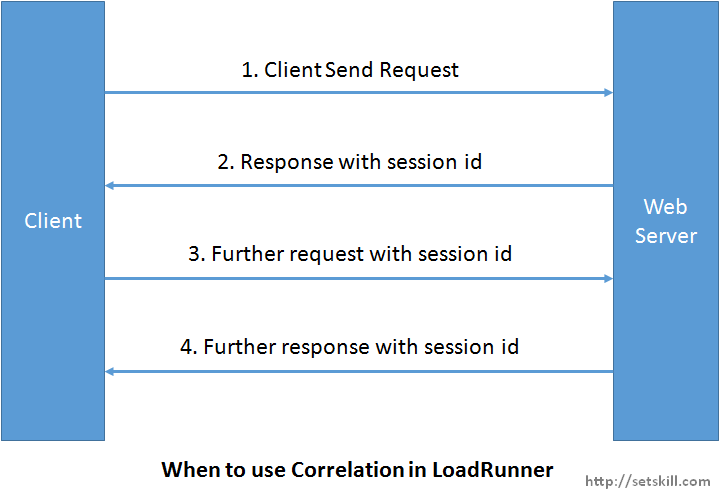
**What is Correlation? Benefits and Examples**

In addition to parameters, dynamic handling of data is also done via **Correlation in LoadRunner**. This is normally done for web application that return data such as session data to the client. It is not possible to do this via parameters because of the dynamic nature of the value returned to the client.

In this tutorial you will learn:

1. [Why do we need correlation](http://setskill.com/hp-loadrunner/what-is-correlation-benefits-and-examples/#correlation_in_loadrunner)
2. [Enabling Automatic Correlation](http://setskill.com/hp-loadrunner/what-is-correlation-benefits-and-examples/#automatic_correlation)
3. [Manual Correlation](http://setskill.com/hp-loadrunner/what-is-correlation-benefits-and-examples/#manual_correlation)

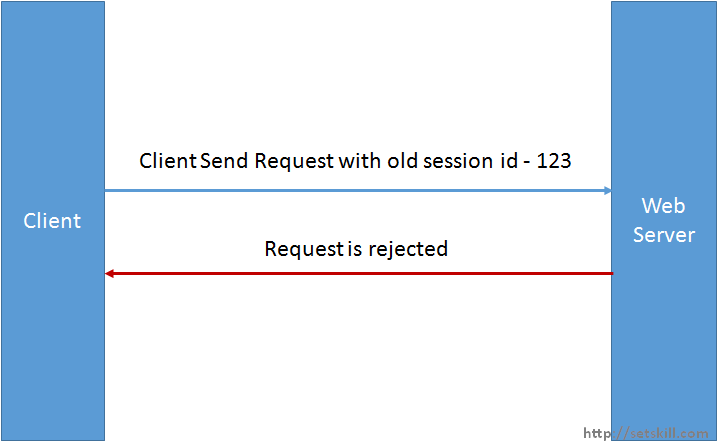
**Why do we need correlation?**

To better understand correlation, let’s look at a classical example of what happens when a user accesses a simple web application.

* First the client browser sends a request to a web server for a particular web page
* Normally the web server will send a response to the client along with a value known as a session ID. Let’s assume that the session ID is 123. This session ID helps the web server keep track of which client is making the request.
* Now whenever the client makes another request to the web server in the next short duration, the session ID will be passed to the server.
* So now when the server see’s the session ID of 123, it will send further responses to the server.

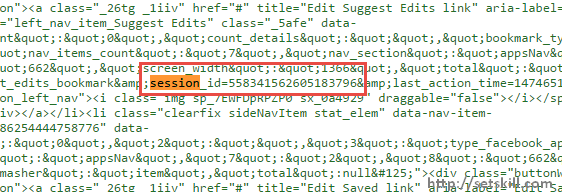
So now let’s consider the following notes with the above steps

* When we record our script in load runner, it will record the session ID as 123.
* Session ID normally timeout after a time interval of 20 to 30 minutes, depending on the site’s configuration.

So now with the above mentioned facts, let’s assume we play back our script after 30 minutes with the above scenario.

When the load test script makes a request with the recorded session id of 123, the web server will not able to validate the request because the session id would have expired. And so in this case, the load test scripts would start failing.

In such cases, we need to make our load test script smart enough to ask the web server for a new session id and then incorporate the same session id in subsequent requests. This is what we call correlation in Load runner.

