

PERFORMANCE EXECUTION TEST PLAN

TABLE OF CONTENTS

1. Introduction.....	4
1.1 Purpose of This Document.....	4
1.2 Document Overview.....	4
1.3 Scope of Performance Testing.....	4
 The Performance testing of XXX application will be carried out as described in this document. The aim would be to collect as much of information as possible. From all the Possible roots i.e. developers and others and define real life scenarios to do the performance testing of XXX application. Performance Engineer will also consider the future growth for XXX application and reproduce load conditions in the test environment to measure the performance and reliability. This performance Engineer will help Us, mentioned goals conduct, ...	
2. Pre-requisites, Assumptions and Dependencies.....	4
Pre-Requisites.....	5
Assumptions.....	5
.....	5
Dependencies.....	5
3. Application and Performance Test Architecture.....	5
XXX application Architecture.....	5
Performance Test Architecture.....	6
Project execution activities.....	7
Performance Test Automation and Execution Resources.....	8
4. Performance requirements.....	8
5. Observation points and criteria.....	8
Transactions.....	9
Connection Speed Type.....	10
Test Scenarios.....	10
1.3.1 Scenario Matrix	10
Scenario Description.....	12
1.3.1.1 General Activity Non Peak Time	12
1.3.1.2 General Activity Peak Time	13
This scenario was derived from the information obtained for the usage of web based system in the production environment. Total concurrent users executing following different activities under specified conditions form this scenario. This scenario will be executed over the course of 60 minutes and based on the generated results reports will be created.....	
	13

6. Performance Test Reports For Each Build In Each Phase.....	13
Scenario Response Time Performance Matrix.....	13
Scenario Server Resource Usage Matrix.....	14
Measure Report.....	14
Summary Report.....	14

1. Introduction

1.1 Purpose of This Document

The purpose of this document is to describe the phases, resources, assumptions, and scenarios for performance testing for XXX Application. This document also identifies the approach to be used in developing the scenarios to be tested, roles and responsibilities, and the concerns associated with the test execution plan.

1.2 Document Overview

This document has been divided into following section(s):

- *Introduction*—identifies document purpose, roles and responsibilities, hardware and software diagrams.
- *Pre-requisites, Assumptions and Dependencies* —identifies constraints, assumptions, concerns
- *Application and Performance Test Architecture* – identifies the application architecture from the performance testing perspective and details of the performance testing
- *Test Data Requirement and Performance Test Report*—identifies performance test data requirement and reports generated at the end of each performance test.

1.3 Scope of Performance Testing

The Performance testing of XXX application will be carried out as described in this document. The aim would be to collect as much of information as possible. From all the Possible roots i.e. developers and others and define real life scenarios to do the performance testing of XXX application. Performance Engineer will also consider the future growth for XXX application and reproduce load conditions in the test environment to measure the performance and reliability. This performance Engineer will help Us, mentioned goals conduct,

- √ To do appropriate performance tuning of the XXX application.
- √ To define the response time under varying load condition for real life scenarios.
- √ To define and/or test some of the early warning criteria for the XXX application.
- √ To measure the scalability of the XXX application architecture.
- √ Current Hardware, scalability.

2. Pre-requisites, Assumptions and Dependencies

Performance, load and stress testing requires the stable application and the testing environment similar to that of production environment. Following is a list of items that are required to be met before starting the performance test. Some of the items below also cover the dependencies that need to be met by other Groups during the performance testing.

Pre-Requisites

- Functionality Testing should be completed successfully. The scenarios chosen for performance testing should be fully functional.
- Working load test environment with a stable load-testable release deployed.
- Web Server Tool to calculate load, performance and stress test.
- Verification and signoff for the test scenarios and result format by the Project domain experts.

Assumptions

- √ (Development) Team should be responsible for detailed error handling and fail-over at a component level, since it is outside the scope of Tech Test.
- √ The Development, Implementation teams will put together the Operation/Production run book.

Dependencies

- Web server Test tool should be available. In case of non-availability of Web server Test tool, we need input from development and/or networking operations team about how are we going to monitor the server side resources like CPU, memory usage etc.
- Support from Development Team/Technical Architect during the test execution to quickly identify the performance issues and to fix those.
- Change in the GUI/Front End code may add extra work for the performance test automation. Performance Test Team should be informed about any significant validation related or navigation related changes in the front end.
- Appropriate access to the servers used in the performance testing. This is required for appropriate monitoring of server side resources and different log files.

