HP Application Lifecycle Management 11.52 Benchmark

ALM PCoE

Document Release Date: May 2013

Software Release Date: May 2013

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.
Microsoft SQL Server is a trademark of the Microsoft Corporation.
Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.
UNIX® is a registered trademark of The Open Group.
Red Hat Linux is a trademark of Red Hat, Inc.
Solaris is a registered trademark of Sun Microsystems, Inc
WebLogic is a trademark of Oracle Corporation
WebSphere® is a trademark of IBM Corporation.
Jboss is a trademark of JBoss Inc.

Contents

1 Introduction

1.1. Audience

This document is for those who plan, size, set up, and maintain the HP ALM 11.52 environment.

1.2. Disclaimer

The results published in this document are accurate for the below tests only when they are run on the specific environments described. Different ALM configurations or environments may have different performance results for the same tests.

Factors that commonly affect performance include: Active Project sizes structure and quantity, user flows, network configurations, JVM configuration, ALM configurations (Caches), Database tuning, and Hardware configuration (RAM, CPU, disk size).

The aim of this document is to provide an analysis of ALM performance as reflected in benchmark tests described below. The document should be used as a general guideline. For more specific capacity planning, conduct load tests directly on your actual environment.

1.3. Purpose and scope

This document summarizes the three main performance tests that were done for ALM 11.52:

1. **Regression.** Verified there are no regressions in Transaction Response Time between versions.
2. **Single user.** Verified there are no regressions in client performance.
3. **Stability.** Tested system endurance for 10 days.

1.4. Terminology

Transaction	A measurement of one user action defined on load script.
Transaction Response Time (TRT)	The time measured by a LoadRunner generator as the response time of the transaction. This is defined by LoadRunner as the time taken for all Web/OTA requests belonging to the transaction to return the responses. It is also called **Transaction Latency**.
Overall TRT	Weighted average of all TRT during test execution.
Performance counters	Counters are used to provide information as to how well the operating system or an application, service, or driver is performing. The counter data can help determine system bottlenecks and fine-tune system and application performance. The operating system, network, and devices provide counter data that an application can consume to provide users with a graphical view of how well the system is performing.
Server Side TRT	TRT of server and network (excluding UI time).
User experience	TRT from user operation in client until response is received, rendered and displayed. Response time includes client, network, file system, database and server time.
JVM Throughput	A JVM's *throughput* accounts for the percentage of total time garbage collection does not take place. For instance, 80 percent throughput implies that garbage collection consumes 20 percent of the JVM's processing while your application consumes only 80 percent.
AUT	Application under test
LoadRunner (LR)	**HP LoadRunner** is an automated performance and *test automation* product. It examines system behavior and performance while generating actual load.

2 Server Side Regression Test

2.1. Executive summary

Server side regression testing in ALM 11.52 includes running the same load on ALM 11.52 and older versions, and then comparing their server performance.

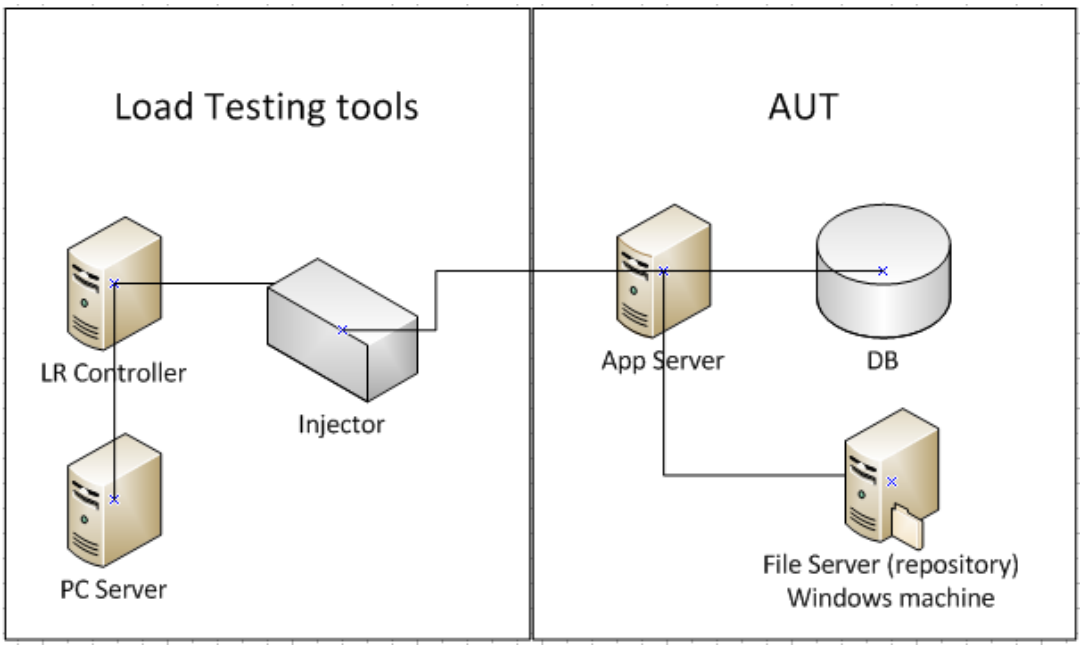
The current ALM version (11.52) was tested and compared to QC 10 patch 30, ALM 11 patch 12 and ALM 11.51.

