

VuGen 內建自動關聯引擎 (auto-correlation engine)，可以自動找出需要關聯的值，並且自動使用關聯函數建立關聯。

自動關聯提供下列二種機制：

關聯機制	描述
Rules Correlation	<p>在錄製過程中VuGen會根據訂定的規則，即時自動找出要關聯的值。規則來源有兩種：</p> <ul style="list-style-type: none"> • 內建 (Built-in Correlation) VuGen已經針對常用的一些應用系統，如AribaBuyer、BlueMartini、BroadVision、InterStage、mySAP、NetDynamics、Oracle、PeopleSoft、Siebel、SilverJRunner等，內建關聯規則，這些應用系統可能會有一種以上的關聯規則。您可以在【Recording Options】 > 【Internet Protocol】 > 【Correlation】中啟用關聯規則，則當錄製這些應用系統的腳本時，VuGen會在腳本中自動建立關聯。 您也可以【Recording Options】 > 【Internet Protocol】 > 【Correlation】檢視每個關聯規則的定義。 • 使用者自訂 (User-defined Rules Correlation) 除了內建的關聯規則之外，使用者也可以自訂關聯規則。您可以在

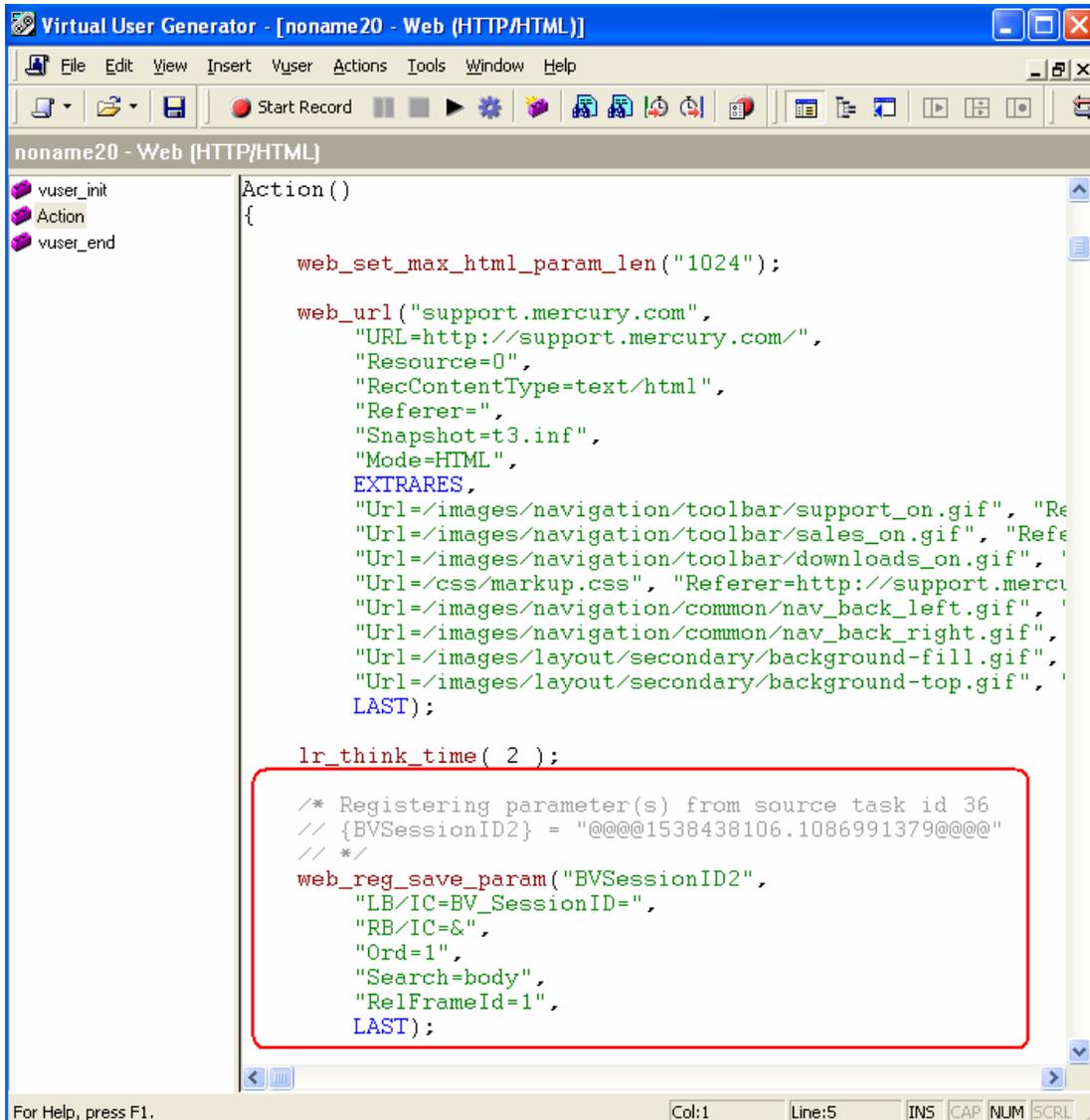
	【Recording Options】>【Internet Protocol】>【Correlation】建立新的關聯規則。
Correlation Studio	有別於Rules Correlation，Correlation Studio則是在執行腳本後才會建立關聯，也就是說當錄製完腳本後，腳本至少須被執行過一次，Correlation Studio才會作用。Correlation Studio會嘗試找出錄製時與執行時，伺服器回應內容的差異部分，藉以找出需要關聯的資料，並建立關聯。