3. Application and Performance Test Architecture

This describes the application architecture from the performance testing perspective and details of the performance testing strategy including the focused activities during different phases of the performance testing.

XXX application Architecture

The XXX application end user interface works over the Internet. The hardware and software configuration of different servers as shown in the diagram below.

IMP NOTE: See the documents in CVS for all.

1. Application Architecture
2. User Guide
3. Configuration
4. Admin Changes
5. All the Framework Design Documents ()

The XXX application is a truly distributed application, which involves multiple systems in executing the transaction. Careful evaluation of performance testing strategy is required to deliver the reliable and dependable application. We will divide the end-to-end XXX application into different functional applications for performance testing and perform testing in different phases to isolate the problems with each of the functional module separately.

Application Architecture. Please Refer Architecture documents in CVS.

1. Application Architecture

Performance Test Architecture

Web server Test tool from Paessler Software will be used to simulate the large volumes of users with different profiles on IE 5.0 browser with real life transaction patterns. Web server Test tool can simulate multiple threaded sessions like browsers to properly exercise server resources, simulate connection speeds to force realistic concurrency rates and can accurately simulate impact of content caching by simulated clients. Following are some of the features that are required for the performance testing and are supported by Web server Test tool.

- √ Captures traffic through proxy technology and creates transactions based on captured traffic
- √ Allows to customize/parameterize the user scripts/transactions to reproduce test scenarios close to real life
- √ Simulates the functionality of the browser during the playback by generating appropriate HTTP traffic

The “Load Generation” part for each phase of testing shows how the test environment will be set-up for load testing. Single-machine Controller simulates users using Agents running on different machines on the network. The HTTP traffic generated by the agents would hit the web server through the intranet firewall. The rest of the process between

Performance test execution plan

appserver is same as when you hit web browser manually. The controller collects the results from all the agents and produces a report at the end of the load test.

There will be one environment available for testing as described below,

1 –Performance Test will start testing performance test environment, which will be available for testing in February. This testing will help Performance Test to identify all the performance related bottlenecks in the XXX application and tune different components of the application for maximum throughput.

Project execution activities

Activity	Start Date	End Date	Deliverable	Contribution/ Dependencies
Definition of performance test execution plan and test scenarios (2 weeks)				
Setup the performance test environment, test data creation and configure the test servers				
Setup the load generation test environment	24/01/03	24/01/03	Setup controller and agent with Web server Test tool in the test environment	24/01/03
Setup the performance test servers in scaled down test environment (environment (E1)	24/01/03	24/01/03	Setup Web server, Environment build deployed.	24/01/03
Load the test data in the scaled down performance test environment (environment (E1)	24/01/03	24/01/03	Test data ready in the and all the related cases to be used in different phases of testing	24/01/03
Test Scenario automation using Web server Test tool				
Install latest build for the application in Environment	24/01/03	24/01/03	Certified environment	After Release
Revalidating the test scenarios for final build and sanity check for the functionality (Build1)	24/01/03	24/01/03	Updated Web server scripts for the test scenario and related functionality validation	
Dry run of the test scenarios	24/01/03	24/01/03	First set of performance results	31/01/03
Performance test execution for Phase 1 – Environment 1				
Test scenario execution and generate the reports for Build 1	30/01/03	30/01/03	Performance test reports and matrix	30/01/03
Test scenario execution and generate the reports for Build 2	30/01/03	30/01/03	Performance test reports and matrix	30/01/03
Test scenario execution and	30/01/03	30/01/03	Performance test	30/01/03

generate the reports for Build 3		reports and matrix	
----------------------------------	--	--------------------	--

Performance Test Automation and Execution Resources

Test Resources	How Many	Duration	Responsibilities
Test Architect	1	10 days	Responsible for defining the complete performance test plan, test strategy for each of the phases and test environments. He will come up with test automation strategy and proof of concept for the performance-testing tool.