All the tests were conducted using minimum hardware requirements and with recommended software installed.

The results of a comparison between QC 10 and ALM 11 to ALM 11.52 showed improved performance response time for most transactions. The improvement was due to enhancements to the jobs cache mechanism efficiency. There is no significant change in performance between ALM 11.52 and ALM 11.50 Integration Enablement Pack.

2.2. Environment settings

The following diagram illustrates the testing environment structure:



Application server software

Version	OS	Database	Web server
ALM 11.52/ ALM 11.50 Integration Enablement Pack	Windows Server 2008 R2 SP1 (64 bit) and Linux Red Hat 6.2 64bit	Microsoft SQL 2008 R2 SP1 and Oracle 11.2.0.3	IIS 7.5/Apache 2.2
ALM 11 Patch12	Linux Red Hat 5.4 64bit	Oracle 10.2.0.4	Weblogic 10.3
QC 10 (Patch 30)	Windows Server 2003 SP2 (64 bit)	Microsoft SQL 2005 SP2	JBoss 4.0.4

Application Server Hardware

Environment	Model	CPU	RAM	Controller
ALM 11.52/ ALM 11.50 Integration Enablement Pack	ESX 5.1 VM version 8	Xeon X5670 (4 Cores) HT enabled	8 GB	Smart Array P410i
ALM 11 Patch12	ESX 5.1 VM version 8	Xeon X5670 (4 Cores) HT enabled	4 GB	Smart Array P410i
QC 10 (Patch 30)	ESX 5.1 VM version 8	Xeon X5670 (4 Cores) HT enabled	4 GB	Smart Array P410i

DB Server Hardware

Environment	Model	CPU	RAM	Controller
ALM 11.52/ ALM 11.50 Integration Enablement Pack	ESX 5.1 VM version 8	2*Xeon X5670 (4 Cores) HT enabled	16 GB	Smart Array P410i
ALM 11 Patch12/ QC 10 (Patch 30)	ESX 5.1 VM version 8	2*Xeon X5670 (4 Cores) HT enabled	8 GB	Smart Array P410i

2.3. Load Profile and Test Scenario

- 1000 active users were loaded.
- 100 active projects:
  - 20 ALM projects which contain real-world data and structure. The project structure is based on selected enterprise ALM customers. See appendix 1 for more details.
  - 80 empty projects, used only for login-logout during the load (for jobs and cache testing).
- 5-6 transactions to server per second. See the "Project Load Profile" section for more details.
- Load execution duration: 5 hours.

2.3.1 Project Load Profile

In order to do a full regression test we used a scenario that contained the main functionalities of common modules. The simulated usage profile (users' distribution among modules and number of events per hour per user) was based on a workload analysis of selected ALM customers, representing different deployment sizes and configurations.

