1ICPC317

AY 2024-25

SDLC Laboratory

Quality Laboratory Manual

Experiment No. 10

To perform performance testing using performance testing tool

Course Instructor –

Mr. Sharanabasava Raddi
ASSISTANT PROFESSOR

Experiment No. 10

Title of Experiment: To perform performance testing using performance testing tool

Aim of Experiment: To understand and conduct performance testing using the tool

System Requirements – Win 10 and above OS, 4GB RAM, 2.33 GHz Processor,

Software/s Requirement –

Load Runner or Apache JMeter

Experiment Objectives:

- Assess speed
- Ensure scalability
- Test stability
- Identify bottlenecks
- Validate load handling
- Support capacity planning
- Benchmark performance
- Verify SLA compliance
- Improve user experience

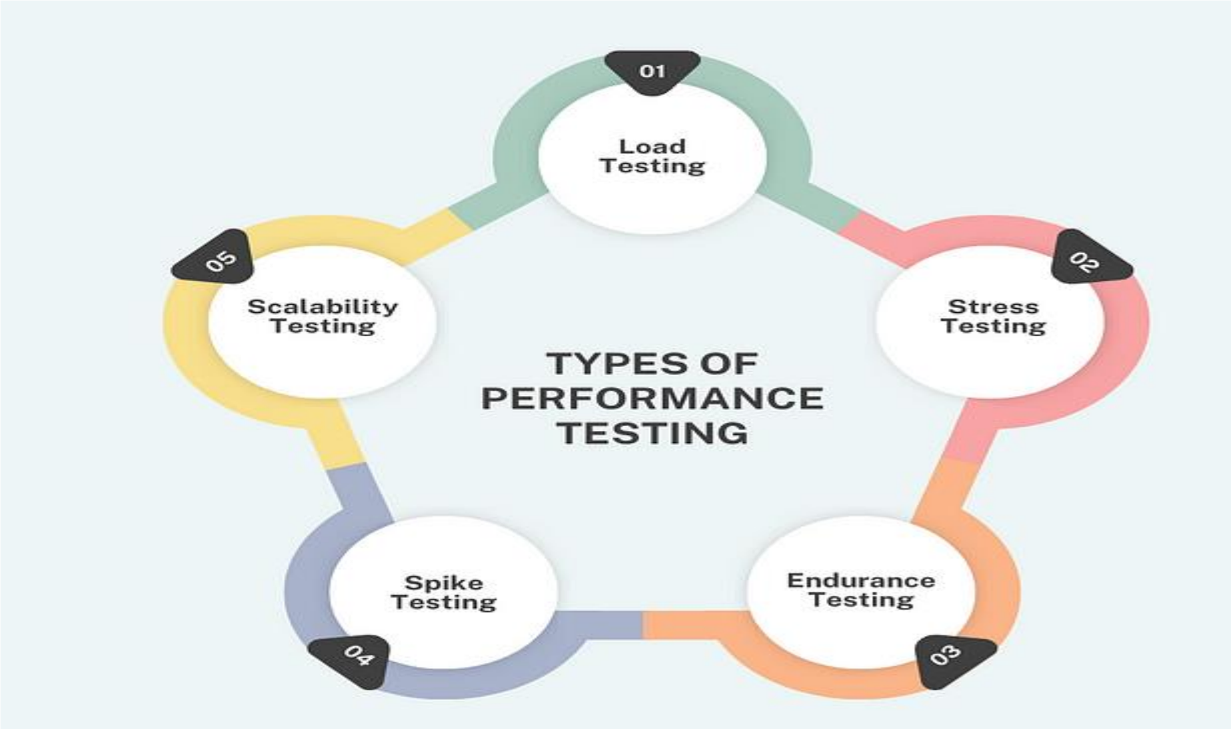
Experiment Outcomes:

- Response time metrics
- Throughput levels
- Resource utilization data (CPU, memory, disk, network)
- System scalability insights
- Bottleneck identification
- Stability under load confirmation
- Capacity limits

Theory:

Performance testing involves evaluating the speed, responsiveness, and stability of a software application under a particular workload. The primary goal is to ensure that the application performs well under normal and peak conditions, providing a better experience for end-users. Performance testing helps in identifying performance bottlenecks and tuning the application to meet desired performance standards. at a time, hence the focus of the tester is on the internal design of the application as well.