Rule Correlation

請依照以下步驟使用 Rule Correlation：

1. 啓用 auto-correlation
 - i. 點選 VuGen 的【Tools】>【Recording Options】，開啓【Recording Options】對話視窗，選取【Internet Protocol】>【Correlation】，勾選【Enable correlation during recording】，以啓用自動關聯。
 - ii. 假如錄製的應用系統屬於內建關聯規則的系統，如 AribaBuyer、BlueMartini、BroadVision、InterStage、mySAP、NetDynamics、Oracle、PeopleSoft、Siebel、SilverJRunner 等，請勾選相對應的應用系統。
 - iii. 或者也可以針對錄製的應用系統加入新的關聯規則，此即為使用者自訂的關聯規則。
 - iv. 設定當 VuGen 偵測到符合關聯規則的資料時，要如何處理：
 - 【Issue a pop-up message and let me decide online】：跳出一個訊息對話視窗，詢問您是否要建立關聯。
 - 【Perform correlation in sceipt】：直接自動建立關聯
2. 錄製腳本

開始錄製腳本，在錄製過程中，當 VuGen 偵測到符合關聯規則的資料時，會依照設定建立關聯，您會在腳本中看到類似以下的腳本，此為 BroadVision 應用系統建立關聯的例子，在腳本註解部分可以看到關聯前的資料為何。



3. 執行腳本驗證關聯是 OK 的。

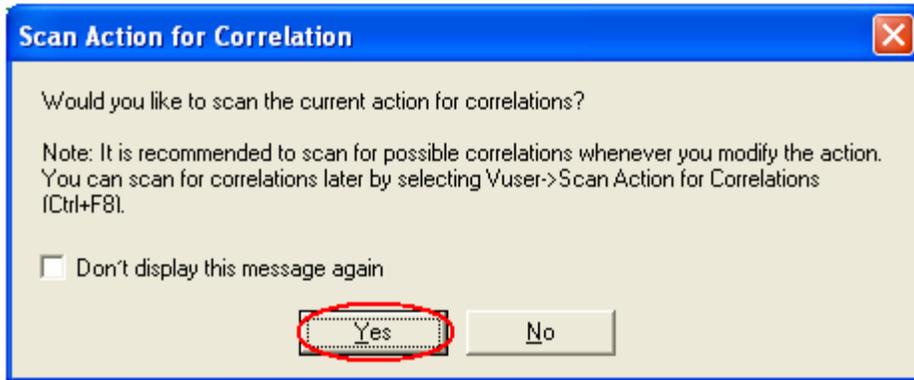
Correlation Studio

當錄製的應用系統不屬於 VuGen 預設支援的應用系統時，Rule Correlation 可能既無法發揮作用，這時可以利用 Correlation Studio 來做關聯。

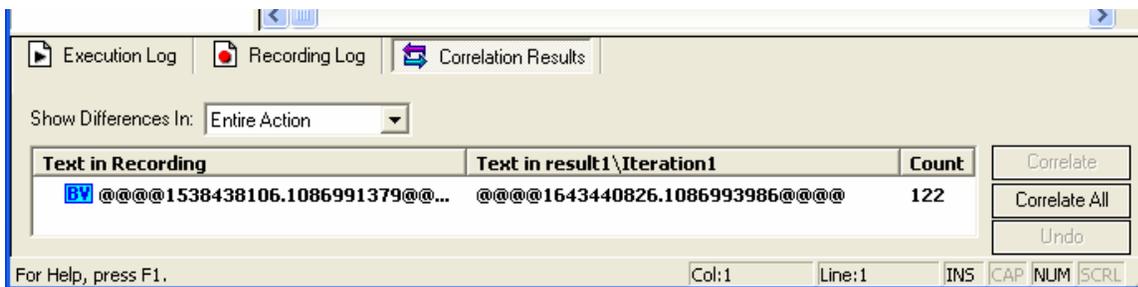
Correlation Studio 會嘗試找出錄製時與執行時，伺服器回應內容的差異部分，藉以找出需要關聯的資料，並建立關聯。

使用 Correlation Studio 的步驟如下：

1. 錄製腳本並執行
2. 執行完畢後，VuGen 會跳出下面的【Scan Action for Correlation】視窗，詢問您是否要掃描腳本並建立關聯，按下【Yes】按鈕。



3. 掃描完後，可以在腳本下方的【Correlation Results】中看到掃描的結果。



4. 檢查一下掃描的結果後，選擇要做關聯的資料，然後按下【Correlate】按鈕，一筆一筆做，或是按下【Correlate All】讓 VuGen 一次就對所有的資料建立關聯。
注意：由於 Correlation Studio 會找出所有有變動的資料，但是並不是所有的資料都需要做關聯，所以不建議您直接用【Correlate All】。
5. 一般來說，您必須一直重複步驟 1~4 直到所有需要做關聯的資料都找出來為止。因為有時前面的關聯還沒做好之前，將無法執行到後面需要做關聯的部份。

有可能有些需要做關聯的動態資料，連 Correlation Studio 都無法偵測出來，這時您就需要自行做手動關聯了。

4.2.2 手動關聯

手動關聯的執行過程大致如下：

1. 使用相同的業務流程與資料，錄製二份腳本
2. 使用 WinDiff 工具協助找出需要關聯的資料
3. 使用 `web_reg_save_param` 函數手動建立關聯
4. 將腳本中有用到關聯的資料，以參數取代