The load profile for the test was based on the four main ALM modules, and the most common and important business processes that exist in all tested versions. The following table shows the number of transactions executed during one hour of load:

Generic Transactions	**Transactions per hour**
File operations	362
Login	595

Module	**Transactions per hour**
Requirements	1,756
Test Plan	5,134
Test Lab	1,786
Defects	10,071

2.4. Test results

- **QC 10 to ALM 11.52**
  - Overall TRT improved in ALM 11.52.
  - One TRT regression was detected – DisconnectProject response time changed from 0.03 sec to 1.2 sec.
- **ALM 11 patch 12 to ALM 11.52**
  - Overall TRT has improved in ALM 11.52.
- **ALM 11.50 Integration Enablement Pack to ALM 11.52**
  - No difference in overall TRT.

3 User Experience Regression Test

3.1. Executive summary

The goal of this test was to verify that there was no regression in user experience compared to older versions.

In this test we defined 40 of the most common user operations and executed them over 3 QC\ALM versions: QC 10, ALM 11 and ALM 11.52.When comparing the response time of operations in ALM 11.52 to ALM 11, we found that all the operations response times have improved or remained the same.

When comparing the response time of operations in ALM 11.52 to QC 10, we found that most of the operations response time have improved or remained the same. Four operations have some regressions in performance.

For further details, see section 3.4.

3.2. Environment settings

For all versions, we used the recommended software for client and server, and the minimum requirements for hardware.

Client machine

Software version	Hardware version
Operating System: Windows 7 Professional sp1 32 bit	Processors: Intel® core™ 2 6600 CPU (2 cores)
	Memory: 4GB RAM

Application server software

Version	OS	Database	Web server
ALM 11.52/ ALM 11.50 Integration Enablement Pack	Windows Server 2008 R2 SP1 (64 bit) and Linux Red Hat 6.2 64bit	Microsoft SQL 2008 R2 SP1 and Oracle 11.2.0.3	IIS 7.5/Apache 2.2
ALM 11 Patch12	Linux Red Hat 5.4 64bit	Oracle 10.2.0.4	Weblogic 10.3

QC 10 (Patch 30)	Windows Server 2003 SP2 (64 bit)	Microsoft SQL 2005 SP2	JBoss 4.0.4

Application Server Hardware

Environment	Model	CPU	RAM	Controller
ALM 11.52/ ALM 11.50 Integration Enablement Pack	ESX 5.1 VM version 8	Xeon X5670 (4 Cores) HT enabled	8 GB	Smart Array P410i
ALM 11 Patch12	ESX 5.1 VM version 8	Xeon X5670 (4 Cores) HT enabled	4 GB	Smart Array P410i
QC 10 (Patch 30)	ESX 5.1 VM version 8	Xeon X5670 (4 Cores) HT enabled	4 GB	Smart Array P410i

DB Server Hardware

Environment	Model	CPU	RAM	Controller
ALM 11.52/ ALM 11.50 Integration Enablement Pack	ESX 5.1 VM version 8	2*Xeon X5670 (4 Cores) HT enabled	16 GB	Smart Array P410i
ALM 11 Patch12/ QC 10 (Patch 30)	ESX 5.1 VM version 8	2*Xeon X5670 (4 Cores) HT enabled	8 GB	Smart Array P410i

WAN between the client and server was simulated using a WAN simulator.

The latency defined was 160 ms for one direction (320 ms for round trip). This was done to simulate very high latency between client and server.

3.3. Load Profile and Test Scenario

Each operation was repeated 3 times. The test was performed on a big project which was synthetically generated and used for all performance testing. Database caches were cleaned between each operation. The operations were run with no load on the background, on a completely isolated environment.

See Appendix 1 for details about the project structure.

Here are examples of the operations and their details:

Operation Name	Details
Create requirement	• Create folder on the 4th level of the tree • Create one requirement
Filter requirement – Tree view	• On tree view, filter requirements by: req parent="Requirements\Requirement_1-1", priotiy=5-Urgent Or "4-Very High",type=Testing
Update requirement	• Open a requirement in the 4th level of tree • Update 3 requirement fields • Submit
Move requirement	• Create Folder at the 4th level • Create 10 Reqs • Move the folder to other tree to the same 4th level
Delete requirement	• Create Folder at the 4th level • Create 10 Reqs • Delete Folder
Copy paste requirement	• Create Folder at the 4th level • Create 10 Reqs • Copy requirement folder • Create a folder at the 5th level of the tree • Paste folder into new folder

In addition, another test with smaller amount of operations was run with the same conditions over load to verify that load on the server produces similar results for a single user.

