

# The TelQA Testing Series

## Part 1: The Performance Testing Process



## The TelQA Testing Series

This article is part of a series of documents which cover the basics of performance testing. See [www.telqa.com](http://www.telqa.com) for other articles in this series and additional downloadable testing resources.

### Overview

This article presents a general testing strategy that will help you create and deploy successful performance tests. Although testing methods must always be flexible enough to adapt to individual projects, having a general structure can be very helpful for both experienced testers and those just getting started.

## The Performance Testing Process

A typical performance test consists of four main stages:

1. Test Planning
  - Determine what you want to achieve from the test
  - Define the type of test(s) you are going to perform
  - Select a suitable test environment
  - Specify inputs (user types and number of users) and expected outputs (performance data collection, page timers, transaction timers etc.)
2. Test Development
  - Capture realistic user interaction for each user type and save as editable scripts.
  - Model captured scripts to create reusable virtual users (VUs)
  - Select appropriate performance monitors for target application
  - Create test using the defined proportion of VU types. The test should also include realistic load application, delays etc.
3. Test Deployment
  - Configure test environment
  - Verify correct operation of test environment
  - Run test
  - Clean up the test environment
4. Results Analysis
  - Analyse captured performance data to determine system performance level
  - Use captured performance data to assist in performance tuning or removing performance bottlenecks.

Let's have a look at these stages in a bit more detail.

## 1. Test Planning

### 1.1 What do you want to achieve from the test?

Determining the purpose of the test is a crucial part of the overall test planning, development and deployment process. Tests which are intended to demonstrate a level of acceptable performance may differ significantly from tests which are to be used as part of scalability planning.

### 1.2 What type of tests do you want to perform?

The type of test will be determined by what you want to achieve. Broadly speaking, there are two main types of performance test, load tests and stress tests. Within these areas there is a large range of more focused tests which target specific aspects of application performance.

### 1.3 Selecting a suitable test environment

The test environment should be representative of the production environment where possible. Any performance constraints caused by the test environment must be taken into the account during the results analysis stage.

### 1.4 Inputs versus outputs

Prior to developing the test it is important to define the user types and the proportion of each user type which constitute the anticipated production load. Likewise, the expected outputs must also be defined. These may include parameters which are visible to the user such as page download times and transaction times or other less obvious parameters such memory usage and database access times.

## 2. Test Development

### 2.1 Capture user interactions

The interactions of an individual user type are captured using a combination of a standard browser and test specific software. These interactions are then stored within an editable script.

### 2.2 Model the script to create a virtual user

A virtual user (VU) is created by modifying (or modelling) the script captured in the first stage of test development. These VUs are used to load the web application during the test run.

### 2.3 Selection of performance monitors

The performance of the application under test will be determined from the values collected by the selected performance monitors. These monitors may range from standard performance counters through to bespoke remote monitoring agents.

## 2.4 Creation of the test

A performance test involves the application of a load consisting of different combinations and volumes of VUs. These VUs must be applied in a realistic manner with appropriate delays during VU application and interaction.

## 3. Test Deployment

### 3.1 Configure the test environment

Where appropriate, the test environment will need to be configured prior to the performance test. This configuration may include the creation of users within databases, passwords etc.

### 3.2 Verify correct operation of the test environment

Before running the performance test, it is advisable to verify the correct operation of the test environment, e.g. user authorisation, passwords, drive-mapping etc.

### 3.3 Running the test

This is where the application under test is loaded with the realistic VU-based load. Wherever possible, the performance test should be run multiple times to obtain more accurate results and to determine whether there were any anomalous test results.

### 3.4 Clean up the test environment

After the test has run, the test environment must be restored to its state prior to the application of load. This may involve the removal of users from databases and any other data created during the test.

## 4. Results Analysis

### 4.1 Analyse captured performance data

The data collected by the performance counters can be compared against the performance acceptance criteria to determine whether the outcome of the test is successful.

### 4.2 Performance tuning and removing bottlenecks

Although performance testing is often used to determine whether the application has met a particular performance level, it is also commonly used to assist in performance tuning as well as the identification and removal of performance bottlenecks.

The areas covered in this part of the TelQA Testing Series are explored in more detail in later parts of the series:

## The TelQA Testing Series

### Part 1: The Performance Testing Process

Part 2: Performance Testing Aims

Part 3: Types of Performance Testing

Part 4: Performance Test Environments

Part 5: Performance Test Inputs and Outputs

Part 6: Capturing User Interactions

Part 7: Modelling Scripts

Part 8: Selecting Performance Monitors

Part 9: Performance Test Creation

Part 10: Configuring Performance Test Environments

Part 11: Verifying Performance Test Environments

Part 12: Running a Performance Test

Part 13: Analysing Performance Data

Part 14: Using Performance Data for Tuning and Bottleneck Removal