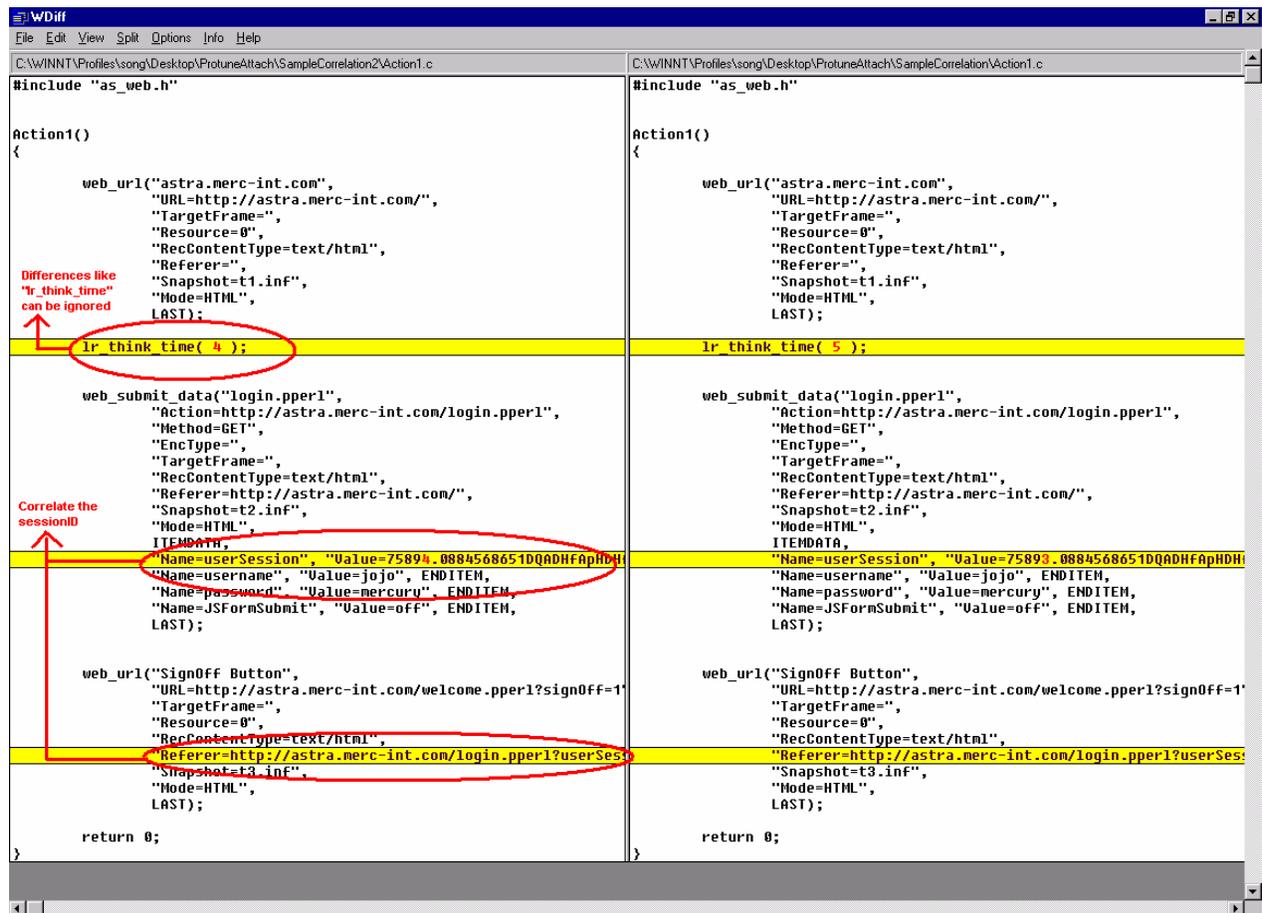
接下來將詳細的說明如何執行每個步驟

4.2.2.1 使用相同的業務流程與資料，錄製二份腳本

1. 先錄製一份腳本並存檔。
2. 依照相同的操作步驟與資料錄製第二份腳本並存檔。注意，所有的步驟和輸入的資料一定都要一樣，這樣才能找出由伺服器端產稱的動態資料。
有時候會遇到真的無法使用相同的輸入資料，那您也要記住您使用的輸入資料，到時才能判斷是您輸入的資料還是變動的資料。

4.2.2.2 使用 WinDiff 工具協助找出需要關聯的資料

1. 在第二份腳本中，點選 VuGen 的【Tools】>【Compare with Vuser...】，並選擇第一份腳本。
2. 接著 WinDiff 會開啓，同時顯示二份腳本，並顯示有差異的地方。WinDiff 會以一整行黃色標示有差異的腳本，並且以紅色的字體顯示真正差異的文字。（假如沒看到紅色字體，請點選【Options】>【View】>【Show Inline Differences】）。
3. 逐一檢視二份腳本中差異的部份，每一個差異都可能是需要做關聯的地方。選取差異的腳本，然後複製。
在複製時，有時並不需要取整行腳本，可能只會選取腳本中的一部分。
注意：請忽略 lr_think_time 的差異部份，因為 lr_think_time 是用來模擬每個步驟之間使用者思考延遲的時間。



4. 接著要在 Recording Log（單一 protocol）或是 Generation Log（多重 protocol）中找這個值。將滑鼠游標點到 Recording Log 的第一行開頭，按下 Ctrl+F，開啓【Find】視窗，貼上剛剛複製的腳本，找出在 Recording Log 第一次出現的位置。

