



LOAD RUNNER

TECH VISION JULY 2016 BATCH

ANAND ANEESHA NAresh RAMANAKAR REDDY VINAY

Contents

1. Introduction:	3
2. Load Runner Components:	4
3. Load Runner Architecture:	4
4. Performance Test Life Cycle:	5
1. Proof of Concept (POC):	5
2. NFR – Non Functional Requirements	6
3. Test Plan	7
4. Work Load Model.	8
5. Recording and Enhancements	8
Correlation:	10
Parameterization:	16
Verification points/checkpoints:	22
Advanced Techniques	23
C Data Types:	23
File operations:	27
Pre-defined functions:	29
LR Functions:	29
Web Functions:	34
Recording Options:	44
6. Test Data:	46
Regular Expressions:	48
Fiddler:	49
CONTROLLER	50
IP Spoofing:	52
Runtime settings:	54
Pacing Calculations:	59
Performance Center:	67
ALM (Application Life Cycle Management)	69
Types of testing	70
ANALYZER	73
Thread Dump:	80

Memory Dump:	81
Difference between Load Runner versions:	82
Profiling Tools:	83
1. JConsole:	83
2. JVisualVM:	84
3. JMC(Java Mission Control):	85
Monitoring Tools	87
1. Site scope:	87
2. Dynatrace:	88
3. HP Shunra:.....	100
4. Perfmon:	101
CVS/VSS (Concurrent Versioning System/Visual Source Safe)	101
Bottlenecks	102
COUNTERS	110
Protocols	113
1. SAP Web Protocol:	113
2. SAP GUI Protocol:	116
3. Web Services Protocol:	118
4. Web click & Script Protocol	123
5. RTE (Remote Terminal Emulator) Protocol	125
6. Citrix ICA Protocol	127
7. Oracle NCA Protocol	129
8. The Ajax TruClient User Interface	130
AWR REPORTS	145

LOAD RUNNER:

1. Introduction:

One of the best automated performance tool. It uses ANSI C as the default programming language and other languages like JAVA & VB.

Advantages:

- It supports all types of protocols (HTTP, FTP and SMTP).
- Easy to analyze the results and generating scripts.

Load Runner was invented by Mercury, and acquired by Hewlett – Packard Organization.

Performance Testing:

Performance Testing is a Non-Functional testing performed to evaluate application performance under some load and stress condition. It is generally measures in terms of speed, scalability & stability for the user activity.

Speed: Determines whether the application responds quickly.

Scalability: Determines maximum load the software application can handle.

Stability: Determines if the application is stable under varying loads.

There are different types of performance testing. They are

- Load Testing
- Stress Testing
- Endurance Testing
- Scalability Testing

Why Performance testing required for any application?

We should verify “the application behaving under load” and to avoid the **business loss**.

Q. Can't we conduct the performance testing without tool?

Sol: We can conduct the performance testing without a tool. In realistic environment application accessed by 10,000 people, to simulate the realistic environment we have to require 10,000 people, which is not possible to require these many resources.

Below are the constraints stopping us to conduct the PT without tool.

- Accuracy
- Time

- Budget
- Resources

To overcome the above issue. We have to use tools instead of manual resources.

Q. What are the tools available in market?

Sol:

Load Runner	-	HP
RPT [Rational Perform Testing]	-	IBM
J Meter	-	Open Source
Neo Load	-	Neotys
OpenSTA	-	Open Source
Oats	-	Oracle
VSTS	-	Micro Soft
SPT	-	Borland
SOASTA	-	Open Source
Web Load	-	Open Source

2. Load Runner Components:

1. VUGen – Virtual User Generator

It will allow you to record and replay script. And allow you to enhance your script.

2. Controller

It will allowing you to design the scenarios and execute the test with multiple users.

3. Analyzer

It will allow to analyze the statistics and finding out the bottlenecks

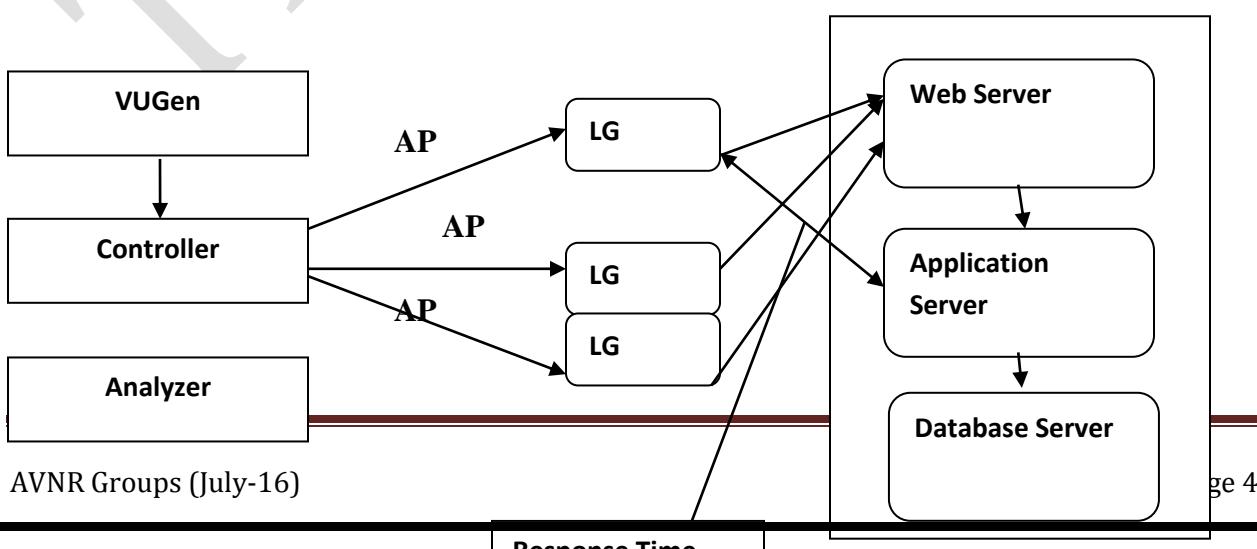
4. Load Generator

It generates the load against the application.

5. Agent Process

It will establish the connection between Load Generator & Controller.

3. Load Runner Architecture:



Response Time: The roundtrip of request is called Response Time.

Q. Load Runner License

- VUGen, Load Generator, Agent Process, Analyzer are available at free of cost.
- We have to buy **control license** using controller you can conduct only one test at any point of time.
- You have to buy **Vuser License**.
- You have to buy **protocol bundle** license.

Note:

- ✓ While procuring HP load runner license you receive SAID (Software Agreement Identification Number).
- ✓ Using SAID we can raise the tickets or requests in HP site to get the assistance from HP to resolve the issues.

4. Performance Test Life Cycle:

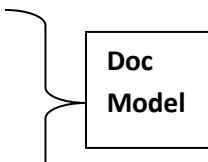
Q. Performance Test Life Cycle (or)

Roles and Responsibilities in current Project (or)

Process usually you will follow in Performance Testing Project

Sol:

1. **POC** - Proof of Concept
2. **NFR** - Non Functional Requirements Gathering
3. **Test Plan**
4. **Work Load Model / Load Model / Volume Model / Volume Metrix**
5. **Scripting & Enhancements**
6. **Test Data**
7. **Scenario Design & Execution**
8. **Analysis**
9. **Reporting & Recommendation.**



1. Proof of Concept (POC):

Sol: As part of POC, I will try to understand the application architecture, what kind of application is this? Which technology they used to develop this application. Which communication mechanism they used to develop this application and complexity of the application.

I will try to identify which tool is compatible for this application, if it is load runner which protocol is compatible for this application, whether it is a single/multiple protocols.

As part of POC, I am going to record simple business scenarios, execute with minimal users & report the response times to the client.

Based on the above POC, Client has to understand which tool he supposed to buy, which protocol bundle he supposed to buy & how many labor hours required to finish this application.

2. NFR – Non Functional Requirements

Sol: We have to prepare NFR document with a help of BA (Business Analyst) people or project team people.

NFR contains CBT's (Critical Business Transactions), peak hours, half peak hours (low load), expected average response times, no. of transactions per hour & hardware threshold statistics like CPU, Memory, Heap, Swapping.

We have to gather and what kind of tests we supposed to perform in terms of availability, serviceability, scalability, recoverability, baseline, benchmark, fileover, abnormality & spike.

After prepare NFR document, we have to get it approved by project team, architects, infrastructure team, NW team and stakeholders.

What is your approach to gather NFR, if client doesn't know anything about performance testing?

Scenario-1:

If Application is already in production

Sol: Get the production log files for 1 year historical data using Site analytical tools (or) splunk tool & try to identify top 5 usage days from that how many no. of visitors accessing that application, how many no. of page views are happening, which JSP/ASP pages are mostly accessed by end users & from which region they are accessing the application.

Based on the visitors you can identify no. of users, based on the page views you can identify no. of transactions.

Based on the JSP/ASP pages you can identify the CBT's, based on the IP address we can identify the location/region & conduct baseline testing to identify the expected response time (or) to baseline the application.

Note:

Whenever we don't have (SLA's: Service Level Agreement) expected response times, we have to conduct baseline test.

Scenario-2:

Client doesn't know anything & application not in production.

Solution: Get the competitor statistics and conduct the PT

Sol: Using N/W traffic utilities (Ex: www.alexa.com) get the competitor statistics & identify NFR's in terms of peak hours, half peak hours, no. of users, no. of page views from which location they are accessing the application, which pages mostly access by end users & what are the expected response times for every page.

Scenario-3:

Client doesn't know about performance testing & application not in production & we don't know have a competitor in market.

Sol: Try to understand the Core Business of the application and convert core business to Online Business. By converting identify how many no. of users, no. of transactions, CBT's, from which region they are accessing the application and conduct the base line test to derive SLA's.

Below are the questions we asked to BA (Business Analyst):

Q-1: How many no. of customers do have in core business?

Sol: 1.3m customers.

Q-2: How many of them are going to register for online services?

Sol: 10%

Q-3: How many of them are active users?

Sol: 10%

Q-4: How many of them are daily users?

Sol: 30%

Q-5: What are the peak hours for our business?

Sol: 4 hours

Based on above statistics we derived the number of users and number of transactions.

Generic Case Study:

I have an application stake holders & project is not in a situation to provide the requirement. So we approached Business Analyst (BA: People to finalize the NFR.

3. Test Plan

Sol: Test Plan is a road map of our test,

Test Plan contains objective of the test, scope, items out of scope, procedure, approach, test data, CBT's, types of testing, monitoring, application architecture, tool architecture, deliverables, roles & responsibilities, environment, risks & migrations, entry criteria, exit criteria, prerequisites & assumption.

Entry Criteria:

Whenever the Pre-requisites are satisfied that is called **Entry Criteria**.

Exit Criteria:

Whenever derived statistics are meeting expected statistics that can be considered as **Exit Criteria**.

Note: Test strategy will describe approach & procedure of the test.

4. Work Load Model

Sol: Work Load Model is a heart of all the documents which will be lead you

“How to test that application?”

&

“What to test that application?”

Work Load Model contains CBT's, business flows for every CBT's, no. of transactions, types of testing, load distribution based on types of testing.

While preparing Work Load Model we have to consider Pacing & Think time calculation to generate ‘Anticipated Load’.

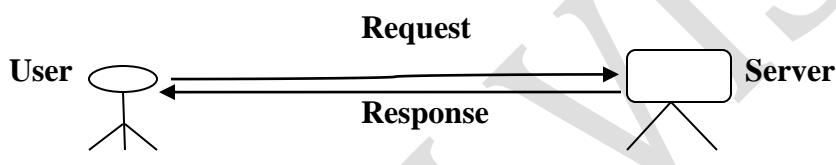
Work load model will assist you to design the scenario.

Input: Use cases, types of testing, number of users

Output: Pacing, think time and transaction mix

5. Recording and Enhancements

Protocol: Protocol is a set of rules.