Types of Performance Testing:



- **Load Performance Testing:** Load performance testing is a critical process that involves simulating multiple users accessing an application simultaneously to evaluate how it handles expected traffic volumes. This type of testing helps identify performance bottlenecks, latency issues, and resource utilization problems under normal operating conditions. By understanding how an application performs under load, developers can optimize code, improve infrastructure, and ensure the system can handle peak usage periods without degradation in performance
- **Stress Performance Testing:** Stress performance testing pushes the application beyond its normal operational capacity to see how it handles extreme conditions. This type of testing helps identify the maximum capacity of the application, uncovering potential weaknesses and failure points that may not be evident under regular usage. By understanding how the application behaves under stress, developers can make necessary adjustments to enhance its robustness and ensure it can handle unexpected spikes in traffic or data load.
- **Beyond Performance Testing:** Beyond performance testing includes other specialized types of testing that focus on specific performance aspects, such as:

1. **Endurance Testing:** Evaluates the application’s performance over an extended period to identify potential memory leaks and other issues that may appear with long term use.
2. **Spike Testing:** Tests how the application handles sudden spikes in user load, ensuring it can maintain performance during unexpected traffic surges.
3. **Scalability Testing:** Assesses the application’s ability to scale up or down in response to increased or decreased user loads.

Common Performance Problems:

Most performance problems revolve around speed, response time, load time, and poor scalability. Speed is often one of the most important attributes of an application. A slow-running application will lose potential users. Performance testing ensures an app runs fast enough to keep a user's attention and interest. Take a look at the following list of common performance problems and notice how speed is a common factor in many of them:

Long Load time – Load time is normally the initial time it takes an application to start. This should generally be kept to a minimum. While some applications are impossible to make load in under a minute, Load time should be kept under a few seconds if possible.

Poor response time – Response time is the time it takes from when a user inputs data into the application until the application outputs a response to that input. Generally, this should be very quick. Again if a user has to wait too long, they lose interest.

Poor scalability – A software product suffers from poor scalability when it cannot handle the expected number of users or when it does not accommodate a wide enough range of users. Load Testing should be done to be certain the application can handle the anticipated number of users.

Bottlenecking – Bottlenecks are obstructions in a system that degrade overall system performance. Bottlenecking is when either coding errors or hardware issues cause a decrease in throughput under certain loads. Bottlenecking is often caused by one faulty section of code. The key to fixing a bottlenecking issue is finding the section of code causing the slow down and trying to fix it there. Bottlenecking is generally fixed by either fixing poor running processes or adding additional Hardware. Some common performance bottlenecks are

CPU utilization

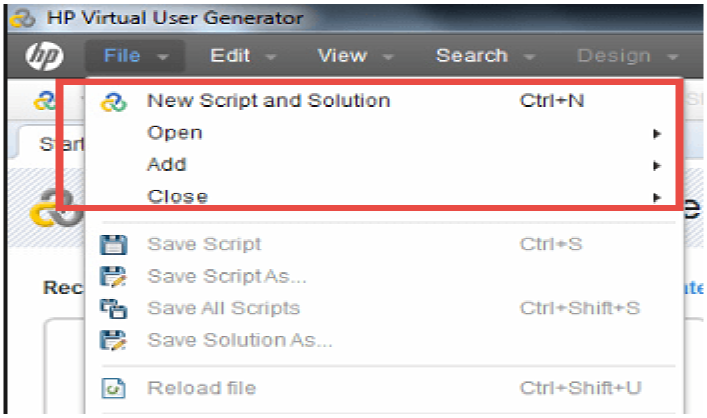
Memory utilization

Network utilization

Operating System limitations

Disk usage

VUgen in Load Runner:

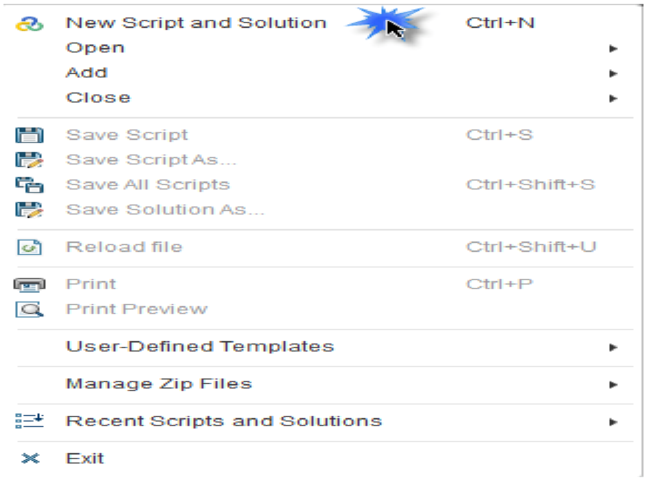


Before you record, make sure the Web Tours App server is running. Remember, you need to keep the server running so do not close it. If the window bothers you, you can minimize it.

In a real-world problem, one needs to get acquainted with the subject application (SUL) especially if it involves complex business workflows and data stages. You can also interact with HP Web Tours to ensure you can repeat the exact steps when recording.

Before one can start with scripting of any application (SUL), one needs to get acquainted with the application flow. With this tutorial, let’s get acquainted with HP Web Tours which is shipped and automatically installed with HP Load Runner.

Step 1) Click on File =>New Script and Solution as shown in the snapshot below:



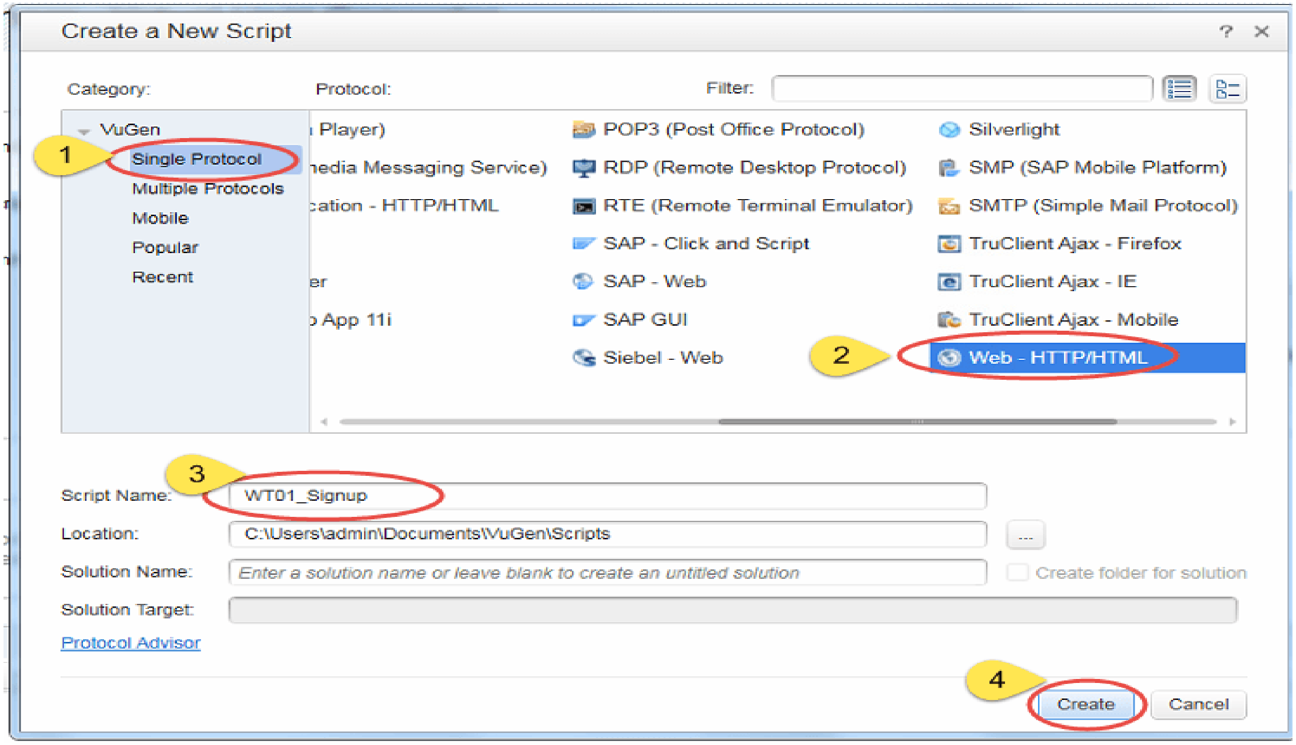
Step 2) A window to select the protocol is shown. The list looks slightly different from previous versions