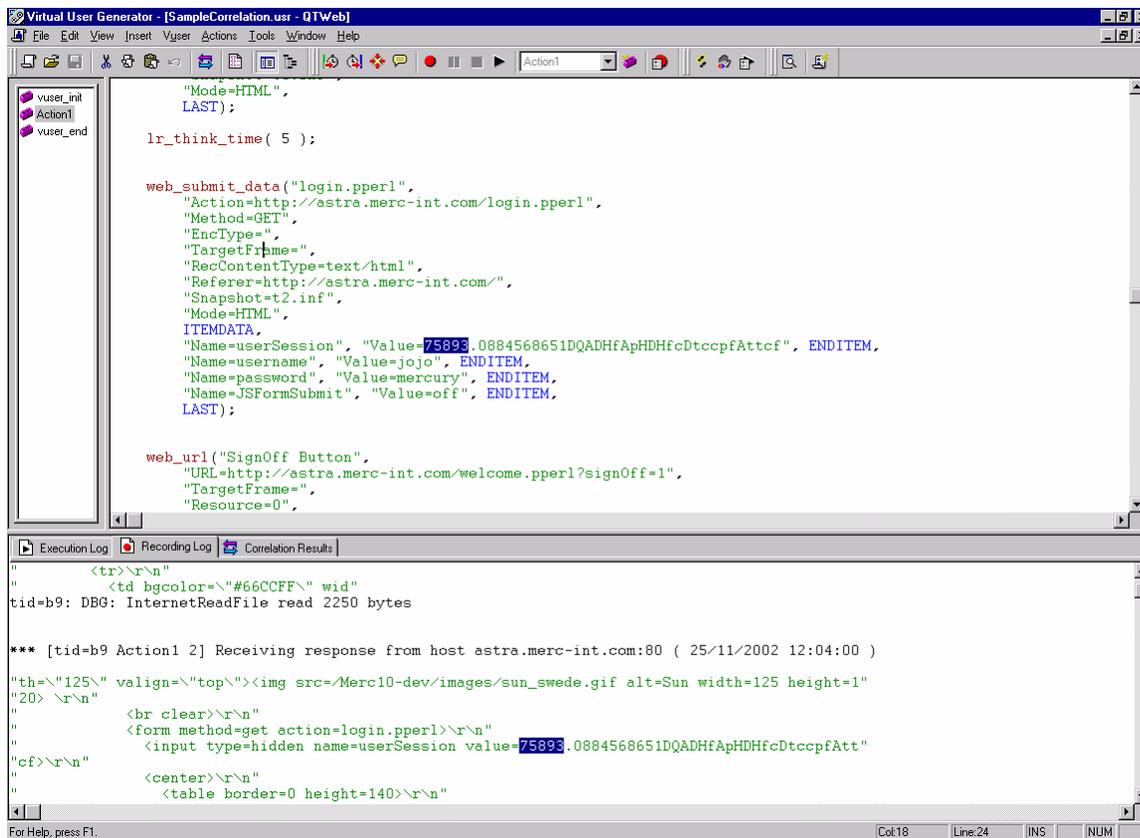


結果會有二種：

- 在 **Recording Log** 中找不到要找的資料，這時請先確認您找對了腳本，畢竟現在開啓了二個幾乎一樣的腳本，很容易弄錯。
- 在 **Recording Log** 中找到了要找的資料，這時要確認資料是從伺服器端傳送過來的。首先可以先檢查資料的標頭，從標頭的 **Receiving response** 可以知道資料是從伺服器端傳送到 client 端的。假如此資料第一次出現是在 **Sending request** 中，則表示此資料是由 client 端產生，不需要做關聯，但是有可能需要做參數化 (parameterized)。

您要找的標頭格式如下：

```
*** [tid=b9 Action1 2] Receiving response from host astra.merc-int.com:80 ( 25/11/2002 12:04:00 )
```



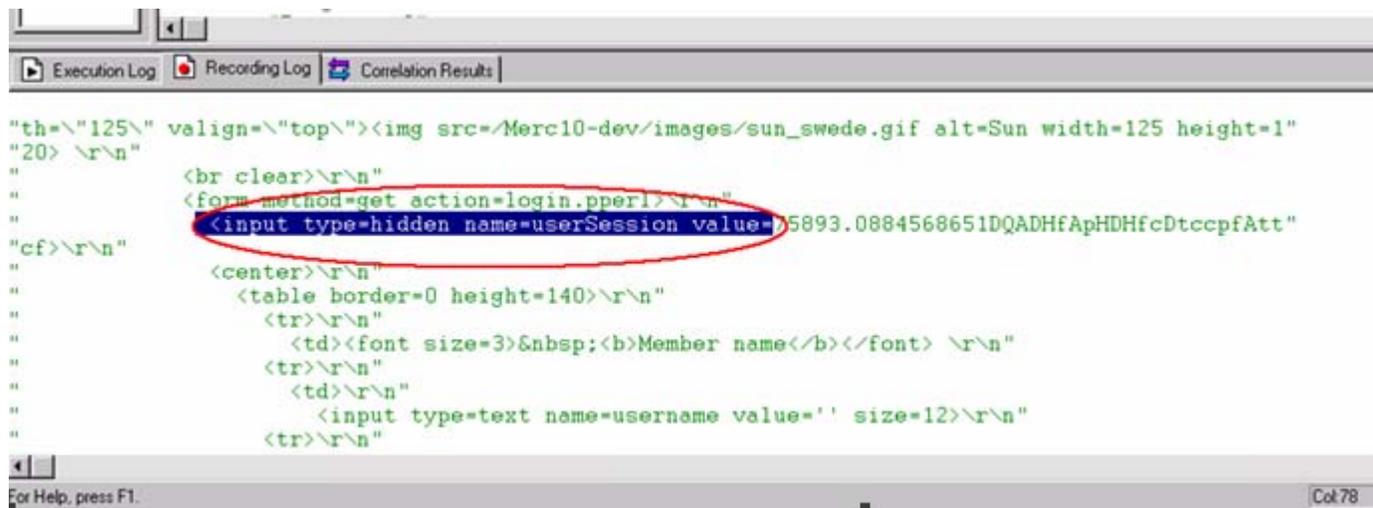
5. 現在您已經找到錄製二次都不一樣，而且是由伺服器所產生的動態資料了，而此資料極有可能需要做關聯。