Protocol Adviser:

This option is available from 9.5 version & allow you to choose appropriate protocol for an application.

Note:

Protocol Adviser never help you to choose the appropriate protocol in realistic environment. We have to understand the communication mechanism & architecture, we used to contact AD people to understand which communication mechanism they used in application development. Based on all these we used to finalize the protocol for an application.

Famous Protocols in current market:

Citrix ICA: Whenever application published in Citrix environment we have to Citrix ICA protocol.

Flex: Flex (Rich internet applications) is a product of Adobe. If the application front end is developed in flex then use Flex protocol.

Ajax TruClient: If you would like to measure the response times which includes server side and browser response times (Java script execution)

RTE (Remote Terminal Emulator): If the application is developed in Unix Environment. We have to use RTE Protocol.

Ex: Cursor based application, Mainframe application.

SAP Family: if the application developed in ECC and netviewer and dynapro portal we can use these protocols.

Web (HTTP/HTML): It is famous among all the protocols. If the application uses http protocol then use this protocol.

Web Services: It is also one of the famous protocols. If the application uses API's or web services or xmls or jsons or restful xmls then we have to use this protocol.

Web (Click and script): It will record browser specific requests

Note:

- ✓ Start HP Web Tours Server.
Then only we access the application.(its specific to web tours application)
- ✓ Web Browser
Invoking application through web browser.
- ✓ VUGen
It invokes the actions & then convert to the script code.

Correlation:

Correlation is the concept to handle the dynamic values which are generating from server side.

(Or)

Dynamic input to the server. You can conduct the correlation in 2 ways:

1. Auto Correlation.
2. Manual Correlation.

1. Auto Correlation:

Automatically handle the dynamic values.

Process:

- **Auto correlation process after 11.5 version**
- 1) Record the script.
 - 2) Replay the script (It's not mandatory from LR 11.5 version).
 - 3) Click the design studio.
 - 4) Choose the dynamic value.
 - 5) Click on "Correlate button".

From the above steps, which will create a function & automatically substitute the dynamic value.

- **Auto correlation process before 11.5 version.**
- 1) Record the script.
 - 2) Replay the script at least once.
 - 3) Click on "Correlate button".
 - 4) Choose the dynamic value from correlate log.
 - 5) Click on "Correlate button".

Note:

A design studio or correlation log compares record time response with replay time response dump the differences into the correlation log or design studio.

2. Manual Correlation

Manually we have to handle the dynamic values. Whatever the steps performed by autocorrelations steps will be performed manually.

Process:

Step1: How to identify the dynamic values in a script?

Sol: Record the same business scenario twice with 2 sets of data & compare both of them using "**beyond-compare**" or "www.text-compare.com" or Tools→ "**Windiff**" (default).

Which values reported differently those values will be considered as dynamic values.

Note:

Which values are not enter through keyboard & which values are chosen from the application, those values will be considered as dynamic values.

Step2: How to identify the left text & right text for a dynamic value?

Sol: web_reg_save_param("parameter_name","lb=","rb=",LAST);

→ Copy the dynamic value.

→ Go to generation log identify first occurrence of the dynamic value.

→ Copy the left text & right text of the dynamic value.

Step3: Where to write the function?

web_reg_save_param("parameter_name","lb=userSession" value="" ","rb=""/>",LAST);

Sol1: Execute the script in full mode (F4→LOG→Extended Log→Enable All Options). Identify the dynamic value based on the left boundary or right boundary in Replay log. Double click on the same line which will assist you to where to write the function or for which request the dynamic value was generated.

Sol2: (Not recommendable for web tours) once you identified the dynamic value in generation log place the cursor on the same line and search for "web_" or "ENDED" then copy the request and search for the same in script. On top of the same request u can place the function.

NOTE: Generation log contains response and request which never change.

Step4: Substitution

Sol: You can substitute the parameter name where ever the dynamic value is present in script using control+H.

Correlation Function with all Arguments

1. ORD:

ORD stands for "Occurrence".

By default ORD=1.

Ex:

Ord=10 stands for 10th occurrence.

Ord=ALL stands for captures all values based on LB & RB.

Whenever you written ORD=ALL your parameter become an Array. It holds all the values with the name of parameter-1, parameter-2 & parameter-N.

Note:

Variable_count let we know the length of an Array.

2 .Save Length:

Allow you to capture specific no. of characters from dynamic value based on LB and RB.

Ex: saveLen=3

By default saveLen="-1".

3. Save offset:

It will specify the no. of characters it supposed to skip from the dynamic value.

By default saveOffset="0".

EX: saveOffset=3 (It skips 3 char)

4. Not found:

By default notfound="error".

If the either LB or RB is not present in the response script will through the error like "No match found for the requested parameter".

Notfound="warning".

If you would like to continue the script execution, even LB or RB is not available in the response you have to use warning.

5. Search:

It is an optimization technique. Which will instruct the user to search for LB & RB in a specific location.

By default search=ALL.

Ex: Search= headers: - it will search for LB n RB in only on header

Ex: Search= Body: - it will search for LB n RB in only on body

6. RelframeID:

This is also optimization technique. Specifying to capture the dynamic value from a specific frame.

Ex: RelframeID=1.2.1

HTTP Status Codes:

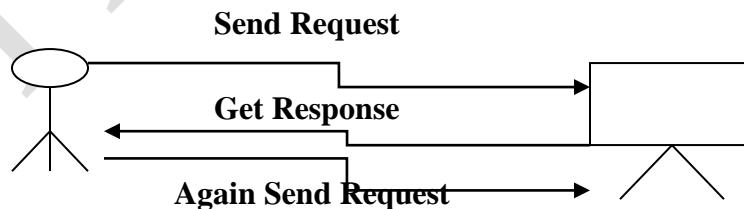
- 100** - Request successfully delivered.
- 200** - Response received successfully.
- 300** - Received response but it was redirected response.
- 400** - Invalid request.
- 500** - Application itself not available.

7. Ignore Redirection:

By default ignore redirection= YES

Instructing correlation function to capture dynamic value from a redirected response or not.

8. Convert:



- | | | |
|-------------|---|-----------------------------------|
| HTML | - | Hyper Text Markup Language |
| XML | - | Extensible Markup Language |

Ex: HTML to URL

Whenever the dynamic value is recorded in the format of URL and the captured value in the format of HTML. Use convert argument to convert HTML to URL but did not “Vice-Versa”.

Ex: Convert= HTML_TO_URL

Interview questions

Q1. Why correlation required?

Sol: Whenever you record a use case request make contains some dynamic values. Which are unique user to user or session to session.

If you are passing the same value in the form of request server won't accept the same. To make our script work for multiple iterations we have to capture server generated from current session pass it into next request. So that your request will be accept by server.

Q2. If the LB or RB is keep on changing. What is your approach to capture the dynamic value?

Sol-1: If the dynamic LB is static use “Save offset” argument.

Sol-2: Use Flag system.

Flag System:

Any dynamic number is a part of LB we can use Flag System.

Ex-1: H1I Hero Welcome.

H2I Honda Welcome.

Sol: web_reg_save_param("pavan","LB/DIG=H#I","RB=Welcome",LAST);

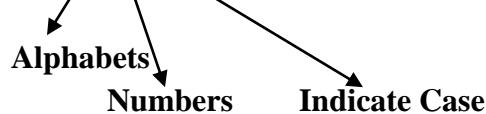
Ex-2: If you're LB contains dynamic alpha numeric characters use below solutions:

H1I Hero Welcome

H2I Honda Welcome

H3I Maruthi Welcome

Sol: web_reg_save_param("pavan","LB/**ALNUMIC**=H^I","RB=Welcome",LAST);



Sol-3: Use strtok() to capture dynamic value (refer page no 38)

Sol-4: Use lr_save_searched_string() (refer page no 33)

Sol-5: Use Regular Expressions (refer page no 47)

Q3. To capture entire response for a request we have to pass empty LB & empty RB?

Sol: By default LR variable can hold 256 characters. If you decide to capture more than 256 characters. We have to allocate memory to the LR variable.

Ex: web_reg_save_param("pavan","LB=","RB=",LAST);
 web_set_max_html_param_len("1024");

Note: 1024 is array size in bytes.

Q4. What is your approach if LB or RB isn't present for a dynamic value?

Sol: If the LB isn't available take any static text from the above line pass it as a LB & identify exact LB from the capture value & pass it as LB.

If the RB isn't present in the dynamic value take any static text from the below line & pass it as a RB & identify exact RB from the capture value & pass it as a RB.

Q5. How many ways you can find the LB, RB for a dynamic value?

Sol:

1. Generation log (which never change contains Response & Request).
2. Replay log: We have to execute script in Full mode then only we can identify the LB & RB dynamic value.
3. Snapshot View or Tree View.
4. Capture entire response using empty LB, empty RB & identify the LB & RB.
5. Using View Source Option in page view.

Q6. My correlation function is keep on failing?

Sol:

1. Invalid LB or RB.
2. Function location might be wrong.
3. May be your trying to capture more than 256 characters.
4. May be the dynamic value itself not available the response.

Q7. What is your approach to capture the dynamic value, if the dynamic value is not present in the response?

Sol: If the dynamic value is not present in the response speak with developer & try to understand why & how the dynamic value is generating then get the same functionality (Code) make it as a external function in VUGen & call the function to generate the dynamic value.

Ex:

OTP – whenever OTP is recorded in your script ask the developer to provide the code which can generate the OTP.

CAPTCHA – Ask the developer to disable the captcha functionality. Which will allowing to execute the script with multiple users.

(or)

You can ask to provide a number which can match for any image.

Q8. How many ways you can conduct correlation?**Sol:**

- 1) Auto Correlation.
- 2) Manual Correlation.
- 3) Right click on dynamic value & click on “Correlate button” [Not available in 11.5].
- 4) Navigate to the snapshot view & find out the dynamic value.
- 5) Right click on choose Correlate button [From 11.5 on words this option isn’t available].
- 6) Correlation studio.

Using insert service option [Not available from 11.5 on words].

Q9. If the dynamic value is generated by a Java Script file & the same value is not available in the response?

Sol: Speak with the developer understand how the dynamic value is generating. If it is generated by Java Script file usually at Client side.

Get the Java Script () create a “.js” file in extra files & call the function whenever you like to generate the dynamic value.

Q10. If the dynamic value isn’t present in the response or it is a hidden value?

Sol: Use web_add_cookie() to pass the dynamic value.

Q11. My first request itself keep on failing. What could be the reason for this?

Sol: Write web_add_header() to make it pass.

Parameterization:

Static input to the server or vary input to the server.

Note:

Which values you enter through keyboard those values should be parameterized. Your script should not contain any hard coded values.