4. Performance requirements

- All the performance tests should be repeatable (should be automated) for the successive release of the application
- Performance Engineer will be conducting the performance test for the real life like user scenarios
- Performance Engineer will be conducting the scalability test by stressing the server under various load conditions.
- While running the performance and scalability test in the end-to-end environment, Performance Engineer will be observing the server side resources to appropriately tune the server.
- It is very difficult to include all the possible scenarios or combination of functionality during the performance testing of the application so the most widely used transactions have been chosen as part of the performance testing.
- Performance Engineer will run the performance testing against the version of application, which has the data record profile similar to that in production.

5. Observation points and criteria

Observation points on Tomcat server in the XXX application will be as follows. The observation of these criteria will require some specific tools on the server and Monitor. Performance team will only observe the servers, which are part of application.

Server	Observation Points
Web Server(Tomcat)	CPU Usage Memory Usage Data Transfer between browser and web server. Hits/Second on web server with varying user load. Disk I/O (??) Web server logs

Performance Test Scenario

The details of the performance test scenarios are defined in two different forms.

1. Matrix form
2. Detail description

Following is the brief definition of Activity, Transaction, Attribute and Scenario used for performance testing.

- Activity = Set of transactions executed in specified order under identified conditions
- Transaction = Unit of operation on the application
- Attribute = Conditions under which activity will be executed
- Scenario = Set of activities (Max. 5) with specified distribution executed on the application to reproduce the real life scenarios

Transactions

Following is the list of transactions that will be used by different activities as defined in the scenario matrix. The details of the implementation of these activities can be found in the Web Server scripts.

Transaction Alias	Use Case Brief Description	Description
T0	Login	Login in to the Application
T1	Change Password	Change Password
T2	Modify Personal Information	Can Change Mail Id
T3	Change User Context	Changing of Environments
T4	Manager Log	Report
T5	Error Log	Report
T6	Create Suite	Suite Creation
T7	Create Test Case	Test Case Creation
T8	Edit Suite	Editing the Suite
T9	Edit Test Case	Editing the Test Case
T10	Manage Package	Creating Packages
T11	Upload File	Upload Files
T12	Repository Browser	Information of Test Cases and Suites

Performance test execution plan

T13	Execute Test Suite	Executing the Suite
T14	Monitor Test Suite	Monitoring
T15	Execute Test Case	Test case Execution
T16	Monitor Test Case	To Monitor Reports
T17	Test Suite Reports	Suite Reports
T18	Test Case Reports	Case Reports
T19	Log Out	LogOut

Connection Speed Type

Following is the list of different connection speed at which the end user will connect with Internet. Connection speed is one of the most important attributes for the performance testing as it defines the rate at which server will face the requests.

Connection Types Alias	Description
C1	28.8 Kbps
C2	56.6 Kbps
C3	Cable Modem

Test Scenarios

Performance Engineer has derived following scenario after careful evaluation of the information of usage of existing application in the production environment. Total 6 performance test scenarios are defined to cover different possible combinations of activities and their impact on the application performance. Number of activities per scenario have been limited to 6 for better management of the performance scenarios. The goal of the performance testing would be to run all the scenarios during each phase of testing and publish appropriate reports. The performance test scenarios are defined in two different forms,

- Scenario Matrix – Concise tabular view of the scenario, its related activities and various attributes
- Scenario Description – Detail description for each scenario including different activities, transactions and attributes.

1.3.1 Scenario Matrix

The performance scenarios identified by testing are described in a tabular form in this section. This form of the scenario is very helpful, while creating the test automation because of the clear definition of each attribute affecting the execution of scenario.

Following is the brief definition of various attributes and abbreviations used in the performance matrix.