4.2.2.3 使用 web_reg_save_param 函數手動建立關聯

在找到是由伺服器所產生的動態資料之後，接下來要做的就是找出適當的位置，使用 web_reg_save_param 函數，將這個動態資料擷取到某個參數中。

1. 要在哪裡使用 web_reg_save_param 函數？

在之前的步驟，我們已經在 Execution Log 找到可能需要關聯的動態資料。在 Execution Log 中選取動態資料前的文字然後複製，我們將會利用這段文字，來幫助我們找出要關聯的動態資料。



```
Execution Log | Recording Log | Correlation Results  
"th=\"125\" valign=\"top\"><img src=/Merc10-dev/images/sun_swede.gif alt=Sun width=125 height=1"  
"20> \r\n"  
" <br clear>\r\n"  
" <form method=get action=login.pperl>\r\n"  
" <input type=hidden name=userSession value=75893.0884568651DQADHfApHDHfcDtccpfAtt"  
"cf>\r\n"  
" <center>\r\n"  
" <table border=0 height=140>\r\n"  
" <tr>\r\n"  
" <td><font size=3>&nbsp;<b>Member name</b></font> \r\n"  
" <tr>\r\n"  
" <td>\r\n"  
" <input type=text name=username value='' size=12>\r\n"  
" <tr>\r\n"
```

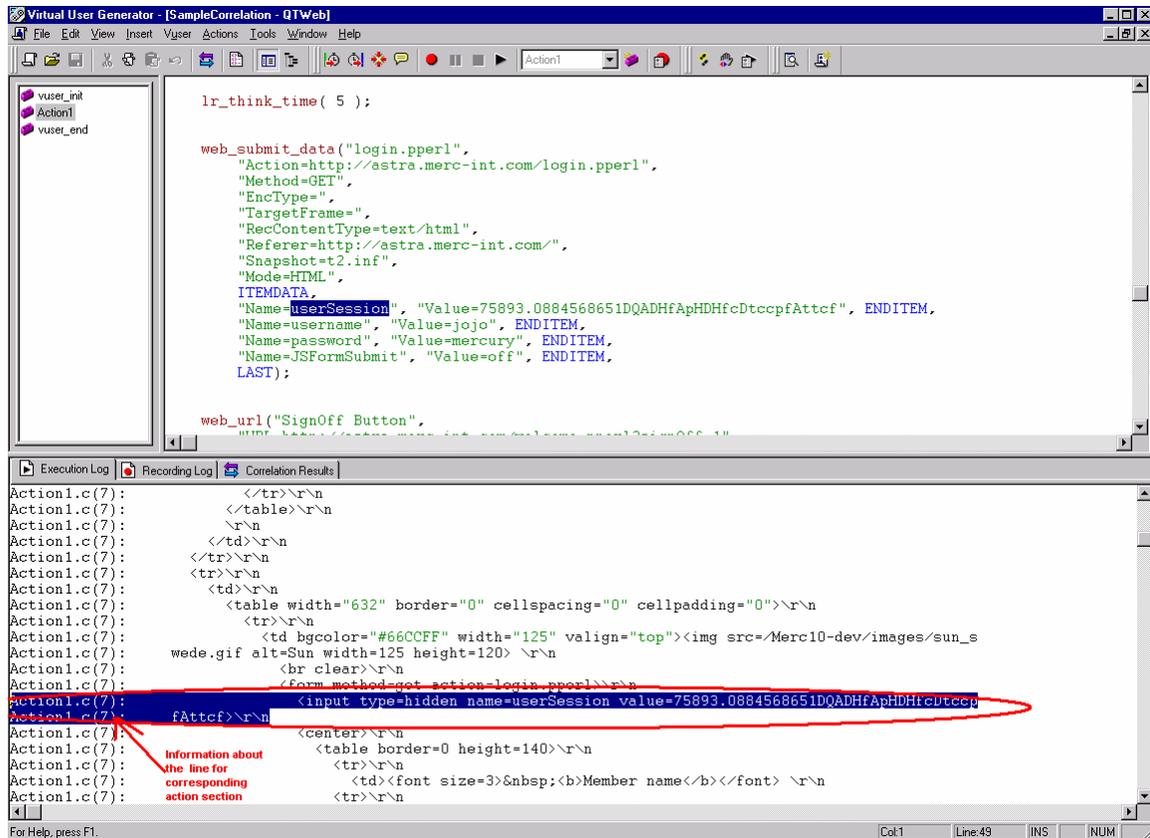
不過在這之前我們要先找出使用 web_reg_save_param 函數的正確位置，所以我們要再重新執行一遍腳本，而且這次會開啓所有的 Log。

- 在 VuGen 中點選【Vuser】>【Run-Time Settings】。
- 點選【General】>【Log】。
- 勾選【Enable logging】、【Always sends messages】、【Extended log】，以及【Extended log】下的所有選項。
- 按下【OK】就可以執行腳本了。

執行完腳本之後，在 Execution Log 中搜尋剛剛複製的字串。找到字串後，在字串前面會有 A.tion1.c(7)，這個 7 就是到時候要插入 web_reg_save_param 函數的位置，也就是要插入到腳本的第 7 行。

在腳本的第 7 行前插入一行空白行，然後輸入 web_reg_save_param("UserSession"， "UserSession" 這個 "UserSession" 就是到時要使用的參數名稱，建議給個有意義的名字。