Parameter Types:

1. File
2. Date & Time
3. Iteration Number
4. Random Number
5. Unique Number
6. VUserID
7. Group Name
8. Load Generator Name
9. Table
10. XML
11. User defect

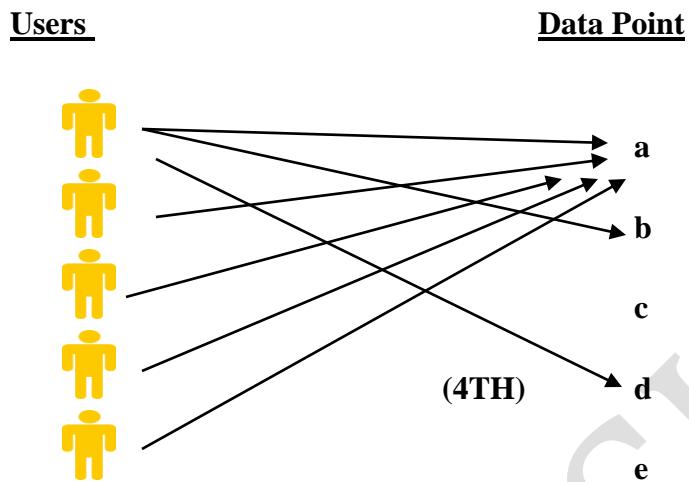
1. File:

Parameter Properties:

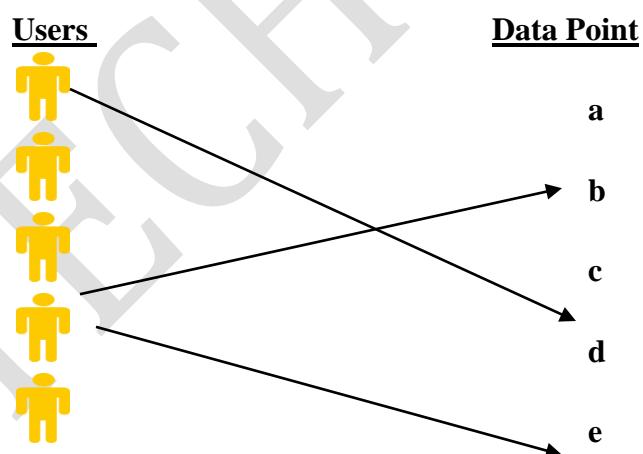
- 1) Sequential Each Iteration
- 2) Random Each Iteration
- 3) Unique Each Iteration
- 4) Sequential Each Occurrence
- 5) Random Each Occurrence
- 6) Unique Each Occurrence
- 7) Sequential Once
- 8) Random Once
- 9) Unique Once
- 10) Same line as some other parameter.

1) Sequential Each Iteration:

Sequential - It stands for every user as to pick the “First Data Point”.
Each Iteration - Data point update has to “Update on Every Iteration”.

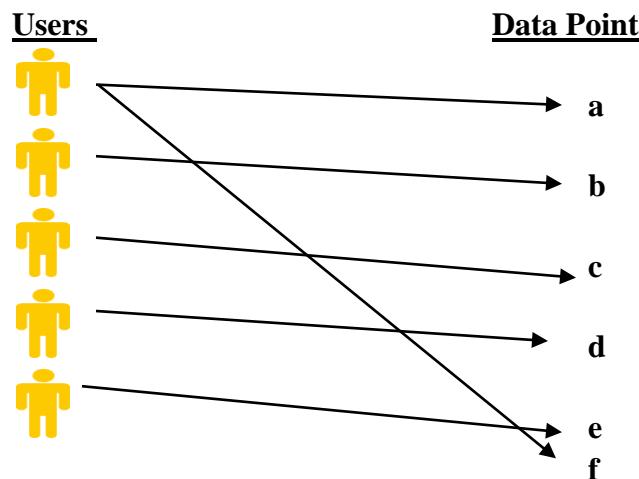
**2) Random Each Iteration:**

Users picks the values randomly.

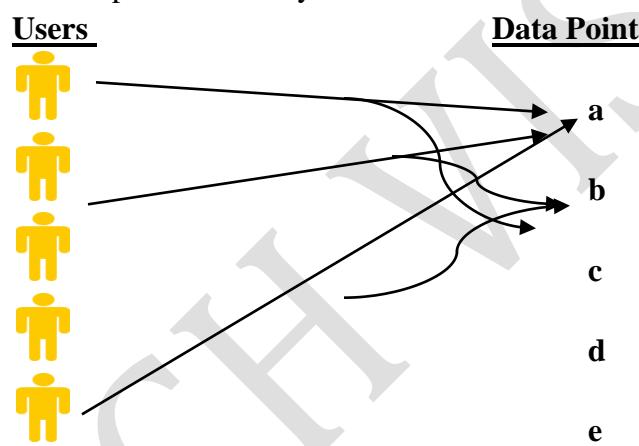
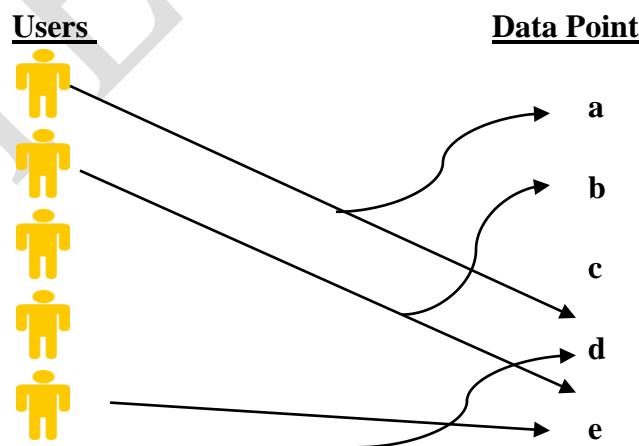


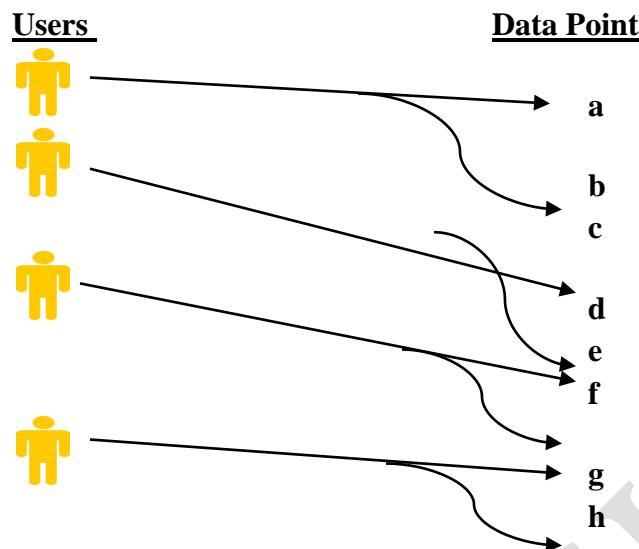
3) Unique Each Iteration:

Every user has to pick “First Unused Data point”.

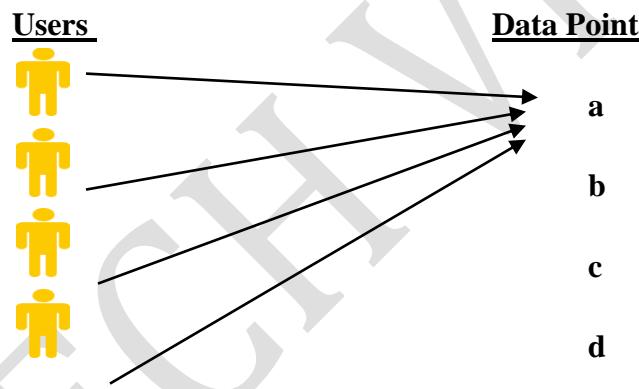
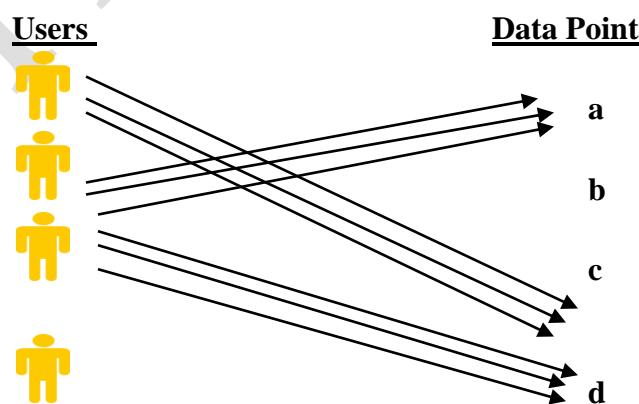
**4) Sequential Each Occurrence:**

Value updates on every occurrence.

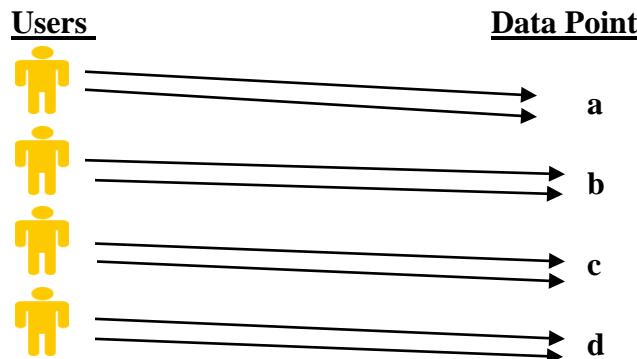
**5) Random Each Occurrence:**

6) Unique Each Occurrence:**7) Sequential Once:**

If the user picks any 1 of the data point. We have to continue the same value for all the iterations & occurrence.

**8) Random Once:**

9) Unique Once:



Note:

One to one mapping.

10) Same line as some other parameter:

In a scenario there is a dependent data (UN & PWD). We have to pick associated value from second data file. Using same line as some other parameter option. We can instruct the user to pick same row data point from second data file.

Complex Parameterization:

We can create one “.dat” file with multiple columns for dependent data. Complex parameterization allow you to handle test data in a proper way.

2. Date & Time:

Using Date & Time parameterization we can pick current date, future date, past date in require format.

Current Date:

Where you can choose the current date.

Note:

Data will pick from mission date from where it is executing.

Step1: Right click create parameter & choose Date & Time option.

Step2: Choose or create require parameter in parameter properties & click ok.

Future Date:

You can choose the future date from which mission script is executing.

Step1: Right click create parameter & choose Date & Time option.

Step2: Choose or create require parameter in parameter properties & click ok.

Step3: Choose offset date & provide the no. of dates.

Past Date:

Where you can choose the past date.

Step1: Right click create parameter & choose Date & Time option.

Step2: Choose or create require parameter in parameter properties & click ok.

Step3: Choose offset date & provide the no. of dates.

Step4: We have to choose the prior to current date option.

Note:

Using working days option, we can skip weekends.

3. Random Number:

As per the business scenario we have to generate the random number in **specified boundaries** in required format.

4. Unique Number:

We can generate the unique number using unique number parameterization by providing **Lower Boundary & Upper Boundary** in required format.

Data Wizard: Will allow you to pull the data from database with the help of connection string and query.

Simluate parameter: It will give an idea how the user will pick the values based on property in controller.

By number: Indicating column number from which column user supposed to pull the data.

By name: Indicating from which column user supposed to pull the data based on column name.

Column Delimeter: Values will be separated based on comma (,) or tab or space.

Note: If your data point contains comma, what is your approach to create .dat file?

Solution 1: Create tab or space based dat file instead of comma delimeter.

Solution 2: Keep the comma (,) in double quotes ("")

First Data line: Indicating the row number from which row user supposed to start picking the value.

Note: You can load 36000 data points in a script.

When out of values: This option is available for unique-each iteration/each occurrence. Instructing the user abort the execution or continue in cyclic manner or continue with the last value when there are no unused data points.

Block concepts in parameterization:

Block concepts is available for unique-each iteration/each occurrence property data points will be divided into blocks and allocated to users.

Option 1: Automatically allocate block size

-controller itself allocate the block size based on number of user data points.

Option 2: Manually we can calculate and decide the block size.

Interview Questions:

What is parameter file extension?

A) .dat

- You can create various types of files like **.dat** or **.xml** or **.csv**
- You can't create **.xls** as parameter file.
- You can create shared **.dat** file in between scripts

Verification points/checkpoints:

We have to verify whether we received proper response or not using below techniques

- | | | |
|------------------------------|---|-------------------|
| 1) Text verification points | } | Famous |
| 2) Image verification points | | Based On Protocol |
| 3) Response size | } | Optional |
| 4) HTTP status codes | | |
| 5) Correlation function | } | |

Text verification points:

web_reg_find("text=xyz",LAST);

Note 1: Which request you are planning to verify, on top of the request you have to write this function

Note 2: web_reg_find is a registered function. You have to write on top of request for verification

Note 3: Preferably use static text verification even you can use dynamic text for verification.