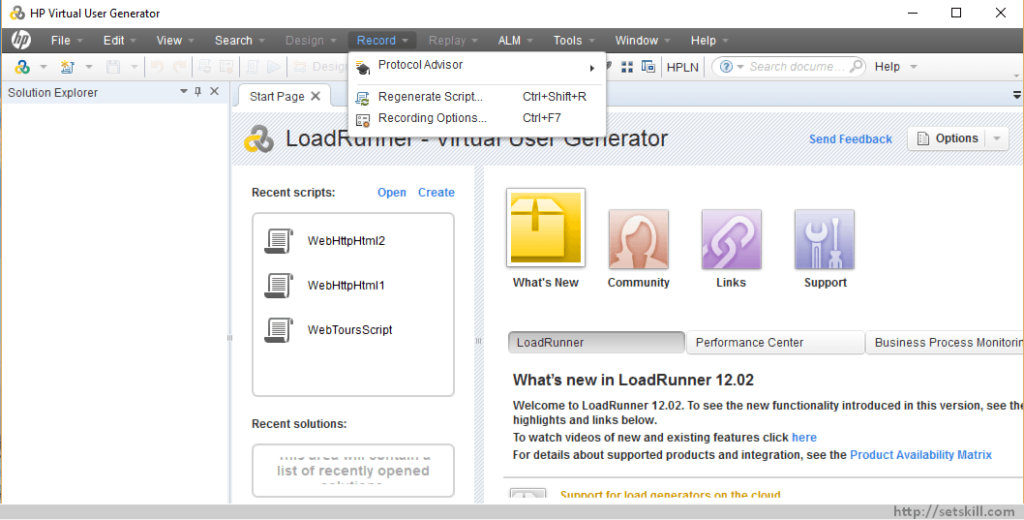
If you make a request to open the Facebook page and then view the source of the Facebook application and search for the session\_id variable, you will actually be able to see the session id. An example is shown below.

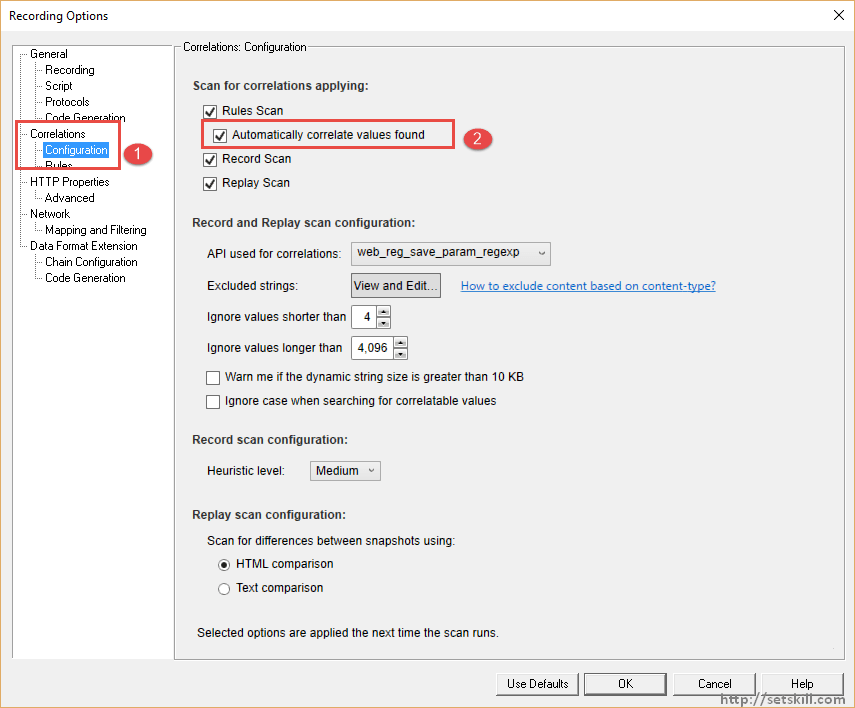
**Enabling Automatic Correlation in LoadRunner**

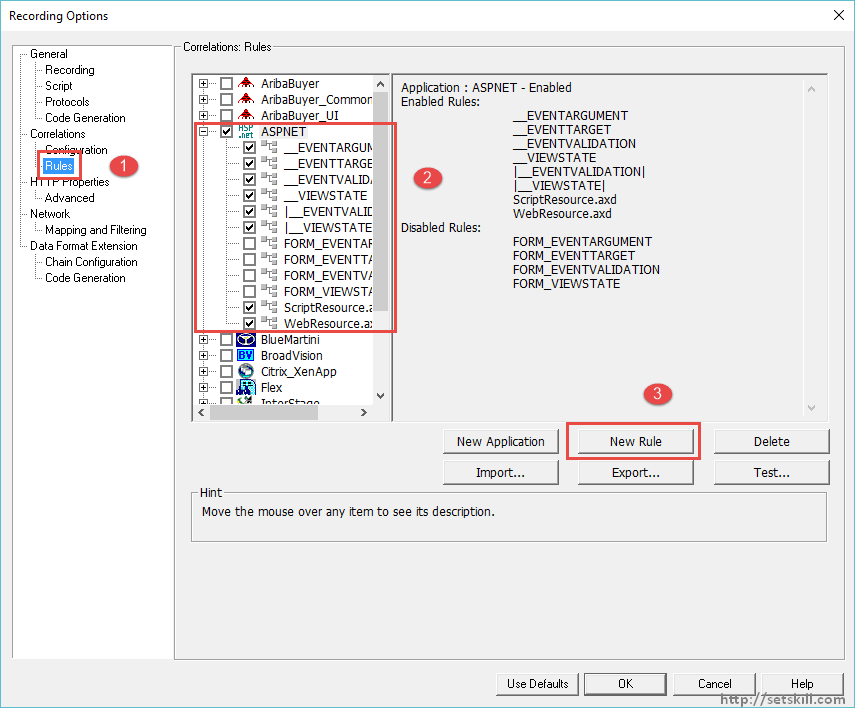
In order to Correlation to work you need to enable this feature in the VUGEN editor.