1. Select Single Protocol

2. Web

3. Enter Script Name

4. Click Create



Create a New Script

Category: VuGen > Single Protocol

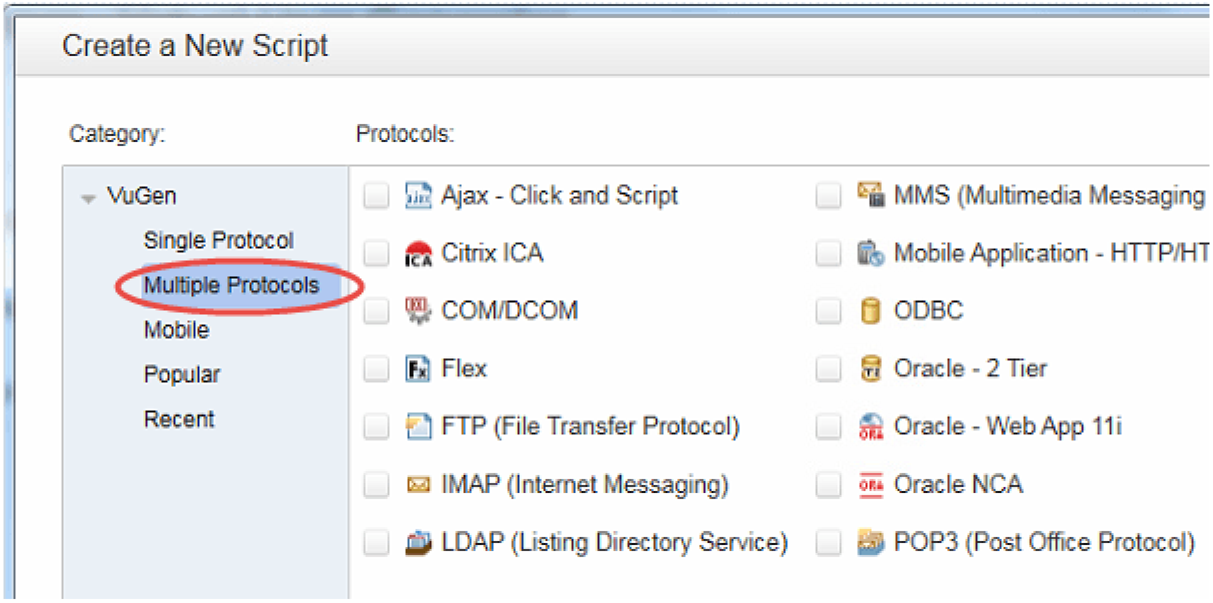
Protocol: Web - HTTP/HTML

Script Name: WT01_Signup

Location: C:\Users\admin\Documents\VuGen\Scripts

Solution Name: Enter a solution name or leave blank to create an untitled solution

Solution Target:



Create a New Script

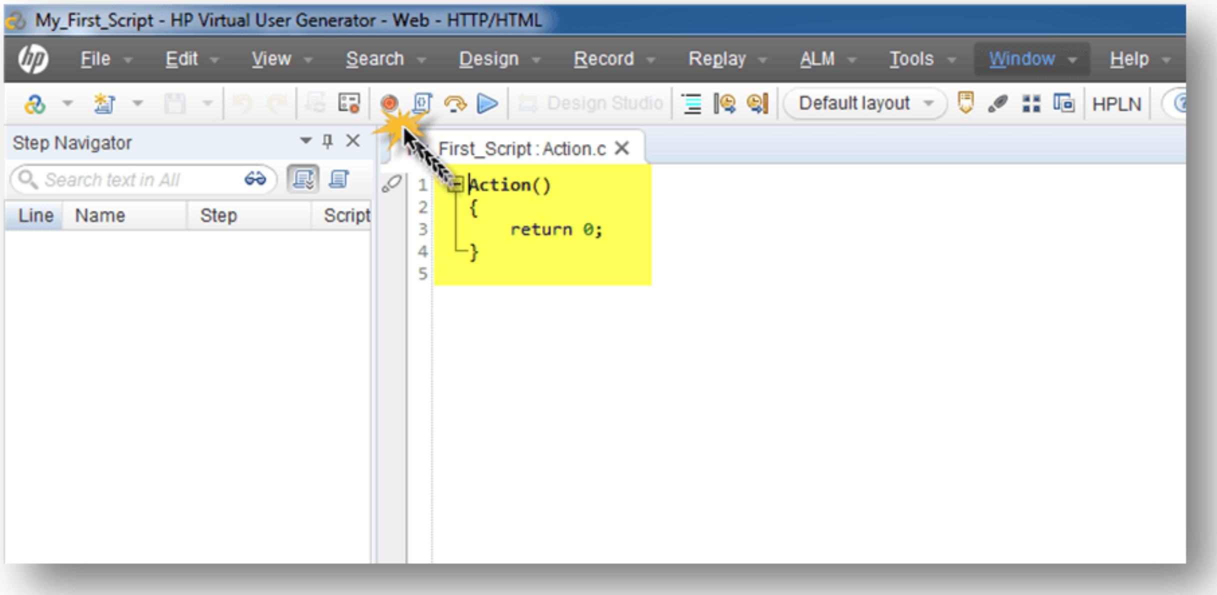
Category:

- VuGen
  - Single Protocol
  - Multiple Protocols
  - Mobile
  - Popular
  - Recent

Protocols:

☐ Ajax - Click and Script	☐ MMS (Multimedia Messaging
☐ Citrix ICA	☐ Mobile Application - HTTP/HT
☐ COM/DCOM	☐ ODBC
☐ Flex	☐ Oracle - 2 Tier
☐ FTP (File Transfer Protocol)	☐ Oracle - Web App 11i
☐ IMAP (Internet Messaging)	☐ Oracle NCA
☐ LDAP (Listing Directory Service)	☐ POP3 (Post Office Protocol)

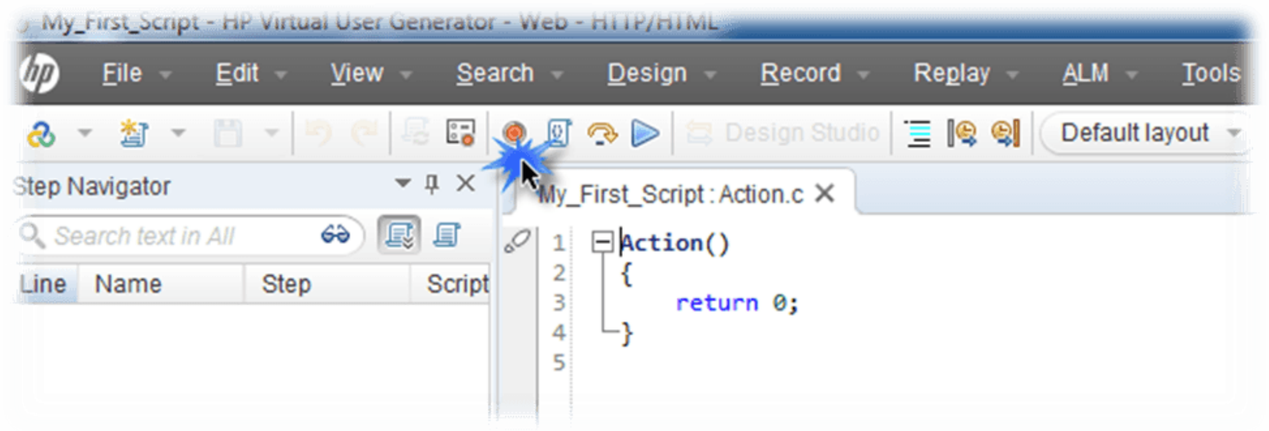
Step 3) once you click Create, HP VUGen will open the IDE (Integrated Development Environment) or code editor. You can notice the script files are blank, except basic signature of Function Action. Below snapshot should help understand it more:



My_First_Script - HP Virtual User Generator - Web - HTTP/HTML

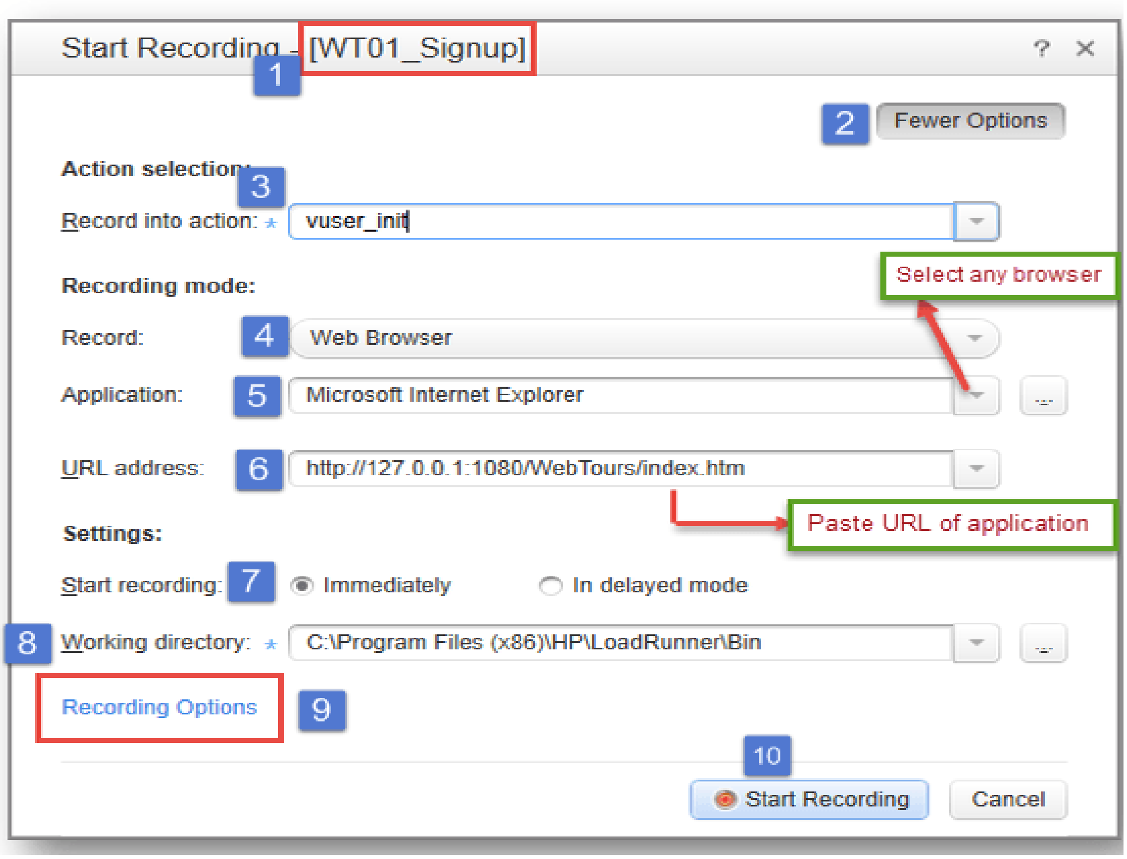
```
Action()
{
    return 0;
}
```

Step 4) Click the Recording button which you can find in the toolbar. Refer to the snapshot below:



```
Action()
{
    return 0;
}
```

Step 5) a new window opens



Start Recording - [WT01_Signup]

Action selection:

Record into action: * vuser_init

Recording mode:

Record: Web Browser

Application: Microsoft Internet Explorer

URL address: http://127.0.0.1:1080/WebTours/index.htm

Settings:

Start recording: ● Immediately ○ In delayed mode

Working directory: * C:\Program Files (x86)\HP\LoadRunner\Bin

Select any browser

Paste URL of application

Step 6) once recording starts, VUGen will invoke the HP Web Tours application using the browser selected.

Conclusion:

Performance testing is a very important aspect of software development, ensuring that applications can handle user loads and perform optimally under various conditions. By understanding the different types of performance testing and following best practices, developers can effectively identify and address performance issues, leading to a better user experience.

Expected Oral Questions:

1. What is Performance?
2. Performance testing is a functional testing or non-functional testing?
3. Mention any two tools used for performance testing?
4. What are the different testing performed under performance testing

FAQs in Interview:

1. What is the difference between functional and non-functional testing?
2. Why we need virtual user while performing performance testing?
3. What is co-relation in performance testing?
4. Differentiate between Load and Stress testing?