Checkpoint function:

web_reg_find("text=welcome","savecount=xyz",LAST);

After request, we have to write

```
If (strcmp(lr_eval_string("{xyz}"),"0")==0)
{
    Lr_output_message("text check failed");
}
else{
    Lr_output_message("text check passed");
}
```

Image verification:

It will verify the response based on image

Syntax: `web_image_check("xyz","src=c://image.gif");`

`web_image_check` is a non-registered function. We have to write after the request to make it work. You have to enable option “enable text and image verification” in runtime settings.

Q: How many ways I can find image name?

- 1) Open application → Right click → Choose view source
- 2) Navigate to snapshot
- 3) In generation log
- 4) Full mode of replay log

Q: How can I find how many times text is present in response?

A: “savecount” will let you know the count of the text

Q: Difference between web_reg_find and web_find?

Web_reg_find	Web_find
• It's a registered function	• It's a non-registered function
• Have to write before the request	• Have to write after the request
• Don't require to enable any option	• Have to enable text and image verification in runtime settings
• Current version we are using this function	• It's a deprecated function

Advanced Techniques

Note: We are going to use test script language (TSL) in load runner sometimes might require to write C or Java to make our business scenario work. By default, we will use C language.

Note: We have only one data type in LR that is “string”

Note: We don't require to declare LR variable.

C Data Types:

- Int,
- char,
- double,
- float,
- flag,
- long,
- short,
- Boolean etc

Scripting Challenges

Q: What are the scripting challenges you faced so far?

(or)

What kind of C techniques you used so far?

Scenario 1:

How to convert an integer value to LR variable?

Solution: “itoa” → It converts integer to string

Lr_eval_string → It reads the value from a variable

Lr_output_message → It displays the message

Example:

```
Int x=0;
```

```
Char abc[100];
```

```
Itoa(x,abc,10); //Converting int to string
```

```
Lr_save_string(abc,"final"); //Assigning value to lr variable
```

```
Lr_output_message("%s", lr_eval_string("{final}"));
```

Scenario 2:

How to generate random value from an array?

Solution:

```
Lr_save_string(lr_paramarr_random("cSource"),"randSource");
```

Scenario 3:

How to compare two LR variables?

Solution:

```
lr_save_string(lr_paramarr_random("c_source"),"Ran_Source");
    lr_save_string(lr_paramarr_random("c_desti"),"Ran_Dest");
    if(strcmp(lr_eval_string("{Ran_Source}"),lr_eval_string("{Ran_Dest}"))==0)
    {
        lr_output_message("SOURCE & DESTINATION ARE SAME, CANT BOOK
TICKET");
        lr_abort();
    }
    else
    {
        lr_output_message("SOURCE & DESTINATION ARE NOT SAME, CAN
BOOK TICKET");
    }
```

Scenario 4:**How to compare two C variables?****Solution:**

```
lr_save_string(lr_paramarr_random("c_source"),"rand_sorc");
lr_save_string(lr_paramarr_random("c_desti"),"rand_dest");
strcpy(source,lr_eval_string("{rand_sorc}"));
lr_output_message(source);
strcpy(desti,lr_eval_string("{rand_dest}"));
lr_output_message(desti);

if(strcmp(source,desti)==0)
{
    lr_output_message("SOURCE & DESTINATIONS ARE SAME, CANT BOOK TICKETS");
}
else
{
    lr_output_message("SOURCE& DESTINATIONS ARE NOT SAME, CAN BOOK TICKETS");
}
```

Scenario 5:

Working for a banking application. Capture the account holder balance and add 10 rs to balance and substitute in the next request.

Solution:

```
y = atoi(lr_eval_string("{c_sid}"));
z=x+y;
lr_output_message("%d",z);
itoa(z,final, 10);
lr_output_message(final);
lr_save_string(final,"final_balance");
```

Scenario 6:**How to use string tokenizer functionality?****Or****How to split a string into multiple strings?****Solution:**

```
char* abc;  
char xyz[100];
```

-----RESPONSE BODY-----

```
strcpy(xyz,lr_eval_string("{c_fno}"));  
abc=(char*)strtok(xyz,";");  
lr_output_message(abc);  
lr_save_string(abc,"final");
```

Scenario 7:**How to split a string into multiple strings?****Solution:**

```
char path[100];  
char separators[] = ":";  
char* token;
```

-----RESPONSE BODY-----

```
strcpy(path,lr_eval_string("{c_fno}"));  
token = (char *)strtok(path, separators); // Get the first token  
if (!token)  
{  
    lr_output_message ("No tokens found in string!");  
    return( -1 );  
}  
while (token != NULL )  
{ // While valid tokens are returned  
    lr_output_message ("%s", token );  
    token = (char *)strtok(NULL, separators); // Get the next token  
}
```

File operations:

Scenario 8:

How to write a value to a local file?

Solution:

```
long file;
```

-----RESPONSE BODY-----

```
file=fopen("D:\\vny_src.txt","a+");
fprintf(file,"SESSION ID : %s\\n",lr_eval_string("{c_sid}"));
fclose(file);
```

Scenario 9:

How to write multiple values/array values into local file?

Or

How to write a for loop?

Solution:

```
long file;
```

```
int i,cnt;
```

```
char* temp;
```

-----RESPONSE BODY-----

```
cnt=atoi(lr_eval_string("{c_desti_count}"));
for(i=1;i<=cnt;i++)
{
    temp=lr_paramarr_idx("c_desti",i);
    lr_save_string(temp,"F_dest");
    file=fopen("D:\\vny_dest.txt","a+");
    fprintf(file,"SOURCE IS : %s\\n",lr_eval_string("{F_dest}"));
    fclose(file);
}
```

Scenario 10

How to generate a random value from an array without using lr_paramarr_random function?

Or

How to concatenate two lr variables?

Or

How to use atoi and itoa functions?

Solution:

```
int cnt,x;
char* temp;
char abc[10];
-----RESPONSE BODY-----
cnt=atoi(lr_eval_string("{c_desti_count}"));
x=1+rand()%cnt;
lr_output_message("%d",x);
itoa(x,abc,10);
lr_save_string(abc,"F_dest");
temp=lr_eval_string(lr_eval_string("{c_desti_{F_dest}}"));
lr_save_string(temp,"final");
```

Scenario 11

Working for a banking application. Capture the account holder balance and add 10.25 rs to balance and substitute in the next request.

Solution:

```
float x=10.75,y,z;
char final[100];
-----RESPONSE BODY-----
y = atof(lr_eval_string("{c_sid}"));
lr_output_message("%.2f",y);
z=x+y;
lr_output_message("%.2f",z);
sprintf(final,"%2f",z);
lr_output_message(final);
lr_save_string(final,"final_balance");
```

Scenario 12

How to download a pdf/doc/xls from an application?

Or

How to use VuserID and iteration number parameter?

Or

How to generate time stamp in milli seconds?

Or

How to create a buffered variable?

Or

How to create unique file naming convention?

Solution:

```
long file;
unsigned long paramlen;
char* newparam;
web_save_timestamp_param("tstamp",LAST);
file=fopen(lr_eval_string("D:\\anand_{puser_id}_{p_iteration}_{tstamp}.pdf"),"wb");
lr_eval_string_ext("{vinay}",strlen("{vinay}"),&newparam,&paramlen,0,0,-1);
fwrite(newparam,paramlen,1,file);
fclose(file);
```

Pre-defined functions:**LR Functions:**

LR functions you can use across the protocols.

1) lr_abort();

This function use for script abort purpose .

example:

```
if (strcmp(lr_eval_string("{ramana}"), "0") == 0)
{
    lr_output_message("Login Failed");
    lr_abort();
}
else
{
    lr_output_message("Login PASSSED");
}
```

2) lr_exit();

Exits the iteration or action.

Arguments:

LR_EXIT_USER
LR_EXIT_ACTION_AND_CONTINUE
LR_EXIT_MAIN_ITERATION_AND_CONTINUE
LR_EXIT_ITERATION_AND_CONTINUE
LR_EXIT_VUSER_AFTER_ITERATION.

Example:

```

if (strcmp(lr_eval_string("{ramana}"), "0") == 0)

{
    lr_output_message("Login Failed");
    lr_exit(LR_EXIT_MAIN_ITERATION_AND_CONTINUE, LR_FAIL); // here v
can change the above arguments.....
}

else
{
    lr_output_message("Login PASSSED");
}

```

3) lr_start_transaction() & lr_stop_transaction():

Allow you to measure the response time for a page/request.

```

lr_start_transaction("launch"); //here launch is txn name. this statement write on top of request
lr_stop_transaction("launch"); //this statement write after the request

```

4) lr_stop_transaction() & lr_resume_transaction():

Transaction will be stopped and resumed based on requirement

5) lr_start_timer() & lr_end_timer():

This function will allow you to calculate wasted time. Nowadays you don't require to calculate wasted time which will be measured by LR itself.

```

double time_elapsed, waste;
merc_timer_handle_t timer;
timer = lr_start_timer(); // this should write after the web function. or after start
transaction
time_elapsed = lr_end_timer(timer); // this should after the web_reg_find function or
before ending transaction

waste = time_elapsed * 1000; // Convert to millisecond.s

lr_wasted_time(time_elapsed); /* Remove the time the checks took from the
transaction */
lr_output_message("%lf",time_elapsed);

```

6) lr_save_string():

It assigns a value to LR variable

Ex: lr_save_string("abc","xyz");

7) lr_save_int():

It assigns integer value in to LR variable.
this also used instead of itoa function.
`lr_save_int(x, "xyz"); // here X is a integer`

8) lr_eval_string():

It evaluate the value after embedding parameter or it reads the value.

9) lr_paramarr_random():

It generates the random value from an array.
`lr_paramarr_random("source");`

10) lr_paramarr_idx():

Returns the value of the parameter at a specified location
`lr_paramarr_idx("source",10); // it reads 10 th value of source.`

11) lr_save_datetime():

Assigns the current date and time to a parameter.

Ex:

`lr_save_datetime("%m/%d/%Y %H:%M", DATE_NOW, "currDateTime"); // This statement write in output message area(after the request)`

12) lr_set_debug_message():

Changes the message level for a request

Syntax:

```
lr_set_debug_message(LR_MSG_CLASS_EXTENDED_LOG  
LR_MSG_CLASS_FULL_TRACE, LR_SWITCH_ON );  
web_url()  
---
```

```
lr_set_debug_message(LR_MSG_CLASS_EXTENDED_LOG  
LR_MSG_CLASS_FULL_TRACE, LR_SWITCH_OFF );
```

Note: Here we need to set standard mode in Runtime Settings.

13) lr_db_connect():

It connects to database with the help of connection statements

14) lr_db_disconnect():

It disconnects from the database

15) lr_db_executeSQLstatements():

Submits SQL statements to database

16) lr_continue_on_error():

On error, scripts use one of these options for continuation:

Syntax:

```
lr_continue_on_error(0); -----> no option  
lr_continue_on_error(1); -----> continue  
lr_continue_on_error(2); -----> skip to next Action  
lr_continue_on_error(3); -----> skip to next iteration  
lr_continue_on_error(4); -----> end user
```

17) lr_load_dll():

The lr_load_dll function loads a DLL (Windows) or shared object (Linux) allowing you to call an external function when replaying using the C interpreter.

Once you load the DLL, you can call any function defined in the DLL, without having to declare it. You can specify a full path for the DLL.

Syntax:

```
lr_load_dll("user32.dll");
```

18) lr_out_message():

It's sends a message to log file, output windows and summary report

19) lr_log_message():

Sends a message to logfile to the application management agent logfile (or) vuser log.

20) lr_message():

Sends a message to the log and o/p window.

21) lr_error_message():

Sends an error message with location details to the output windows, log files, and other test report summaries.