3.4. Test Results

When comparing the response times of operations in ALM 11.52 to ALM 11, we found that all the operations response times improved or remained the same.

When comparing the response times of operations in ALM 11.52 to QC 10, we found that most of the operations response times improved or remained the same. A small number of operations did have some regression in performance.

There were 2 operations which had changes in functionality which resulted in a higher TRT:

1. **Move requirement (bulk).** A locking mechanism was added on requirement move.

Duration in QC 10	Duration in ALM 11	Duration in ALM 11.52
5 sec	25 sec	8 sec

2. **Test coverage.** Multiple entities were added to ALM Test Configurations and Test Criteria. ALM 11.52 provides a finer breakdown of the coverage analysis

Duration in QC 10	Duration in ALM 11	Duration in ALM 11.52
3 sec	18 sec	15 sec

In addition, the following transactions also showed some regression in TRT:

1. **Customization.** Regression when opening the customization module.

Duration in QC 10	Duration in ALM 11	Duration in ALM 11.52
5 sec	17 sec	9 sec

2. **Customization**. Regression when creating a new test type.

Duration in QC 10	Duration in ALM 11	Duration in ALM 11.52
1.5 sec	11 sec	8 sec

4 Stability Test

4.1. Executive summary

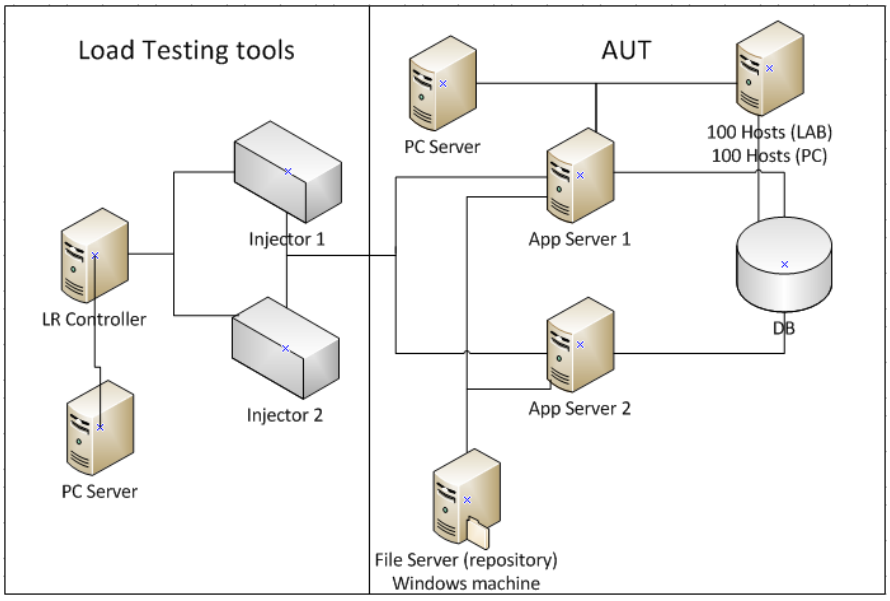
The purpose of this test was to check the endurance of the full ALM system when under high load.

The test was performed 3 consecutive times, each run for 80 hours. The application server was up for the entire test time frame.

During the entire run, the system was stable – no increase or peaks in TRT and system resources.

4.2. Environment settings

The stability test was run on 2 environments, both with the same structure described in the diagram below. The difference between the environments is in the application server's hardware and software configuration details. The application servers used for the tests are described on the tables below.



Environment Software Versions

Software type	Version
App server OS	Windows 2008 R2 EE & Linux RedHat 6.2
Database	SQL 2008R2 SP1 & Oracle 11.2.03g
Web reverse proxy	IIS 7.5 & Apache 2.2

Environment Hardware Versions

Application Server Hardware

Environment	Model	CPU	RAM	Controller
Windows	ProLiant BL460c G7	2 * Xeon X5650 (6 Cores) HT enabled	16 GB	Smart Array P410i
Windows	ProLiant BL460c G7	2 * Xeon X5650 (6 Cores) HT enabled	16 GB	Smart Array P410i
Linux	ProLiant BL460c G7	2 * Xeon X5650 (6 Cores) HT enabled	16 GB	Smart Array P410i
Linux	ProLiant BL460c G7	2 * Xeon X5650 (6 Cores) HT enabled	16 GB	Smart Array P410i

