Load Runner

“LoadRunner” is a performance testing tool. Loadrunner has a variety of protocol support that covers nearly all types of software platforms.

# History

* Loadrunner was developed by Mercury Interactive
* HPE acquired LoadRunner, as part of its acquisition of Mercury Interactive in November 2006 and named HP LoadRunner.
* On 1 September 2017, the announcement of the merger of Micro Focus with HPE gave it a name called “Micro Focus LoadRunner”,
* In Jan 2023 OpenText acquired Micro Focus and became the product of OpenText.

# How loadrunner works

LoadRunner works on the concept of recording and replaying the user activities and generating the desired load on the server. It simply simulates the real-world user’s actions and helps to identify the performance of the software application or system by generating a virtual load. The main steps include:

1. **Recording/Scripting:** To capture the user action into a script.
2. **Test Execution:**Replaying the script along with the virtual load to simulate the real-world situation in the test environment.
3. **Result Analysis:** To provide accurate results in terms of load handling capacity and responsiveness of the application.

LoadRunner simulates real user activities in the form of scripts (programs) and runs those scripts by generating virtual users (threads/processes). These virtual users are known as ‘Vusers’. During performance test execution, Vusers run concurrently and generate the traffic on the server. On the completion of the test, LoadRunner collates the results and saves them in a file (called Raw results). This file can be opened through the Microfocus Analysis tool and performed further analysis of the test result. In the end, the Analysis tool generates the report (in pdf, HTML, excel etc. format) which concludes the test result.

# Loadrunner components

Micro Focus LoadRunner has the following components:

* **Virtual User Generator** or **VuGen:**It records end-user business processes and creates an automated test script in the form of a programming language. The generated script is called a ‘VuGen Script’ or ‘Test Script’.
* **Load Generators:** These are machines which generate the virtual load as per the requirement. During test execution, the Controller distributes each Vuser in the scenario to a load generator. A load generator is also used to simulate the geographical location of the users.
* **Controller:** It organizes, drives, manages, and monitors the load test.   
  Another function of a controller is to command the Load Generators via Agent during the test execution which involves
  + Which script needs to pick?
  + How much load needs to generate and at what time?
  + When to stop the test? etc.
* **Agent:**The LoadRunner agent sets up communication between the Controller and the Load Generator.
* **Analysis:**It helps to view, dissect, and compare the results of the load tests. The analysis tool displays the result graphs and statistics to validate the test result against the defined NFRs (Non-functional requirement) and to find out the bottleneck (performance bugs). The analysis tool also generates a report based on the test result.

# Loadrunner test process

# Creating Testscripts using VuGen

About VuGen

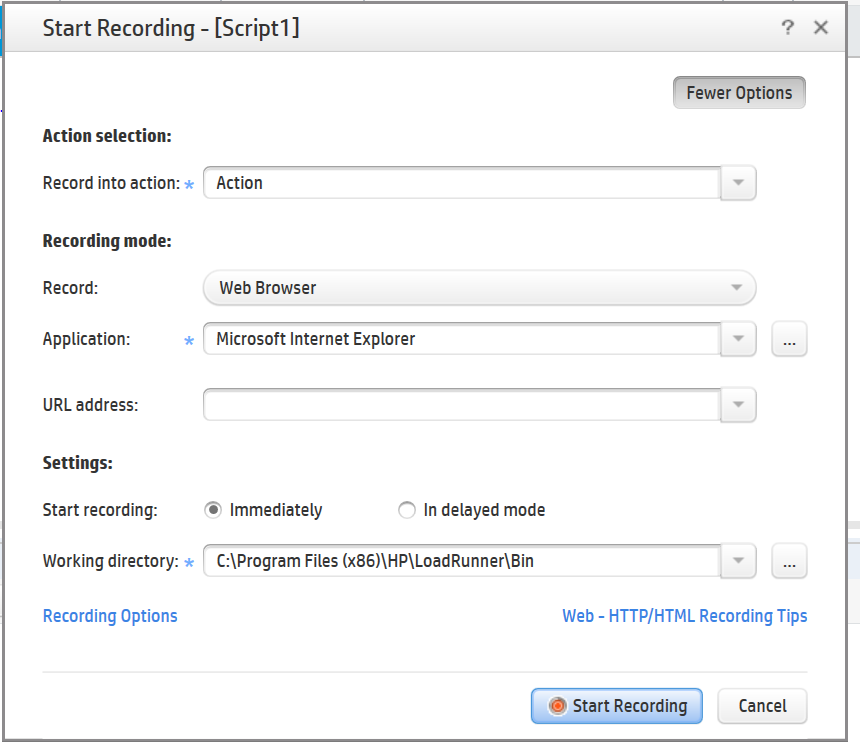
1. Click on the + button to create a new test
2. Select the protocol depending on the application type. We can either select a single protocol or multiple protocol
3. Give a name and location of the script
4. Create a script
5. Loadrunner generates the scripts in C language
6. Three actions are created by default