Automatic Correlation is where you give the job to the VUGEN editor to understand how to automatically perform the correlation. But please note that you need to validate your script multiple times to ensure it works when you perform automatic correlation. The correlation will happen when the script is recorded.

Perform the below mentioned steps to have this setup.

Step 1) From the menu option choose the Record->Recording Options.

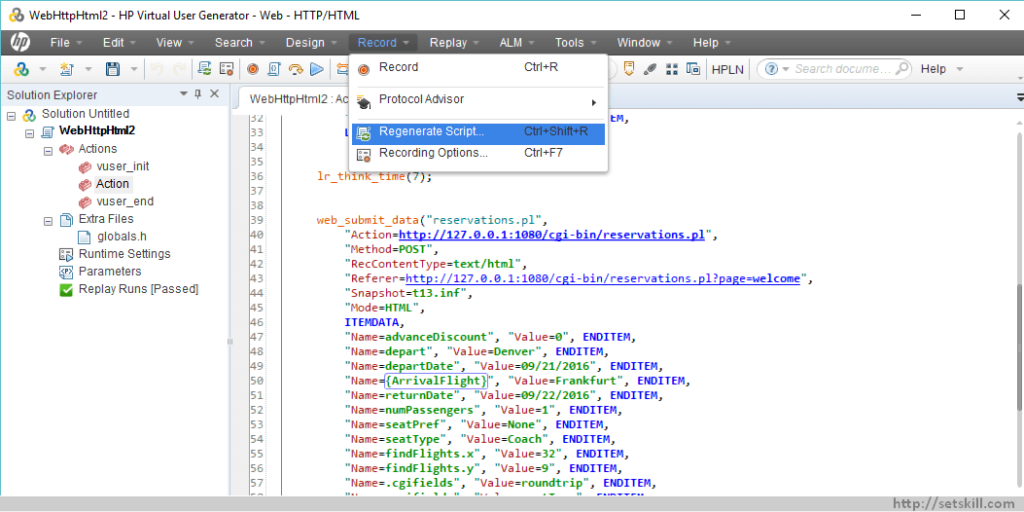
Step 2) On the left hand side, go to Correlations->Configuration. Make sure that the checkbox for ‘Automatically correlate values found’ is checked.

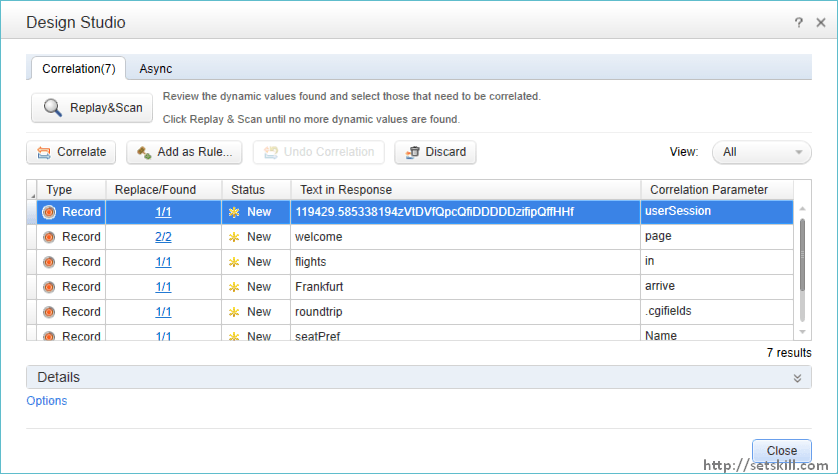
Step 3) You then need to go to the Rules section and enable the rules for the required technology.