Database Server Hardware

Environment	Model	CPU	RAM	Controller
Windows	ProLiant DL380 G7	2 * Xeon X5650 (6 Cores) HT enabled	32 GB	Smart Array P410i
Linux	ProLiant DL380 G7	2 * Xeon X5650 (6 Cores) HT enabled	64 GB	Smart Array P410i

4.3. Load profile and test Scenario

- Load of 1000 active users.
- 300 active projects in site:
  - 20 ALM projects which contained real-world data and structure. The project structure was based on selected enterprise ALM customers. See Appendix 1 for more details.
  - 280 empty projects, used only for login-logout during the load (for jobs and cache testing).
- 9-10 requests per second on both nodes (4-5 per node). See the "Project Load Profile" section for more details.
- Load execution duration: 72 hours. ALM full system includes: 2 ALM Servers, PC Server, 100 PC hosts, 100 LAB hosts.
- 65 concurrent PC runs
- 65 concurrent LAB runs

4.3.1 Project Load Profile

The stability test provides information regarding the product's stability by covering the main functionalities of common modules. The simulated usage profile (i.e., users' distribution among modules and number of events per hour per user) was based on a workload analysis of selected ALM customers, representing different deployment sizes and configurations.

The load profile used for the test is based on the four main ALM modules, and most common and important business processes.

The following table shows the number of transactions that ran on each module during one hour of the system load:

Generic Transactions	**Transactions per hour**
File Operations	362
Login	1,185

Module	**Transactions per hour**
Requirements	3,350
Test Plan	10,217
Test Lab	3,558
Defects	18,115

4.4. Test Results

The test was performed 3 consecutive times, each run for 80 hours. The application server was up for the entire test time frame.

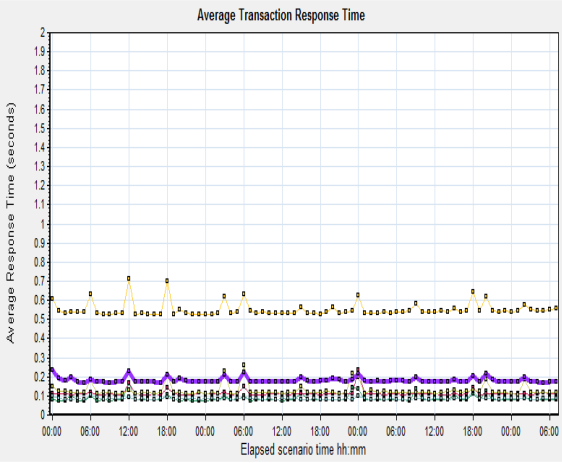
During the entire run, the system was stable – no increases or peaks in TRT and system resources.

The following are some measurements taken from one 80 hours run:

Category	Measurement	Avg. duration (sec)	99 Percentile (sec)
Transaction Response Time (TRT)	Create defect	0.085	0.234
	Modify defect	0.113	0.238
	Filter defect	0.550	0.842
	Create requirement	0.086	0.203
	Create manual run	0.127	0.328
	Expand test folder	0.180	0.437