* **Actions**
* **Vuser init** – Iterates at the beginning of the test( like a before annotation)
* **Vuser\_end** – Executed once at the end of the test

1. Only the action part will be iterated and vsuer\_init cannot be iterated
2. Similarly, Vuser\_end will be executed once at the end of the script.
3. All the header files are saved in globals.h
4. Vugen will prepare the script but the performance test execution will be done by the controller

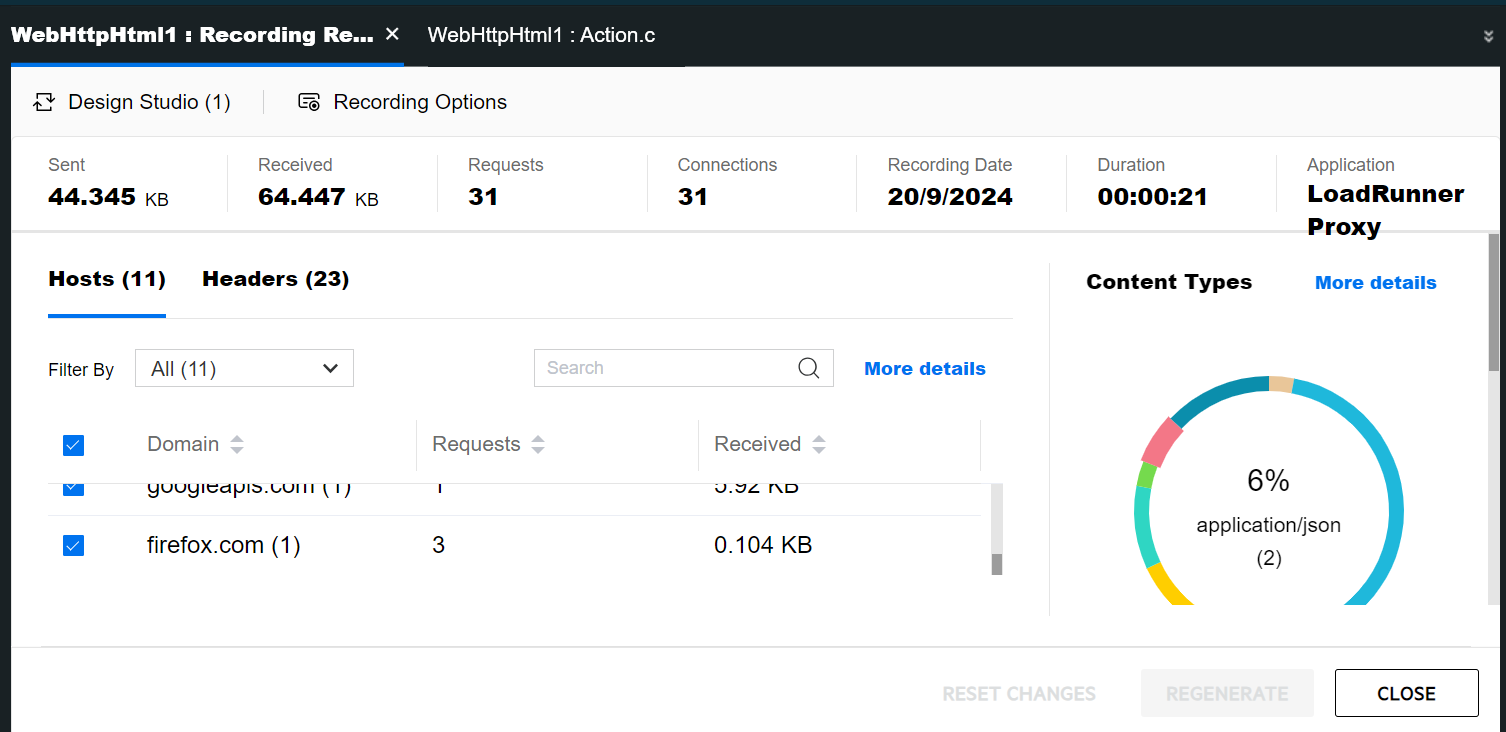
## Recording a Test

1. Click on the record button
2. Select the browser
3. Enter the application URL
4. Start the recording
5. Record the steps
6. Stop the recording
7. Note the script generated



https://petstore.octoperf.com/actions/Catalog.action

### Recording Report



# Different Actions in Loadrunner

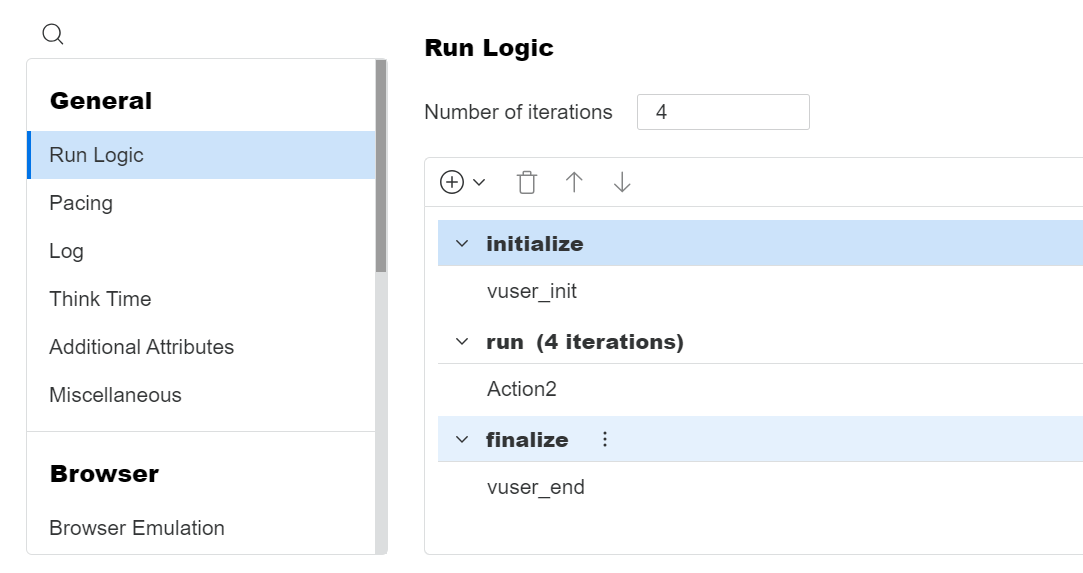
Types of actions

1. Actions
2. VUser\_init 🡪 Initialize action
3. VUser\_end 🡪 Ending action like closing application/Close connections

Recording in different actions

## Running Iterations

Run iterations are given in Runtime settings. Only the actions are iterated. Init and end are executed only once.





# Recording Settings

## General Settings

* HTML based scripts – Application side scripts
* URL Based scripts – Server side scripts

## Script related settings

* Close all AUT processes when recording stops
* Add think time- default 3 seconds
* Generated recorded events log – default added, by selecting this option we can add additional logs in the output
* Think time greater than threshold – If the think time is more than 3 seconds only then the time is captured.
* Maximum number of lines in action – If it exceeds the mentioned threshold, it will automatically add a new action for the remaining lines
* Warning – If the application faces issues

## Protocols

This is selected when a new script is created. Here we can see the details of the protocol(s) selected during script creation.