注意：到這裡整個 web_reg_save_param 函數還沒完成。



2. 找出 web_reg_save_param 中要用到的邊界

web_reg_save_param 函數主要是透過動態資料的前面和後面的固定字串，來辨識要擷取的動態資料的，所以我們還需要找出動態資料的邊界字串。

找出左邊界字串

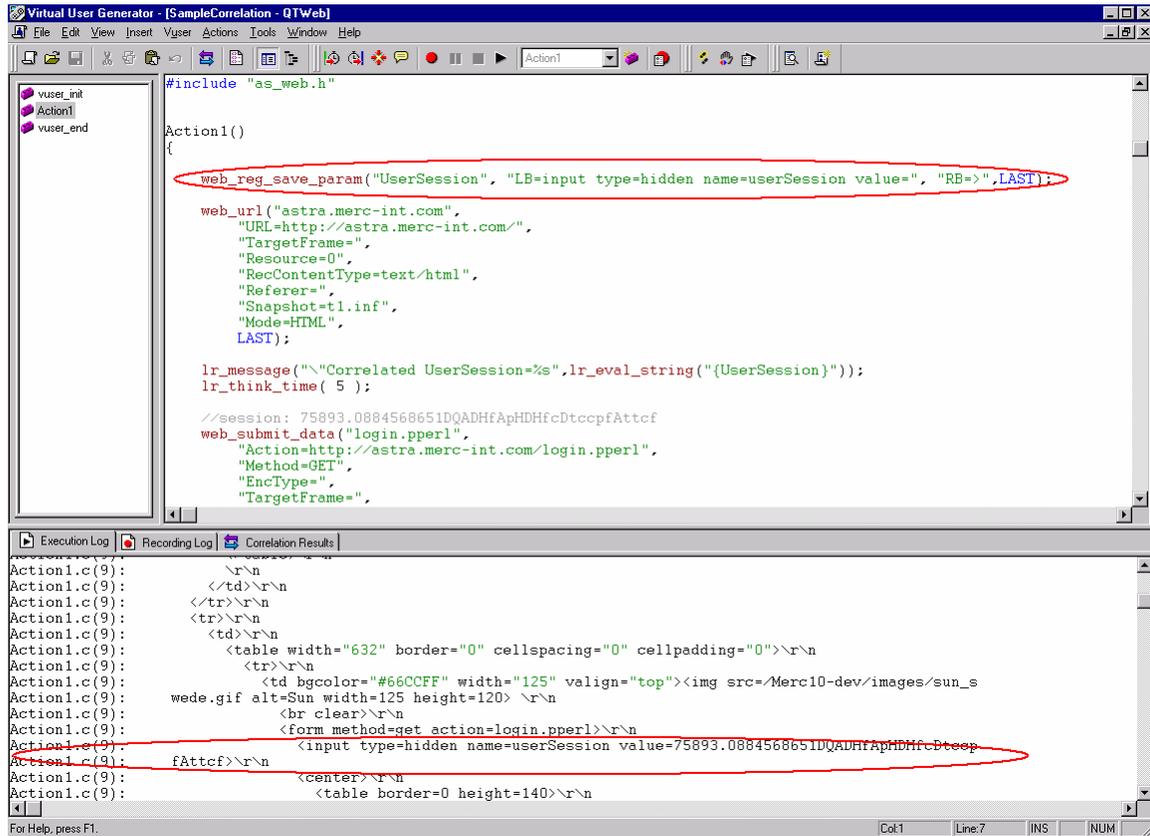
再回到 Execution Log 中，選取動態資料前的字串並且複製它。
 這時會有個問題，到底要選取多少字串才足以唯一識別要找的動態資料呢？建議是越多越好，但是盡量不要包含到特殊字元。
 在這邊我們選取「input type=hidden name=userSession value=」字串。選好之後，還要再確認一次這段字串真的是可以唯一識別的，所以我們在 Execution Log 中透過 Ctrl+F 的搜尋，找找看這段字串是否可以找到要找的動態資料。假如找不到，web_reg_save_param 函數還有個 ORD 參數可以使用，ORD 參數可以設定出現在第幾次的字串才是要找的字串。
 將這個邊界字串加到未完成的 web_reg_save_param 函數中：
 web_reg_save_param("UserSession", "LB= input type=hidden name=userSession value=",

找出右邊界字串

接下來要找出動態資料的右邊界字串，這個字串就比較好找了，從動態資料的最後一個字元開始，通常就是我們要找的右邊界字串了。

以這個例子來看，就是「>」，所以再把右邊界字串加入，web_reg_save_param 函數中，這時 web_reg_save_param 函數已經快完成了。最後再加上「LAST);」就完成整個 web_reg_save_param 函數了。

web_reg_save_param("UserSession", "LB= input type=hidden name=userSession value=", "RB=>", LAST);