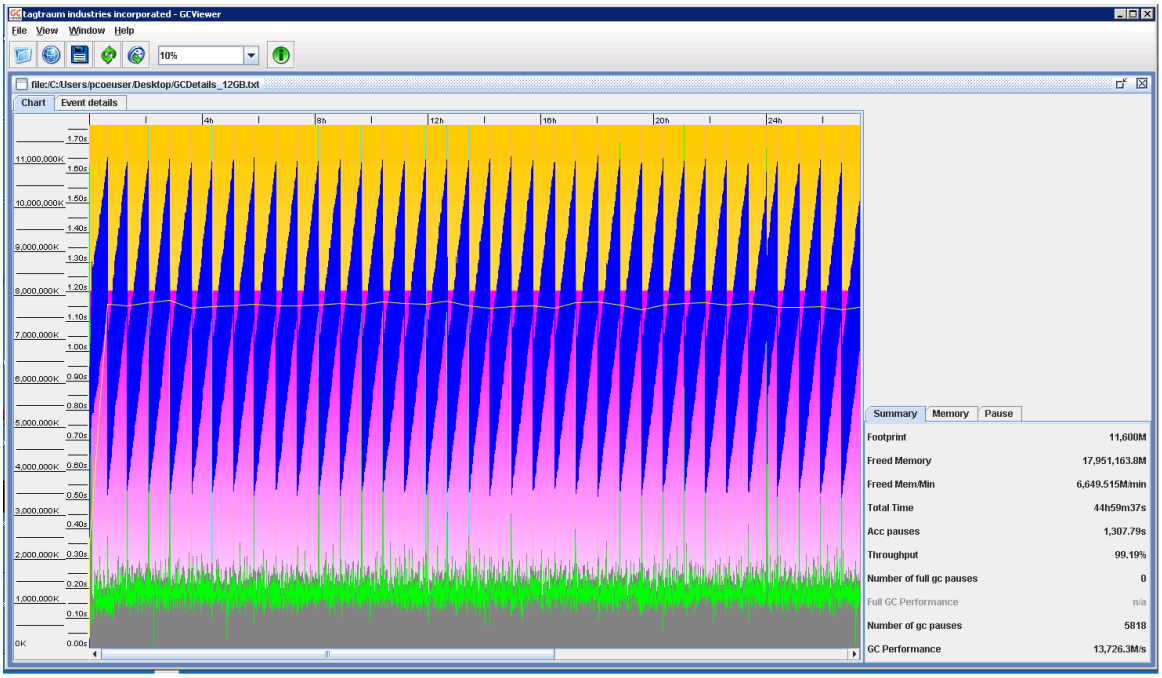
Category	Measurement	Avg. Value	Max. Value
System resources	% total CPU time	3.4 %	21.5 %
	JVM throughput	99.19 %	---------

The following graph shows TRT during the entire load. Each line represents the TRT of one transaction in the load over time. The graph indicates that transactions are stable over time:



Color	Scale	Measurement	Average	99 Percentil
	1	Create Defect	0.085	0.234
	1	Modify Defect	0.113	0.238
	1	Filter Defects	0.550	0.842
	1	Create Requirement	0.086	0.203
	1	Create Manual Run	0.127	0.328
	1	Expand Test Folder	0.180	0.437

The following graph shows JVM heap usage of one node over time. The blue line indicates JVM heap usage. A sharp decrease in heap usage occurs during the JVM full garbage collection. This graph depicts good JVM behavior: no increase in memory usage (memory leak) and full garbage collection every few hours which results in a very high throughput.



Appendix 1: Project Data Profile

The following table contains the details of the profile used for each ALM entity in the project:

Module	Entity	Amount	Structure
Release	Release Folders	36	1 Root Folder -> 5 Folders -> 2 Folders -> 10 Releases -> 10 Release cycle
	Releases	200	
	Release Cycles	2000	
Requirements	Requirements	30692	1 Root Requirement -> 30 Requirements -> 2 Req (9-10 Levels)
Business Components	Component Folder	107	1 root folder-> 100 folders -> 50 components -> 100 steps
	Component	5210	
	Component Step	506000	
Test Plan	Test Folder	41152	1 root folder -> 70 folders
	Test	185261	1 root folder -> 19 folders -> 2 Folders with 4 Tests (9 levels) -> 5 tests 1 root folder -> 50 folders -> 20 tests
	BP Test to component	20210	1 root folder -> 50 folders -> 20 tests -> 20 components
	BP Parameter	202100	1 root folder -> 50 folders-> 20 tests -> 20 components -> 10 parameters
Test Lab	Cycle Folder	2802	1 root folder -> 50 folders -> 5 folders -> 2 folders -> 4 folders -> 10 test sets -> 20 tests
	Cycle	20002	
	Run	400100	
Defects	Defect	60000	60000 defects
ERI	Asset Relation	337656	1 root folder -> 10 resource folder -

	Asset Repository Items	414396	> 2 resource folder -> 5 resource folder -> 5 resource folder -> 2 resource folder -> 9 files (5 Lib, 3 Object Rep. 1 recovery scenario)
LAB	Time slots	3500	
	Hosts	100	
PC	Hosts	100	