## HTTP Properties

There are settings related to the protocol selected. If more protocols are selected during script creation then more tabs will appear in the settings.

## Mapping and Filtering

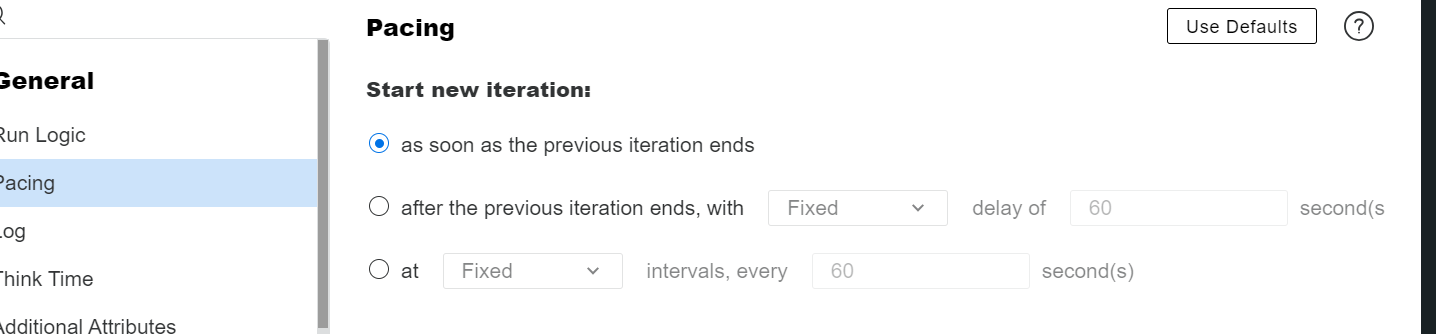
Here we can give server and port details of the application so VuGen is aware of the application details that is being recorded.

## Chain Config

Easier scripting of web applications by providing the ability to encode and decode the formatted data. I,e. Exchange between the client and the server.

# Runtime Settings

* Run Login – Set number of iteration, Insert new action
* Pacing – Gap between the iterations



Delay – starts after the first iteration ends

Interval – Gives a fixed delay after the start of first iteration

* Logs – Capturing logs Always or During error
* Think time – Time taken by the user while recording. How much time user took to select option
  + Ignore think time
  + Replay think time as recorded
  + Multiply recorded think time
  + Use random think time percentage of recorded think time
  + Limit think time

# HTML Recording vs URL Recording

A famous LoadRunner Web HTTP/HTML protocol is used in 70% of performance testing scenarios. This protocol is used for web-based applications and record client and server communication. Web HTTP/HTML protocol has two recording modes which are HTML mode and URL mode. Each mode has its own features and advantages.

HTML-based mode, records script for every user action that is performed while URL-based mode records each and every browser request to the server and resources received from the server.

HTML based mode records as you perform clicks and doesn’t give you inside information like what is happening behind the recording while URL based mode records each and every step and emulate Javascript code.