22) lr_get_vuser_ip():

Returns the ip address of the user

Syntax:

```
char* ip;  
Ip= lr_get_vuser_ip();
```

23) lr_get_host_name():

Returns the name of the host

Syntax:

```
char* name;  
Ip= lr_get_host_name();
```

24) lr_get_master_host_name():

Retrieve the controller machine name

25) lr_whoami():

Returns the information about vuser

26) lr_vuser_status_message():

Sends a message to vuser status area in controller

27) lr_save_searched_string():

It will captures substring from a main string

Syntax:

ex: for flight number 300;365;05/08/2016

```
char abc[100];  
//this code write after the request  
no option  
strcpy(abc,lr_eval_string("{outboundFlight}));
```

```
lr_save_searched_string(abc, strlen(abc),  
1,";", //Searched for 2nd occurrence of ";"  
0, //Indicates no skip after ";"  
3, //Captures next 3 characters  
"year"); // here year is final value (or )abc saved in year
```

Web Functions:

These functions we can use only in web protocol

GET	POST
1. Not a secured request	1. It's a secured request
2. It will send small amount of information to server	2. Sends large amount of information
3. It can be available in history, cached and bookmarks.	3. It can't be available in history, cached and bookmarks.
4. To retrieve the information	4. To submit the data to server

Note: Method types → Get, Post, Put, Options etc

Note: “Resource = 0” indicating it’s a html resource it’s downloading a html page.

1) web_url():

To launch the application or load the specified URL.

2) web_set_max_html_param_length():

Sets the maximum length of any html string that can be retrieved and stored in a variable.

Syntax: web_set_max_html_param_length("1024");

3) web_save_timestamp_param():

Save the timestamp to a parameter. It generates a timestamp in milliseconds which is a 13 digit number.

Syntax: web_save_timestamp_param("tstamp",LAST);
lr_output_message("%s", lr_eval_string("{tstamp}"));

4) web_reg_save_param():

Allow you to capture dynamic value from response.

5) web_reg_save_param_ex():

Allow you to capture dynamic value which is introduced from 11.0 version. It has different parameters.

web_reg_save_param	web_reg_save_param_ex
1. Parameter name	1. Parameter name = xyz
2. Before ORD = 1	2. Now ORDINAL = 1
3. Before search = header/body	3. Now scope = body
4. RelframeID = 1.2.1	4. RequestURL = URL
	5. They added content type, data execution format (def) arguments
	6. They deprecated convert argument.

6) web_reg_find():

For text verification purpose.

Syntax: web_reg_find("Text=Welcome","SaveCount=Welcome_Count", LAST);

7) web_find():

Text verification purpose. It is deprecated.

8) web_image_check():

Image verification purpose.

Syntax: web_image_check("Text=Welcome", "src=D:\\xyz.gif", LAST);

9) web_set_proxy():

Specifies that all requests should be redirected to a specific proxy server.

Syntax: web_set_proxy(10.10.10.10:8080);

Note: Default port number of https request → 443

Default port number of http request → 8080

10) web_submit_data():

It is an unconditional context less request which will submit data in form of item data.

11) web_submit_form():

It submits the form and context based request. Submit data in the form of item data.

Note: Context based means request will be dependent on previous response.

Note: Usually we are using web_submit_data option to record scenario.

12) web_get_int_property():

Using this function, you can verify response whether it is proper or not based on HTTP status codes and download size

Syntax:

```
int HttpRetCode = web_get_int_property(HTTP_INFO_RETURN_CODE);
lr_output_message("%d", HttpRetCode);
if(HttpRetCode == 200){
    lr_log_message("Script passed");
} else{
    lr_log_message("Script failed");
    lr_abort();}
```

Q: How to capture download size for a response?

A: web_get_int_property(HTTP_INFO_DOWNLOAD_SIZE);

13) web_set_user():

For all NTLM (Native Windows Authentication) Applications, we should submit the credentials against web_server.

Syntax:

```
web_set_user("CA:\\Jojo","Bean", HTTP://webtours.kword.com:80);
```

14) web_set_certificate():

Causes a script to use a specific certificate that is listed in internet explorer registry

Syntax: web_set_certificate ("2");

Extension: .prm/.cer/.pem

15) web_set_socket_option():

Setting the socket level option for a request

Syntax:

```
web_set_socket_option("SSL_VERSION", "TSL1.1");
```

Instead of using above function, you can execute the script using wininet engine but which is not preferable.

16) web_cleanup_cookies() & web_cache_cleanup():

These functions will allow you to cleanup cookie files and cache files from browser. Have to write these functions at end of the script.

17) web_convert_param():

Which will convert a string html to url and url to plain vice versa.

Syntax:

```
web_convert_param("HTML", "SourceEncoding=HTML", "TargetEncoding=URL", LAST );
web_convert_param("HTML1","SourceEncoding=HTML", "TargetEncoding=PLAIN", LAST );
web_convert_param("Plaintext", "SourceEncoding=HTML","TargetEncoding=URL", LAST );
```

18) web_add_header():

Scenario 1: My first/any request keep on failing even after conducting correlation and parameterization

Solution: Write web_add_header() function on top of request.

Step 1: Verify correlation and parameterization.

Step 2: Verify whether the page is opening manually or not. If it is opening, then problem is with header.

Step 3: Verify whether we need to add any SSL.

Q: How to add header information?**Sol:****Step 1:** Right click on request and choose snapshot view**Step 2:** Verify record time and replay time headers**Step 3:** If there is a difference, please add information using web_add_header() function.**19) web_custom_request():**

It will submit the data in the form of body.

Scenario 1: Based on my previous input, number of fields are changing in current page.**Scenario 2:** When I am accessing application in morning, it reported 10 records. In the evening 15 records, next day 20 records. How to make it work for anytime?**Scenario 3:** How to write web_custom_request()**Scenario 4:** Client asked me to book random number of tickets for every user but based on random number, number of passenger fields are changing in payment page. How to make my script work for vary number of passengers?**Sol:**

```

int xyz;
char abc[100], lmn[100];
xyz=atoi(lr_eval_string("{prandpass}"));
for(i=1;i<=xyz;i++)
{
    sprintf(abc,"pass%d=", i);           //pass1=
    strcat(abc,lr_eval_string("{ppass}")); //pass1=anand
    strcat(abc,"&");                  //pass1=anand&
    strcat(lmn,abc);  }
lr_save_string(lmn,"r_no_pass");

```

Q: How to read a value from a local file?**Sol:**

```

char buffer[100];
long file;
file = fopen("D:\\xyz.txt","r+");
freadf(buffer,sizeof(char), 20, file);
lr_save_string(buffer, "read_value");
return 0;

```

Q: How to write a do while condition?**Sol:**

```

do{   web_reg_find("Text-departure","savecount=xyz",LAST);
web_url("Search Flight Button",
}while(atoi(lr_eval_string("{xyz}"))==0);

```

C Functions:

1) strcpy()

Copies one string to another

Syntax:

```
Char hi[10];  
Strcpy(hi,"747");
```

2) strcmp()

Compares two strings

Syntax:

```
Strcmp(hi,rao);
```

3) strcat()

Concatenates two strings

Syntax:

```
Strcat(abc,"&");
```

4) strlen()

Returns length of the string

Syntax:

```
Strlen(abc);
```

5) strstr()

Captures substring from main string

6) strtok()

Tokenize the string based on delimiter

Syntax:

```
xyz = (char*) strtok(abc,";");
```

7) strncpy()

Copies n number of characters to another

8) sprint()

Writes a formatted output to a string

File Operation Functions:

1) fopen()

Opens a new/existing file

2) fwrite()

Writes an unformatted data to file

3) fprintf()

Prints a formatted value to a file

4) fread()

To read the data from a file

5) fclose()

Closes the file

User Defined Functions:

Function: Set of reusable statements is called function which has to return something.

Syntax:

```
vuser_init()  
{  
Return 0;  
}
```

Note: Dividing your script statements into multiple functions is called modular approach

Q: How to create dynamic transaction name convention?

Or

How to declare a function?

Or

How to call a function?

Or

How to overwrite the .dat file properties using lr_next_row() and lr_advance_param() function?

Sol:

```
lr_start_transaction(lr_eval_string("login_{puser}"));
web_reg_find("text=welcome", "savecount=xyz",last);
web_submit_data("login.pl"           //request

if(strcmp(lr_eval_string("{xyz}"),"0")==0)
{
lr_output_message("text check failed");
lr_end_transaction(lr_eval_string("login_{puser}"),last);
login();
}
else
{
lr_output_message("text check passed");
lr_end_transaction(lr_eval_string("login_{puser}"),last);
}
```

Q: How to create a header file?

Or

How to use function across the project?

Or

Example for modular approach

Or

How to create a framework in LR?

Sol:

Step 1: Open the notepad.

Step 2: Write the statements into notepad.

Step 3: Save it as ".h" file.

Step 4: Keep it in shared drive.

```
#include "D:\\AR_H.h"
```

(In this scenario we are saving file in specific drive)

Step 5: If you are saving header file into bin directory, then you don't required to mention path

Note: Even you can include this file into extra files.

Header file code:

```
#ifdef _AR_H  
#define _AR_H  
  
int sum(int a, int b, int c)  
{  
    return a+b+c;  
}  
#endif      //AR_H
```

Calling from Script:

```
#include AR_H.h  
{  
int z;  
z=sum(10,20,30);  
lr_output_message("%d", z);  
}
```

Note: vuser_init() contains logic procedure.
action() contains business procedures.
vuser_end() contains logoff procedure.

Note: You can iterate only action() part.

Note: You can create multiple actions.

Scenario 1:

In one of my business scenario, user has to login to application once and has to perform business scenario 10 times, then logoff from application

Sol:

To achieve above scenario:

Record:

Login procedure into vuser_init()

Business procedures into action().

Logoff procedure into vuser_end().

Scenario 2:

User has to login to application, book and cancel tickets for multiple times. Every iteration has to perform login and logoff

Sol:

Everything has to be recorded into action().

Scenario 3:**How to create multiple actions?****Sol 1:**

While recording itself we can create multiple actions.

Sol 2:

We can create new action using create new action option.

Note: If the actions are not calling from run logic, those will be faded out.

Scenario 4:

In one of my scenarios, have to book tickets for 10 times and cancel the ticket for 5 times

Or

Have to approve the loan 100 times and cancel the loan 40 times

By creating blocks in the **runlogic**,

we can achieve the above scenario

Step 1: Go to run logic.

Step 2: Create insert block.

Step 3: Insert actions into block.

Sol 1:

Create two blocks. Insert book action into 1st block, cancel action into 2nd block. Provide block level iteration.

Sol 2:

Using for loop also, you can achieve above scenario.

Scenario 5:

In one of my loan application, 70% loan has to be approved. 30% loan has to be rejected irrespective of the number of iterations.

Sol:

To achieve above scenario, I will insert both of the actions into one block and provide the % by choosing random property.

Note: Script extension is .usr

Default script folder contains below files:

Data folder, vuser_init, action, end notepad, global.h, default.cfg, default.usp files, breakpoints file, custom_body_variables, ReplaySummaryReport, .usr file

Scenario 6:**Scrubbing method for multiple users?****or****I have received 10000 users some of them are not working. How to segregate which are working, which are not working?**

We need to do parameterization for user name and password

long file;

long file1;

web_reg_find("Text=Welcome","Savecount=ramana",LAST);
web_submit_data("login.pl",

```
if (strcmp(lr_eval_string("{ ramana }"),"0")==0)
{
    lr_output_message("Text check failed");
    file= fopen("E:\\invalid.txt","a+");
    fprintf(file,"invalid:%s\\n",lr_eval_string("{ pUser }"));
    fclose(file);
    lr_exit(LR_EXIT_ITERATION_AND_CONTINUE,LR_FAIL);// SCRIPT FAILED
MESSAGE WILL DISPLAYED, BUT SCRIPT PASSED FOR OTHER ITERATIONS.
```

}

else{

```
    lr_output_message("text check passed");
    file1= fopen("E:valid.txt","a+");
    fprintf(file1,"valid:%s\\n",lr_eval_string("{ pUser }"));
    fclose(file1);
}
```

Recording Options:

Shortcut (Ctrl + F7)

1. Recording:

We do have two modes in HTTP/HTML protocol

- 1) HTML Mode
- 2) URL Mode

HTML Mode: Generates a separate step for every user action even you can record non HTML resources.