4.2.2.4 將腳本中有用到關聯的資料，以參數取代

當使用 `web_reg_save_param` 建立參數後，接下來就是用“UserSession”參數去取代腳本中寫死的（hard-coded）資料。

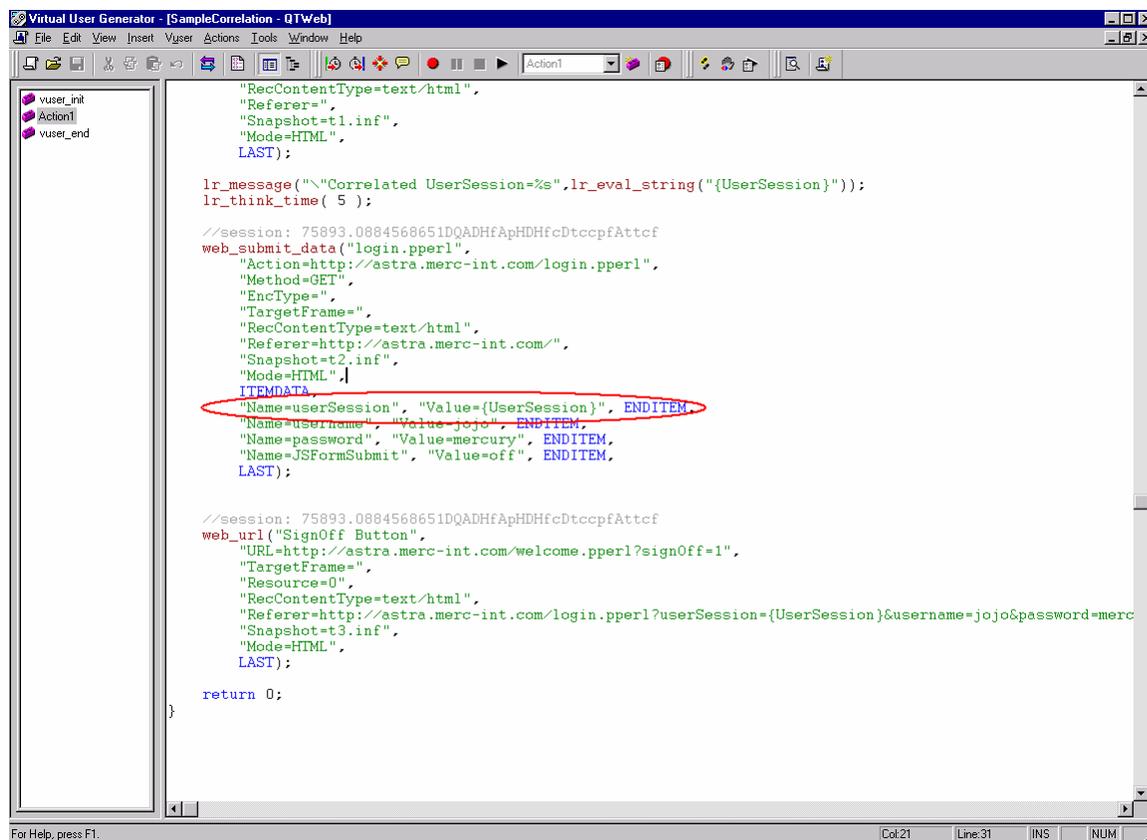
範例：

將

“Name=userSession”, “Value=75893.0884568651DQADHfApHDHfcDtccpfAttcf”, ENDITEM,

換成

“Name=userSession”, “Value={UserSession}”, ENDITEM,



```
Virtual User Generator - [SampleCorrelation - QTWeb]
File Edit View Insert Vuser Actions Tools Window Help

"RecContentType=text/html",
"Referer=",
"Snapshot=t1.inf",
"Mode=HTML",
LAST);

lr_message("\\"Correlated UserSession=%s",lr_eval_string("{UserSession}"));
lr_think_time( 5 );

//session: 75893.0884568651DQADHfApHDHfcDtccpfAttcf
web_submit_data("login.pperl",
"Action=http://astra.merc-int.com/login.pperl",
"Method=GET",
"EncType=",
"TargetFrame=",
"RecContentType=text/html",
"Referer=http://astra.merc-int.com/",
"Snapshot=t2.inf",
"Mode=HTML",
ITEMDATA
"Name=userSession", "Value={UserSession}", ENDITEM
"Name=username", "Value=jojo", ENDITEM,
"Name=password", "Value=mercury", ENDITEM,
"Name=JSFormSubmit", "Value=off", ENDITEM,
LAST);

//session: 75893.0884568651DQADHfApHDHfcDtccpfAttcf
web_url("SignOff Button",
"URL=http://astra.merc-int.com/welcome.pperl?signOff=1",
"TargetFrame=",
"Resource=0",
"RecContentType=text/html",
"Referer=http://astra.merc-int.com/login.pperl?userSession={UserSession}&username=jojo&password=merc",
"Snapshot=t3.inf",
"Mode=HTML",
LAST);

return 0;
}
```

到這裡您已經完成了一個關聯了，接下來就是執行腳本，是否能成功運行，假如還是有問題，就要檢查看看是否還需要再做另一個關聯。

5. 建立關聯時常用的小技巧

5.1 如何顯示參數值？

有時你會想知道參數的值是什麼，這時你可以使用 `lr_eval_string` 與 `lr_output_message` 這二個函數來做到。例如：

```
lr_output_message("Value Captured = %s", lr_eval_string("{ParameterName}"));
```

`lr_eval_string` 與 `lr_output_message` 函數的使用說明請參考 [LoadRunner Online Function Reference](#)。

5.2 關聯時如何處理特殊字元？

5.3 在腳本的 `data` 目錄下找不到路製時的快照 (snapshot)

造成在腳本的 `data` 目錄下找不到路製時的快照 (snapshot) 的可能原因如下：

- 腳本是由 **VuGen 6.02** 或更早的版本所錄製的
- 匯入的 **Action** 不會包含快照 (snapshot) 的檔案
- 腳本是儲存在唯讀的目錄下，早成 **VuGen** 無法儲存執行時擷取的快照 (snapshot)
- 某些步驟並不會產生快照 (snapshot)，如瀏覽某個資源
- 快照 (snapshot) 功能被取消
【Tools】 > 【General options】 > 【Correlation】 tab > 【Save correlation information during replay】