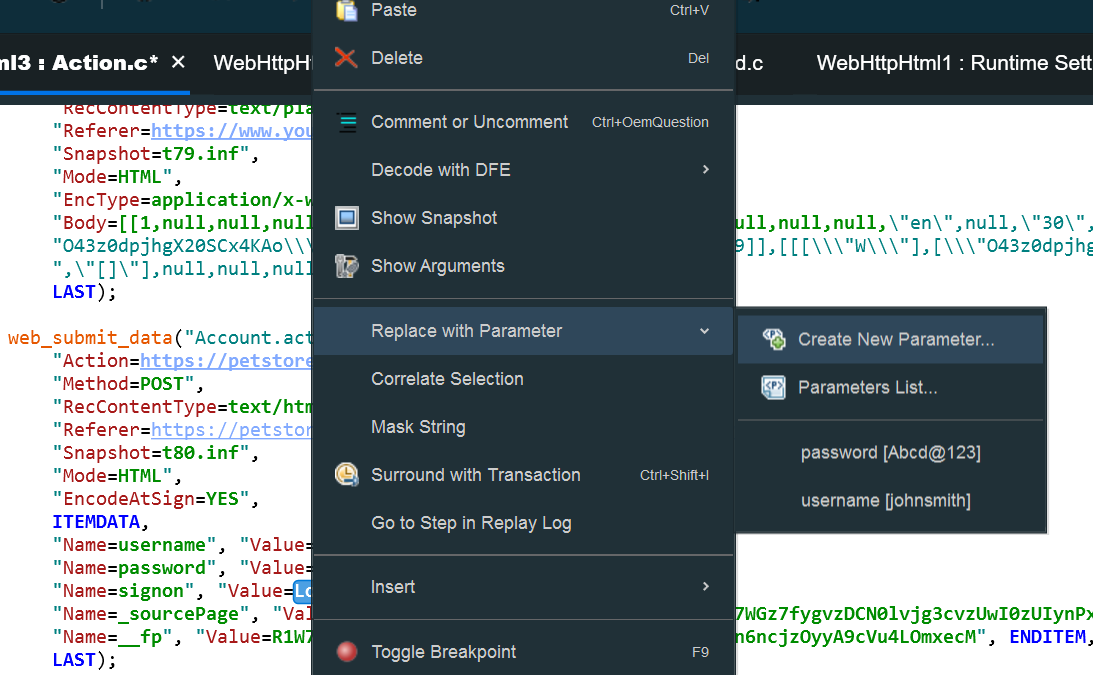
1. HTML mode would have less correlation to do while URL mode has much more complex correlation requirements.
2. HTML mode is smaller and is more intuitive to read as the statements are inside the functions corresponding to the user action performed. In the case of URL based, all statements get recorded into web\_url()
3. HTML mode is recommended for browser applications *while* URL mode is recommended for non-browser applications.

# Parameterization

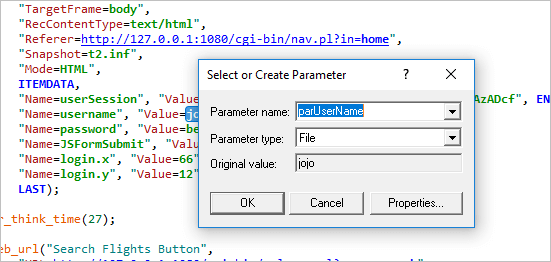
Parameterization is replacing the hard-coded values with parameters

If we want to parameterize the username value in our script, select the value, right-click and select **‘Replace with Parameter’ -> ‘Create new parameter’**.

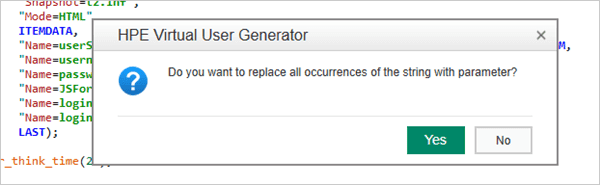
Parameters can also be added from design tab.



Enter the parameter name. We can enter any name here



Next, VuGen will prompt for this.



If we Click ‘Yes’. VuGen will replace all occurrences of the username value with the parameter.

* In order to view the parameter(s) that we created, click on ‘Parameters’ item on the ‘Solution Explorer’.
* It can also be viewed from the Design tab.
* This will open the parameters list window. This list will show the username parameter that we created.
* The parameter name is ‘parUserName’ (the parameter name that we entered), the parameter type is ‘File’ (this is the default parameter type) and the parameter file name is ‘parUserName.dat’ (parameters are generally stored in a text file that is saved in the script folder, by default the name of this text file is the parameter name itself).
* One more thing to note is VuGen by default creates a new text file for each of the parameters created.
* We can also save more than one parameter value in a single text file by putting them under different columns separated by a delimiter
* We can also provide the order in which data must be taken.
* Number of data rows need not match with number of iteration.
* The data taken at run time can be viewed in Runtime data tab.