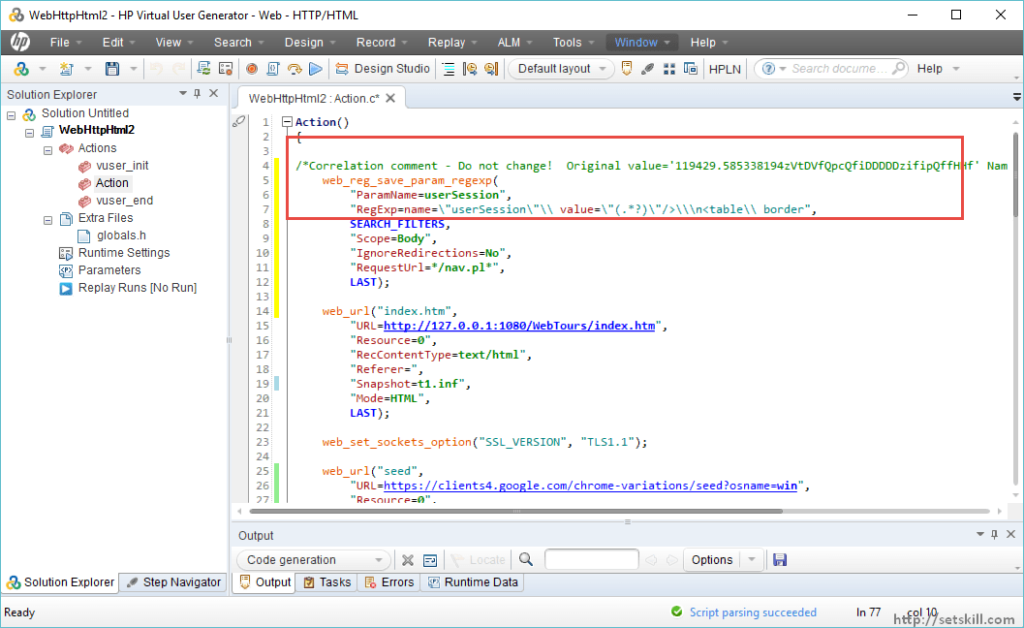
* Go to the Rules section
* As an example, we are going to enable the rules for the ASP.Net technology. Here we can enable or disable the subset of rules for the technology.

In ASP.Net Viewtstate is a dynamic parameter which keeps on changing based on the web server. So in the above case, we are saying to enable the rule which allows the script to correlate all the Viewstate values in the script when generated.

* You can also create rules if required by using the ‘New’ rule button.

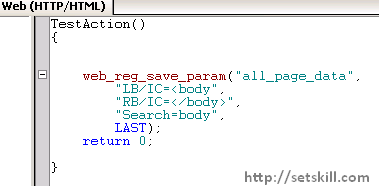
Suppose if you already have an existing script, you can use the regenerate script option after setting the required correlation options.

You will then be prompted for all possible correlations found in the script. Click on each row and click on the Correlate button and then press the Close button to proceed.

You will then see the correlation made in the script. There will be comments inserted, so that it will be easier for you to see where all the correlations have been made.

**Manual Correlation in LoadRunner**

For Manual correlation, you need to perform the correlation yourself. For this you need to understand how to understand how to extract the data yourself. This is more complex for a tester to do, but can lead to better results when running scripts.

A snippet of how a manual correlation is shown below, where you yourself would need to determine what is the parameter name and parameter value to be inserted into the script.

The general steps in performing manual correlation is given below

1. Understand how the value to be replaced is constructed. For example session id will normally be as session\_id=value&
2. Find the right and left delimters, so in the case of sesion\_id, to get the value, you search for ‘=’ as the beginning delimiter and “&” as the end delimiter of the value.
3. Add a web\_reg\_save\_param function to the script with the parameter name as session\_id and value as the extracted value.
4. Parameterize the dynamic value in the script every time it occurs in the script.
5. Verify the script works properly

The **web\_reg\_save\_param** **function** has the following arguments

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Object | An expression evaluating to an object of type WebApi |
| ParamName | A null terminated string indicating the name of the parameter to create. |
| List of Attributes | 1) Convert  The possible values are: HTML\_TO\_URL: convert HTML encoded data to a URL encoded data format HTML\_TO\_TEXT: convert HTML encoded data to plain text format  2) IgnoreRedirections  If “IgnoreRedirections=Yes” is specified and the server response is redirection information  3) LB  The left boundary of the parameter or the dynamic data  4) NOTFOUND  The handling option when a boundary is not found and an empty string is generated.  5) RB  The right boundary of the parameter or the dynamic data  6) RelFrameID  The hierarchy level of the HTML page relative to the requested URL  7) SaveLen  The length of a sub-string of the found value, from the specified offset, to save to the parameter  8) SaveOffset  The offset of a sub-string of the found value, to save to the parameter.  9) Search  The scope of the search where to search for the delimited data |