- Activity = Set of transactions executed in specified order under identified conditions
- Transaction = Unit of operation on the application. Each transaction is formed of one or more http request.
- Attribute = Conditions under which activity will be executed
- Scenario = Set of activities (Max. 6) with specified distribution executed on the application to reproduce the real life scenario
- Total User = Maximum number of users performing different activities that will be generated during the execution of scenario
- Run Time (RT) = Total time for the execution of the scenario during which different users will be added incrementally, performance observations will be collected and gradually users will be removed from the system.
- Ramp-up Time (RUT) = Ramp up time is the duration during which new users performing different activities will be added to system at regular interval. The duration of ramp-up time (RUT) is derived as $RUT = (RT - OT) / 2$. This will give equal time to ramp-up and ramp-down the users during the performance testing.
- Observation Time (OT) = Observation time is the duration during which the performance results are collected to measure the application response time under the maximum load conditions. The duration of observation time (OT) is derived as $OT = 2 * \text{maximum time taken by any of the activity in the scenario}$
- Ramp-down Time (RDT) = Ramp up time is the duration during which new users performing different activities will be added to system at regular interval. The duration of ramp-down time (RDT) is derived as $RUT = (RT - OT) / 2$. This will give equal time to ramp-up and ramp-down the users during the performance testing.
- Load Increment (LI) = This defines the rate at which new users are added to the increase the load.
- Connection Speed (CS) = Connection speed comprises of % of users connected at different connection speed to the internet
- Activity Load (AL) = Activity load defines the % load of user executing selected activity compared to the total number of users for the respective scenario.
- Activity Time (AT) = Activity time is the total time taken to execute one iteration of the respective activity.
- User Profile (UP) = User profile defines the type of account that will be used by the group of users executing respective activity.

Performance test execution plan

Scenario Name	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6
Scenario 1 Total users = Run Time= minutes Ramp-Up Time= Observation Time= Ramp-Down Time=						

Special Notes:

- To address the impact of different speeds to connect to the Internet for each of the individual user, the variation has been included in speed however the distribution of Connection Speed is constant for all scenarios.
- Activities for all scenarios are common due to the High Volume Usage for this particular set of activities as derived from the statistical information of the current usage of application in testing .
- Activities in all scenarios are defined by the performance testing based on the knowledge of the projection of use of respective functionality.

Scenario Description

- This section provides the detail description of the performance test scenarios. Each scenario defined below is a combination of different activities being executed under different conditions to reproduce the real life scenario that application need to handle in the production environment.

1.3.1.1 General Activity Non Peak Time

Total concurrent users executing following different activities under specified conditions form this scenario. This scenario will be executed over the course of 60 minutes and based on the generated results reports will be created.

1.3.1.2 General Activity Peak Time

This scenario was derived from the information obtained for the usage of web based system in the production environment. Total concurrent users executing following different activities under specified conditions form this scenario. This scenario will be executed over the course of 60 minutes and based on the generated results reports will be created.

This scenario was derived from the information obtained for the usage of web-based system in the production environment. Total concurrent users executing following different activities under specified conditions form this scenario. This scenario will be executed over the course of 60 minutes and based on the generated results reports will be created.

Incremental Load: Following characteristics are observed while creating the incremental load for all the activities.

- Transaction defined for a user must complete execution before the time specified to increase the number of users.
- Results will be skewed if transactions take longer to execute than the incremental time specified.

Stress Test: Stress Test is equivalent to running a load test with no delays (think time) for human interactions and executed for measuring the scalability of the application.

Connection Speed: To duplicate the environment that typical users would use, the connection speed is an important parameter in load testing.

6. Performance Test Reports For Each Build In Each Phase

This section is defining the template/format of the reports/matrix that will be generated on completion of the performance test. The details on any of the reports can be obtained from the Web Server Stress Tool if required.

Scenario Response Time Performance Matrix

This matrix is the consolidated report of the average response/execution time for all scenarios with varying load condition. This report is not generated automatically by Web Performer but is consolidated at the end of complete test cycle/phase.

Scenarios-Load and Response Time	1 User	2 Users	3 Users	4 Users	5 Users	10 Users	20 Users

Performance test execution plan

S1							
S2							
S3							
S4							
S5							
S6							

Scenario Server Resource Usage Matrix

During different phases of testing the performance testing will be observing the resources used on various servers of the application. Memory and CPU utilization would be the primary focus of observation and following report would be derived at the end of complete test cycle/phase.

Scenario/Resource Usage	Web Server		App Server		Database Server	
	Memory	CPU	Memory	CPU	Memory	CPU
Server Status:						

Measure Report

Following is an example of the measure report generated by Web Server tool at the end of test after collecting all the results. This report includes the average, minimum and maximum time taken for selected transactions or activity executed during the test.

.

Summary Report

The summary report generated by Web Server tool at the end of test after collecting all the results. This report gives the details of various details of HTTP transaction which includes total request sent, total request failed, total amount of data sent and received, occurrences of successful and unsuccessful connections etc.