HTML Advanced: Using advanced techniques, you can record web_submit_form or web_submit_data/

Usually we will use web_submit_data along with record within script.

Advantages:

- Easy to understand the script.
- Easy to maintain the script.
- Very less number of lines of code.
- Generate script for HTML and non HTML resources.
- Script never fails for non HTML resources. If any one of non HTML resource is not available, it will throw a warning like resource unavailable HTTP status code 403.

URL Mode: Records not only user actions, even it records server side resources too. It will generate web_custom_request and web_url.

- You can find huge number of lines of code.
- It's very difficult to maintain the script.
- It will record non HTML resources in the form of concurrent groups.
- URL mode prefers for non-browser specific applications.

If any one of non HTML resource is not available or failed to download, the same script will be aborted by throwing an error.

If application having java script files, prefer to use URL mode.

Q: In which scenario you are going to use URL mode?

Scenario 1: My application is a non browser specific application. So we use URL mode to make ensure that all resources should be downloaded.

Scenario 2: My application downloading lot of java files for every page. So we use URL mode to make ensure that every java script file has to be downloaded.

Note: Request will be recorded in between web_concurrent_start and web_concurrent_end in between both of the function we can't write anything like think time, transaction name.

2. Script:

Script option will allow you to generate the auto think time, limits number of lines in a script.

3. Protocols:

Displays opted/chosen protocol.

4. Code generation:

Allow you to conduct auto scan for dynamic values.

5. Configuration:

Dealing with only auto correlation by specifying record scan, replay scan, rules scan, correlation function, min and max length of dynamic value.

6. Rules or Correlation Studio (till 11.0):

We can create our own rules for common dynamic values. Those rules will be used across the project.

Step 1: Create new application.

Step 2: Create new rule by providing LB, RB, param.

Note: We have to select boundary based scan type

Step 3: Test the rule.

Step 4: You can export the rule by clicking export button.

Step 5: You can import the rule by clicking import button.

Note: Correlation rule extension is “.cor”

7. Advanced:

We can generate auto text verification function using this option.

We can generate auto headers.

8. Mapping and Filtering:

Q: I am recording a business scenario but I failed to record the events.

Solution:

If you failed to launch the application, then change the DEP (Data Execution Prevention) settings in my computer properties.

→My Computer

→Properties

→Advanced System Settings

→Advanced Performance Settings

→DEP

→Choose turn on DEP for all programs.

Q: Application is launching, events are generating but script is not available.

Solution 1: Problem might be SSL encryption.

Solution 2: Problem might be with browser compatibility.

Q: Events and Scripts both are not generating

Solution 1: Problem might be with protocol.

Solution 2: Change the capture level, socket level to wininet.

Solution 3: Verify whether auto SSL is enabled or not.

Solution 4: Verify the port numbers.

Solution 5: Still if you are not able to record, record the communication using Fiddler, convert Fiddler request to Vugen request.

Socket level data: For all web based applications, we have to use this option.

Wininet data: For all NTLM based application or window based applications, we have to use this option.

Q: How many ways you can conduct correlation?

Solution 1: Through Design Studio.

Solution 2: By right clicking on dynamic value and choosing correlation selection.

Solution 3: Navigate to snapshot view → tree view → select dynamic value from response → choose create correlation.

Solution 4: Navigate to snapshot view → find out dynamic value → choose create correlation.

Solution 5: From generation log.

Q: How many places you can find server response?

- Full mode of replay log.
- Generation log.
- Snapshot view.
- Tree view.
- Empty LB and empty RB function.

6. Test Data:

Usually you will get test data from DBA's. While developing the script itself, we have to prepare a test data requirement sheet which contains which required for every use case how much test data you required and segregate which is reusable and which is not.

Sometimes you can generate test data using LR script itself.

Example: You can create username and password if the signup functionality is available in the application. If the functionality is not available to generate test data, we have to request DBA to provide it.

If the test data is not reusable, we have to request DBA to create database restoration point or take the flashback of database.

Scenario 1: Once the test is completed, we have to request DBA to change DB to previous restoration point. So that data will be available for next test.

Scenario 2: Once the test is completed, we have to request DBA, flashback the database or load previous database instance.

Scenario 3:

Q: One of the script is generating purchase order number and second script is processing same purchase order number. What is your approach to design the script?

Or

How to pass value from one script to another?

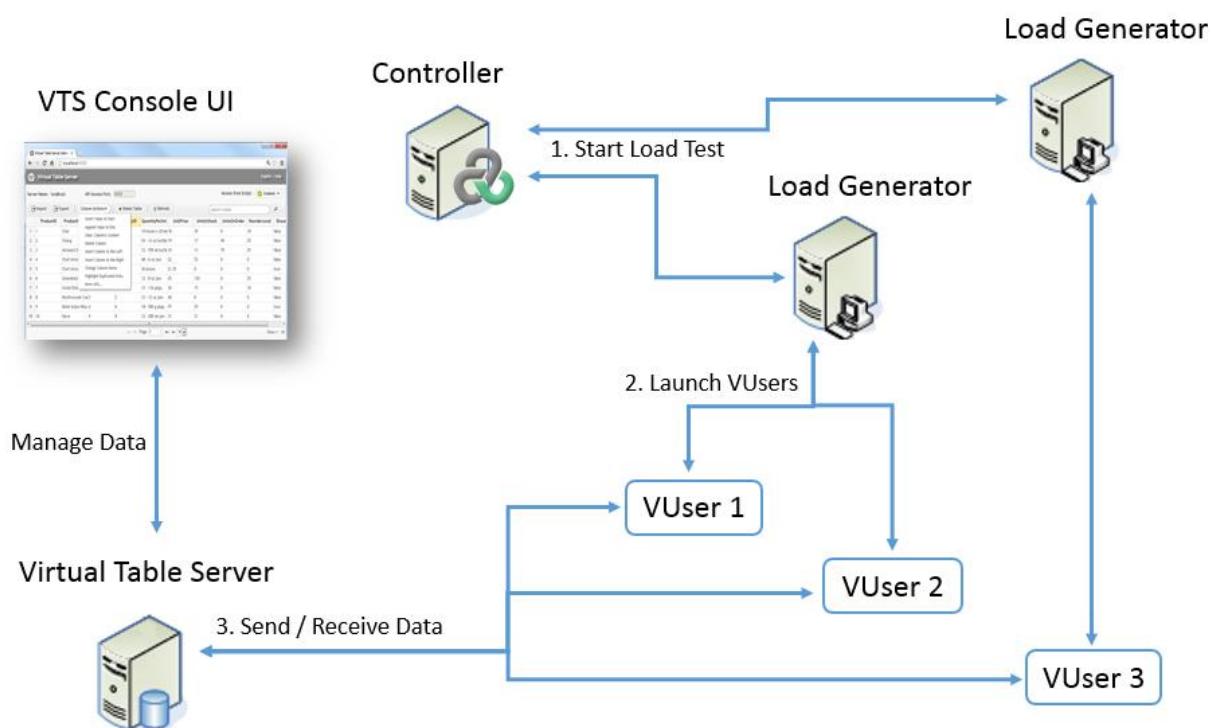
Solution 1: Create two actions in one script for both of the scenarios. Capture purchase order number from 1st action and pass into the second action.

Solution 2:**Data Staging:**

Execute the first script with multiple users in controller. Write purchase order numbers into local file before starting actual test. Copy purchase order numbers from local file. Load into second script so that both of the script will execute simultaneously.

Solution 3:**VTS (Virtual Table Server):**

Virtual Table Server is a tool that enables sharing of test data and parameters between LoadRunner Virtual Users (VUsers). Usually performance testers use parameterization in their scripts, but in Virtual Table Server it acts as centralized repository to store data which can be shared between VUsers during the test execution.



Regular Expressions:

Whenever you have dynamic LB & RB, we can use regular expressions. This function works on expression but not on boundaries.

Steps to construct regular expressions:

- Step 1:** Copy the LB & RB from generation log.
- Step 2:** Construct the expression.
- Step 3:** Validate in www.rubular.com
- Step 4:** Write web_reg_save_param_regex() function.

Example:

```
web_reg_save_param_regex("parameter name = reg_exp",
"regex=name =\" u[a-z]er[A-Z]ession\" \\\\value=(.*?)" , LAST);
```

u[a-z]er[A-Z]ession → Function searches for userSession based on this regular expression.

[a-z] → Skips any single character in the range a-z

[A-Z] → Skips any single character in the range A-Z

Note: Refer regex quick reference in www.rubular.com to construct regular expression.

Fiddler:

Fiddler is a web debugging tool or stripping tool.

Fiddler versions:

2.2, 2.4, 2.5, 2.6, 4.4

Q: Why do we fail to launch the application in vugen sometimes?

Reasons:

- DEP settings
- Protocol selection
- Browser compatibility
- Wininet level/socket level
- Fiddler

When we failed to record some objects using vugen, we can use fiddler to record the communication.

Fiddler file extension is “.saz”

Q: In one of my application, we failed to record some of the get and post requests.

Solution: We used fiddler to develop the script

Scenario 1: How to convert fiddler get request to vugen get request?

Solution:

- Select the request in fiddler
- Right click and copy just url
- Write web_url in vugen and construct a request

Scenario 2: How to convert post requests?

Solution:

- Select the request in fiddler
- Right click and copy just url
- Go to inspectors panel
- Choose text view and copy the content

Using above just url, content and construct the web_custom_request

In current version of LR, we can directly open the fiddler file.

CONTROLLER

Note: Controller extension is “.lrs”.

Manual scenario: Manually we have to design the scenario by providing ramp up, ramp down, duration, number of users to generate the anticipated load against the application.

Percentage mode: It will distribute the load in between use cases in terms of percentage.

Goal oriented scenario: Controller itself will design the scenario as per the goal.

- Controller having 3tabs:
 1. Design tab.
 2. Run tab.
 3. Diagnostics tab.

1. Design tab: where you can design the scenario.

Note: scenario extension is “.lrs”.

- **Ramp up:** gradually increasing the load against application.
- **Duration:** indicates test duration which extends ramp up and ramp down time. this is called “duration”, “standalone”, “steady time”.
- **Ramp down:** gradually reducing load from application.
- **Elapsed time:** this is the test duration time which includes ramp up, steady time, ramp down.
- **Throughput:** bits received from server.
- **Hits per second:** per second how many hits happening.
- **Schedule by scenario:** it considers all the scripts like a scenario and configure ramp up, ramp down, duration.

(or)

Consider as a scenario by sharing ramp up, ramp down, duration.

- **Schedule by group:** every script consider as a different scenario by having their own ramp up, ramp down, duration.
- **Real world schedule:** allow you to create multiple actions (ramp up, ramp down, duration).
- **Basic schedule:** you have only basic action under this option (one ramp up, one ramp down).

2. Run tab:

- **User status:**

Down
Pending
Initialization
Ready
Run
Rendezvous
Passed
Failed
Error
Gradual exiting
Exiting
Stopped

- **Gradual exiting:** whenever test duration completed by the time user are in mid of iteration. So those users will be moved to gradual exiting once the iteration is finished users will be come out from test.

Note: elapsed time stats whenever you hit the start button.

3. Diagnostics tab: By default web page diagnostics are available at free of cost. We have to buy license for java, Siebel, sap, oracle etc.

Manual scenario:

- **Check list:**

1. Choose manual scenario.
2. Push the script into controller.
3. Choose schedule by scenario (or) schedule by group.
4. Choose real world (or) basic schedule.
5. Assign the quantity.
6. Assign load generators.
7. Verify connectivity with LG's.
8. Configure SLA's (if required).
9. Configure runtimes settings for every script.
10. Set the result path.
11. Provide ramp up, ramp down, duration.