5.4 開啓 WinDiff 時出現「File no longer available」的錯誤訊息

WinDiff 這個工具有些限制，無法開啓包含空白字元的目錄或是腳本，所以建議命名時不要使用空白字元，並且儘可能將名稱取短一點。

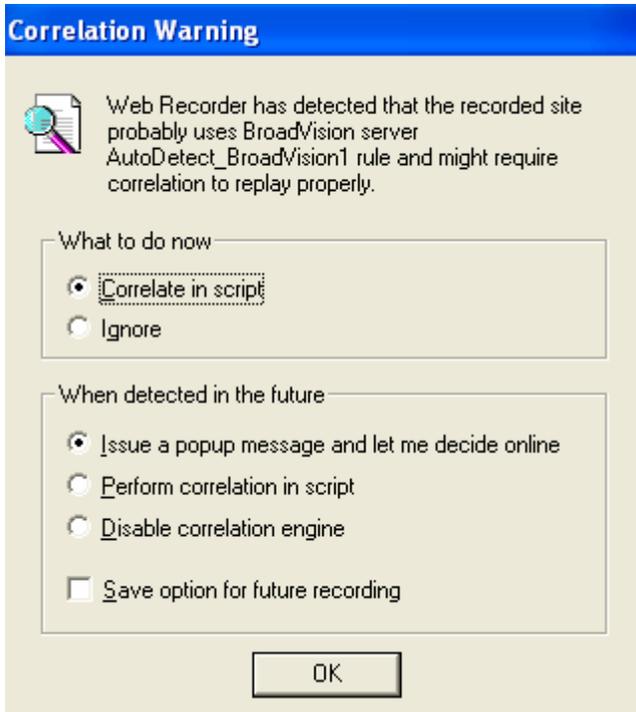
5.5 執行腳本時出現「No match found for the requested parameter 'ParameterName'. If the data you want to save exceeds xx bytes, use web_set_max_html_param_len to increase the parameter size.」的錯誤訊息

5.6 錄製時突然跳出【Correlation warning】對話視窗

當你有勾選自動關聯的【Issue a popup message and let me decide online】選項，當 **VuGen** 發現有可能要做關聯的資料時，就會跳出【Correlation warning】的視窗，詢問你要做關聯 (Correlation in script) 還是要忽略 (Ignore)。

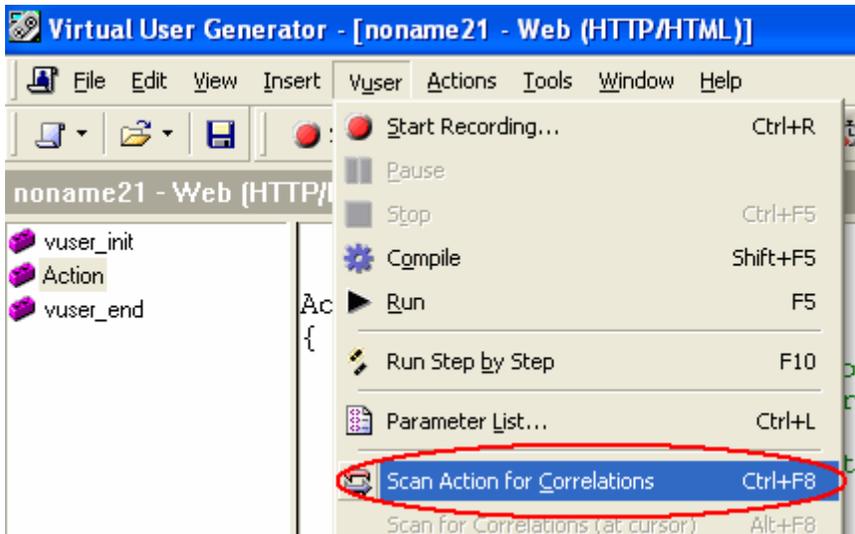
另外你也可以勾選【Perform correlation in script】，讓 **VuGen** 自動作關聯，不會再跳出詢問視窗。

或是勾選【Disable correlation engine】，關閉自動關聯的功能。



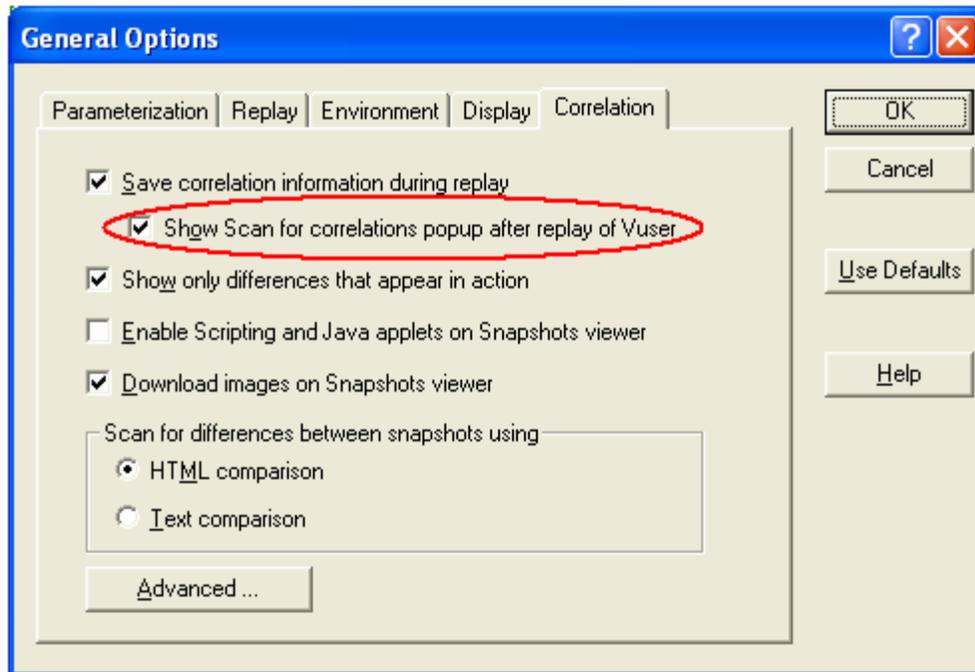
5.7 如何手動啟動「Scan action for correlation」的功能

要手動啟動「Scan action for correlation」的功能，請先執行腳本一次後，點選【Vuser】>【Scan Action for Correlation】。



5.8 執行完腳本後並未出現【Scan Action for Correlation】視窗

要啟用【Scan Action for Correlation】功能，請點選【Tools】>【General options】>【Correlation】tab，勾選【Show Scan for correlation popup after replay of Vuser】選項。



5.9 為什麼 Correlation Studio 無法偵測到所有需要做關聯的動態資料？

6. 其它關於關聯的資源