Note: whenever situations demands like every script should run for multiple durations with vary ramp up's then you can choose schedule by group.

Differences between request and hit:

User action is a request.

Successful request is a hit.

Note: One request may contains multiple hits.

- **Goal oriented scenario:** controller itself will design the scenario according to goal.
- **Check list:**
 1. Choose goal oriented scenario.
 2. Push the scripts into controller.
 3. Click on edit scenario goal.
 4. Provide profile name.
 5. Provide goal type and threshold point.
 6. Provide max and min users.
 7. Provide duration once it reaches the goal.
 8. Configure notifications if it is not able to reach goal.
 9. Distribute load in terms of percentage.
 10. Assign LG and check the connectivity.
 11. Configure SLA's (if required).
 12. Set the result path.

Note: Goal types

1. Virtual users.
2. Transaction per second.
3. Hits per second.
4. Response time.
5. Pages per minute.

IP Spoofing:

Masking the original IP address and using the different IP address is called IP spoofing.

Whenever load balancer is not functioning. Due to the request source of IP address. We have to mask the original IP address and make sure that every user has to use different IP address.

Process: In realistic environment every end users using different IP address to access the application but in LR environment all the users are invoking from same LG and same IP which is not realistic.

Step1: Request IT infrastructure guys to provide dynamic IP address

Step2: Ask them to configure these IP address in DHCP (dynamic host control protocol) server.

Step3: We have to configure these IP address in load generator using IP wizard option.

Step4: Enable IP spoofing option in controller.

Load balancing: Load balancer is a URL, which will distribute load in between web instances.

As part of performance testing you will receive two kinds of URL.

1. **Direct URL:** It will access web instance directly.
2. **Load balancer URL:** which will distribute the load in between web instances.

We do have two types of load balancer

1. Hardware load balancer
2. Software load balancer

Among the above, hardware load balancer is accurate and good
Load balancer URL will distribute the load based on below algorithms.

1. IP sticky
2. Least connections
3. Round robin
4. Round robin
5. Least load

Q: While running the test how to verify whether load balancing is happening properly are not?

A: While running the test we have to monitor resource utilization (C.P.U and Memory) each and every instance. If all the instances are using same level of C.P.U and memory, then we can conclude that LB is happening properly. If any one of the instance reporting less C.P.U memory utilization then we have to conclude LB is not happening properly.

Q: What are the actions you can perform while running the test?

A:

1. Add the users.
2. Delete or kill the users.
3. Add the script.
4. Disable or delete the script.
5. Verify the user runtime viewer.
6. Verify the user log.
7. Verify how many users running currently.
8. Add or change the LG for down status users.
9. Identifying which transactions are passing and failing.
10. Verify who the users are ramping up, throughput, hits per second, connections, response time and resource utilization.

Q: How to verify the errors for which request, which user, which LG, which line number, which script, how many time occurred, which user id?

A:

Check list:

- ➔ Click on errors.
- ➔ Verify each and every error for which script it was occurred by choosing script column.
- ➔ Verify which LG it was occurred by choosing generators column.
- ➔ Verify which user id causing the issue by choosing vuser id column.

Runtime settings:

Extension of RTS is “.cfg” & “.usp”.

Note: RTS will be transferred to controller from vugen script but not vice versa.

1. Run Logic: Indicates number of Iterations.

Note1: Test duration setting will override the Run Logic.

Note2: To make our script iterates for specific number of iterations, we have to choose “run until complete”.

2. Pacing: Time delay between the Iterations

We have 4 types of Pacing.

1. No pacing/ no delay.
2. Fixed Pacing, waiting time.

3. Random Pacing (generate random value and wait for the same).

4. Interval Pacing: Instructs users to finish with in a time, which includes pacing time, iteration time.

Note1: Pacing is the time delay to start new iteration after finishing previous iteration.

Note2: Pacing will allow you to control the number of iteration and number of transactions.

Note3: Pacing calculation is very important while preparing work load model.

3. Log: Logs will help you to debug the script.

- Enable logging: You will receive log messages based on the settings.
- Disable log: You are going to use this option while running the test to avoid unnecessary

4. Think time:

Think Time is the time to choose new action after getting previous response.

(or)

Time delay b/w the User actions.

Note: Even you can pass float numbers as a think time.

Q: Why think time is required?

A:

In realistic environment end users are taking sometime to choose the new action after receiving previous response.

But in my script virtual users are not waiting for to choose the new action and they are firing back to back request.

To simulate the realistic environment we have to instruct the users to pause in between the request with the help of think time.

Q: Who is going to provide the think time?**A:**

As a performance tester we have to calculate the think time. How long a normal user is waiting to choose the new action on every page.

Note: we should not use recorded think time.

Think time option:

1. **Ignore Think time:** It is going the function and execute back to back request.
2. **As Recorded:** It will pause the script execution as per the function time.
3. **Multiply think time:** Multiplies the think time.
4. **Random Think time:** it will generate the random number based on minimum and maximum percentage and passes the same.
5. **Limit Think time:** It limits the think time.

Q: Where do you place the think time?

A: We should not place the think time in between start and end transaction.

Q: What is the impact of think time on the response time?**A:****Scenario1:**

If you reduce the think time you will receive higher response time. Because of less think time users will perform more iterations and transactions, which will impact the server performance.

- Lower think time will give less breathing time, which will impact the transaction response time.

Scenario2:

- Higher think time will give good response times.
- Due to the higher think time server will get more breathing time which will process the request very fast.

Global think time:

Define the variable in **global.h**

Ex: **int X=10;**
 lr_think_time(x); (Action)

Note: If we forgot to take out the think time from start to end transaction you can filter those response times in lra file with the help of properties.

5. ADDITIONAL ATTRIBUTES:

To declare the Environment variables.

Q: How to create environment variables in LR?

(or)

How to pass a new value in to the script without opening the script?

A:

```
char * server;  
server = lr_get_attrib_string("host");  
lr_save_string(server,"url");
```

- Write the above statement in the script.
- Set the value in the RTS and pass vary arguments in runtime.

6. Miscellaneous:**Continue on Error:**

Continue script execution even when an Error Occurs.

Generate snap shot on Error:

It will generate the snap shot for every Error you can verify them in Result file (or) Controller vuser log (by clicking camera symbol).

Q: Where we can find a screen shot?

A: Go to the vuser log and click on the camera symbol.

Navigate to the LG results path and you can find HTML page for every error.

Multithreading:**Run vuser as a process:**

For all client server apps (SAP GUI or Desktop based apps. EX: SAP GUI/ Calculator) you have to run vuser as a Process.

If we are running vuser as a process every vuser required one MDRV (Multi Driver Program) Engine.

Every MDRV engine required 5Mb memory in the LG Machine.

Running Vuser as a Thread:

For all web based applications you have to run vuser as a Thread. If you are running vuser as a thread multiple users will share one MDRV engine.

NOTE: Approximately 50 vusers use one MDRV engine.

Automatic Transaction:

Allow you to generate the automatic transactions to measure the response time.

7. Network:

Speed simulation:

It specifies to use maximum/ predefined/ custom band width for your test.

Usually we are using maximum bandwidth option until or unless there is a requirement, we are not going to use custom or advanced bandwidth.

If you like to test your application with a specific network bandwidth, then use custom or advanced bandwidth.

8. Browser Emulation:

It will allow you to use multiple browsers for test.

Note:

Prerequisites:

We should install the browser in the load generator.

1. Simulate browser cache: Enabling this option instructs the users to use or simulate cache files from browser.

If disable this option, cache files will be deleted or will not be simulated.

2. Simulate a new user on each iteration: If the users are iterating for multiple times, we have to make him behave like a new user on every iteration by enabling this option.

9. Internet Protocol:

Content check:

It is a global text verification option. Verifies the text on every page.

Note: web_reg_find is a local verification point.

Steps:

1. Create application by clicking new application.
2. Create a rule under the application.
3. Provide the text and match case.
4. You can export or import the rule and extension is “.xml”.

10. Proxy:

It will allow you to configure proxy settings for all requests. It will redirect all the requests to proxy server.

Options are

- 1) No proxy
- 2) Obtain proxy settings from browser
- 3) Custom proxy

11. Preferences:

1) Enable image and text check:

This option has to be enabled for web_find and web_image_check.

2) Wininet replay instead of socket:

For NTLM based applications or SSL based applications, you can use wininet replay.

If you don't want to use web_set_socket option, then use wininet replay.

Options:

- **HTTP request connection timeout:** A unit time within which request connection operation should finish. Default is 120 seconds.
- **HTTP request receive timeout:** A time unit within which receive operation should finish. Default is 120 seconds.
- **Step Download timeout:** A time unit within which entire step has to be finished. Default is 120 seconds.

12. Download filters:

To exclude or include specific URL, use download filter option.

Q: How to design shared RTS? (or) How to configure RTS for multiple scripts?

A: Two ways we can configure the shared RTS

1) **Vugen level:** Configure RTS in one script, copy “.cfg” and “.usp” files and paste into remaining script folders.

2) **Controller level:** Select all scripts and choose shared RTS and configure the same.

Note: Whenever script having vary number of actions and names, you should not create shared RTS or should not copy “.cfg” and “.usp” files to some other scripts.

SLA configuration:

SLA configuration will allow you to compare derived statistics with expected statistics.

Step 1: Click on new under SLA

Step 2: Choose SLA measurement like total, average, throughput, hits per second, response time and errors.

Step 3: Provide threshold point and click on finish.

Rendezvous test:

Rendezvous point is the point to instruct the users to wait at a certain location once the specified number of users arrived at that point. It will execute the subsequent request.

Syntax:

```
lr_rendezvous("xyz");
```

Step 1: Write the function in the script.

Step 2: Go to controller and select rendezvous under scenario.

Step 3: Configure the policy by providing number of users and timeout.

Pacing Calculations:
(or)
Little's Law:
(or)
How to calculate TPH (Transactions per hour)?

Q : Target 1800 transactions per 1hr

1 Script contain 30 transactions

1 Iteration is taking 30 seconds

Calculate pacing?

A:

Step 1:

$$\begin{aligned} \text{Calculate the total no.of iterations} &= \text{target transactions/script transactions} \\ &= 1800/30 \\ &= 60 \text{ (iterations)} \end{aligned}$$

Step 2:

$$\begin{aligned} \text{Time for target iteration} &= \text{Target iteration * 1 iteration time} \\ &= 60*30 \\ &= 1800 \text{ sec} \end{aligned}$$

Step 3:

$$\begin{aligned} \text{Remaining Time} &= \text{Target time- Target iteration time} \\ &= 3600-1800 \\ &= 1800 \text{sec} \end{aligned}$$

Step 4:

$$\begin{aligned} \text{Pacing} &= \text{Remaining time/Target iterations} \\ &= 1800/60 \\ &= 30 \text{ sec} \end{aligned}$$

Scenario 2:
Reverse Engineering:

Step 1:

$$\begin{aligned}\text{Iteration time} &= \text{Script time} + \text{Pacing} \\ &= 30+30 \\ &= 60 \text{ seconds/one iteration}\end{aligned}$$

Step 2:

$$\begin{aligned}\text{Total Number of transactions} &= \text{Number of iteration} * \text{one iteration time} \\ &= 60 * 60 \\ &= 3600\end{aligned}$$

Q: Target 3000 transactions per 1hr

1 Script contain 60 transactions

1 Iteration is taking 10 seconds

No.of users 10 and duration 1 hr

Calculate pacing?

A:

Step 1: Calculate the total no.of iterations = target transactions/script transactions
 $= 3000/60$
 $= 50 \text{ (iterations)}$

Step 2: 1 User Iteration = Total iterations/no.of Users
 $= 50/10=5$

Step 3: Time for target iteration = (Target iteration-1) * 1 iteration time
 $= (5-1)*20$
 $= 4*20$
 $= 80 \text{ sec}$

Step 4: Remaining Time = Target time- Target iteration time
 $= 3600-80$
 $= 3520 \text{ sec}$

Step 5: Pacing = Reaming time/Target iterations
 $= 3520/5$
 $= 704 \text{ sec}$

Note: Whenever you are dealing minimal number of users and minimal number of transactions, we have to reduce one iteration from target iteration.

Formulae:

Pacing calculation $P = R/I$

P → Pacing

R → Remaining Time

I → Number of iterations

Remaining time $R = D - (T * I)$

D → Duration

T → Target transactions

I → Number of iterations

Number of iterations $I = T/S$

I → Number of iterations

T → Target transactions

S → Script transactions

Formulae 2:

Number of users = $TP * (R + T)$

TP → Throughput per second

R → Response time

T → Think time

Pacing $P = R + T$

R → Response time

T → Think time

Q: How to add multiple load generators for single script?**Solution:**

- Select the group
- Click on vusers
- Add new LGs against vusers

Q: How to add multiple scripts against one group?**Solution:**

- Select the group
- Click on vusers
- Add new script for existing users or new users

Q: How to schedule a test to start at particular time?

Solution: Using start time option (under scenario schedule), we can schedule a test to run at particular time.

Note: Click start button after scheduling.

Q: How to create multiple schedules in one .lrs file?

Solution: Using new schedule option, we can create multiple schedules for various kind of test.

Q: On what basis you are deciding how many LGs required for test?

Solution: Based on below parameters, you can decide how many LGs are required

1. Number of users
2. LG configuration
3. Script variables and size
4. Protocol type
5. Running vuser as a thread or process

Note: You have to conduct a memory footprint test and decide how many users can support by LG.

Memory footprint test: Testing with single user or 20 users and 50 users conducting how many users it can sustain called memory footprint test.

Current version of LR will send the notification whenever it reaches the threshold point.

Memory Foot Print in Load Generators:

The no.of Load Generators depend on the below items.

1. Ram size of the Load Generator.
2. No.of Variables & Memory allocation for variables in LR Script
3. How you are running vuser as a Process or a Thread.
 - If you are running vuser as a process
1 vuser → process → 1 MDRV Engine → 5mb
 - If you are running vuser as a Thread
1 vuser → (50) Thread → 1 MDRV Engine → 2mb

Q: Is it possible to extend the test duration while running the test?

Solution: It is possible to extend the test duration using pause and resume schedule option.

Q: How to copy scheduler settings (ramp up, duration and ramp down) from one script to another script?

Solution: Using copy scheduler options, we can copy the scheduler settings from one script to multiple scripts.

Q: What is the process name (in task manager) for controller?

A: wlrun.exe

Q: When you are going to run agent as a process or service?

A: For all web based applications, run agent as a service.

For all client server applications or which applications are communicating with hardware of the machine (NTLM or windows based applications), run agent as a process.

Q: How to run agent through command prompt?

A:

magentproc.exe → for process

magentservice.exe → for service

Q: How to invoke these manually?

A:

→ Start

→ All programs

→ HP Software

→ HP Load Runner

→ Advanced settings

→ Agent process/service

Note: If you like to make it run as a service by default while installing LG software, you have to choose the same option.

Q: My script is working fine in vugen but not in controller?

A:

Verify whether you are able to access the application from the LG

Verify LG runtime quota.

May be load balancing is not happening properly.

Q: Who will provide the rampup, rampdown settings?

A: No one will suggest to use specific rampup and rampdown. As a performance tester, we have to derive the rampup time based on below parameters

- 1) Application type (web/client server)
- 2) Type of load balancer

Note: Usually we will try to finish the rampup within 30-45 minutes if you are running test with 300-500 users.

Note: If it is a client side application, the rampup might be high.

Scenario 1:

In one of my application, we design the test by Ramping up user for one every 5 seconds. But failed to execute this test due to the application crash?

Solution:

Application is working with minimal users but failing in full blown test. So we requested help from architect people or developers and return the same test. They found problem is with load balancer due to the rampup rate.

Whenever we are ramping up the users fastly, LB pushing all users into one instance which is causing the failure. So they suggested to increase the rampup time.

Common controller output messages:

No match found for the requested parameter

Issue: Connection issues

Solution: Fix the same

Connection to load generator fail or process lr_bridge.exe was not created on host machine (or)

Communication error fail to connect to remote host or two way communication fail

Solution 1: Verify both machines whether those machines are on same network or not.

Solution 2: Verify both machines by pinging.

Solution 3: If there is a proxy server in between both of the machines, speak with IT admin people to open the port number.

Solution 4: If there is a firewall in between both of the machines, speak with IT admin people to install listeners in LGs under MI Listeners.

Files and users are not transferring to the LG

Causes:

- Script path is too long, reduce the path.
- Parameter naming convention might be a problem. Parameter naming convention should not have any special characters.
- Script might be corrupted, fix the issue.
- Script naming convention should not contain any special characters.
- Parameter naming convention should match with “.dat” file naming convention in “.prm” file

Solution:

Open “.prm” file, change the naming convention.

web_reg_find failed to find the text

Cause: Text verification failed.

Solution: Open the script, fix the issue.

Step download timed out error

Cause: Entire request is not finishing within 120 seconds or application is too slow

Solution: Increase the step download time limit (1000 or 2000)

User abnormally terminated by MDRV (Multi Driver) engine

Cause: User running as a thread instead of process

Solution: Change thread to process.

Authentication failed 403 forbidden error

Solution: Use web_set_user to submit the credentials to the web server.

Time out exceeded while waiting to receive data from the URL

Cause: Network issue or application issue.

Solution: If network or application is very slow, increase the timeout limit

Could not resolve address of host

Cause: Failed to connect to application due to invalid URL or proxy server issue or SSL issue.

Solution: Resolve invalid URL or proxy server issue or SSL issues.

Customized error message

Cause: Based on the requirement, you might have written lr_error_message to understand where it is failing.

Solution: Open the application, fix the same.

HTTP 500 error

Cause: Server busy or application busy or down

Solution: Open the application, fix the same.

Compilation error

Cause: Syntax issue, code issue... in the script

Solution: Open the script and fix the same.

C interpreter error

Cause: Runtime error which is caused by C code.

Solution: Fix the C code issue.

Memory violation exception error

Cause 1: Generating some random values from an empty array.

Cause 2: Variable memory allocation issue.

Solution: Fix the same

Performance Center:

Performance center is a web based application, which is a web interface of controller. Using PC we can design, execute, and download the results from anywhere any time. Using PC you can manage your resources (controller, LCD and no. of users) in perfect manner by hooking time slots.

Note: You have to buy PC license as well as Controller License.

Versions: 9.1/9.5 and 11.0 (Integrated with ALM)

Activities in Performance center:

- 1) User level access
- 2) Project level access
- 3) Time slot bookings
- 4) Test design and Test execution
- 5) Upload & Download the scripts.
- 6) Download the results anywhere Even you can monitor the servers by integrating site scope or other tool.

Advantages:

1. It is a web based application.
2. You can monitor the resources in b/w your team members.

Process to connect to the PC:

- Launch the PC URL
- Enter credentials
- Choose your project
- Push the scripts into controller.

Options or Tabs:

1. Status: Shows how many test executions are currently running, upcoming schedule test & how many recently ran (executed).

2. Host: Host shows how many LGs, controllers, firewalls connected to the PC.

3. Timeslots: Here you can book the timeslots for your test by specifying duration tests, number of users and number of LGs.

Note: You have to book extra 15 min to import the Results.

Vusers scripts: Here you can upload, download, copy the scripts from other project, and also duplicate, view & delete the scripts.

There are two ways to upload the scripts:

1. Zip the script folder & upload to PC

2. Manual Procedure

- Vugen
- Tools
- HP ALM (11.0)
- Connection (or) PC Connection (based on version)
- Provide the PC URL
- Click connect button
- Credentials of PC.

Monitoring Profiles: We can monitor the machines health by adding profiles.

Auto start viewer: It indicates the test schedule to start automatically as per the time frame.

Change project: To change the project this option.

Load Test:

Manage: New Load test → Design → RUN

General Tab: Load test name, TCP/IP spoofing options.

Scheduler: Same as controller design scenarios.

Design groups: Load generators.

We can download “.lrr” files after collating the results.

Performance Center	Controller
1. Web based application	1. Standalone application
2. There is a time slot mechanism	2. No time slot mechanism
3. We can download/upload results anywhere and anytime.	3. We can't upload/download results until you get login.

ALM (Application Life Cycle Management)

Lunch the ALM application using URL upload the scripts in to ALM by zipping or save as option on my computer.

Navigate the test plan under testing,

Move the script test plan by creating a folder.

Click on edit test, to provide by number/ percentile, manual/ or goal, by group /or by schedule

Push script into scenario, provide RTS, ramp up, ramp down, duration, LG and load distribution.

Click on submit button.

Go to time slot, book a specific time or block a specific time by L.G by providing no. of users date, start time and end time.

Note: you can manually specify the L.G's or you can choose automatic option.

You can start the test automatically or manually.

Go to test results, choose your run id, download RAW- results or zip file.

Menu:

Dashboards

Managements

Requirements

Cloud settings

Testing

Resources

Defects

Performance center

Defect: When there is a deviation between the actual and expected, we have to raise a ticket under defect tab by providing severity, priority, expect, actual results, attachments.

Testing host: It will allow you to verify how many LG'S and controllers are configure whether those are operational are not operational and in which location or hosted.

Types of testing

1) Warm up test/Dry-run test/Discovery test:

This is not the actual test but making ensure that all scripts, test data, environment and application running fine and stable.

Note: Warm up test will conduct with 10% or 100% of actual load for short duration.

2) Performance test/Baseline test:

Whenever you don't have SLAs, you have to conduct baseline test with single user, single script, single iteration and execute the script in standalone mode and get the response times, consider as a baseline response time.

Step 1: Choose schedule by group option

Step 2: Choose start scenario begin option to make scripts run one by one.

Note: Application should behave properly under the load how it was behaved with a single user.

3) Load Test:

Load testing is the testing to verify the application behavior under load.

We are designing the scenario with 100% load.

Refer graph below

4) Endurance Test/Soak Test/Longitivity Test:

Verifying whether application is available for longer duration or not.

We have to design endurance test with normal load (50-60% of peak load) for longer duration (12h, 18h, 24h)

Note 1: Objective of this test is to identify the memory leakages.

Note 2: In some cases, we might required to design scenario with normal load by increasing think time and pacing time.

Refer graph below

5) Stress Test:

Stress test is the test to identify the breaking point or performance degradation point of application.

We can stress the application in two ways:

- 1) By increasing the number of users
- 2) By reducing the pacing and think time

Note: We can increase the number of transactions by reducing the pacing and think time without adding the users.

Q: I have license for 500 users but client asked me to test the scenario for 700 users.

Solution: To achieve the above objective, we reduce the pacing and think time to generate more number of transactions. Usually we are designing the scenario with 1.25X, 1.5X, 1.75X, 2X, 3X, 4X, 5X loads.

In three ways, you can add the vusers

- 1) Design the scenario and add the users manually for every time interval.
- 2) Design the scenario with more number of users using pause and resume scheduler, you can add the users automatically.
- 3) Using real world scenario, you can add multiple actions for multiple rampups and durations.

6) Failover Test (based on client request):

In the absence of first data cord, verifying whether second data cord can take the load without any failure transactions or performance degradation.

Process:

In one of my test, we designed a failover test for 2 hours duration. After one hour duration, we reported to IT admin and architect people to unplug the first data cord from the network. We verified whether test reported any failure transactions or performance degradations. Monitored second data cord whether it is able to take entire load in the absence of first data cord.

In the above scenario, my role is very minimal. Plugging and unplugging performed by network people.

7) Benchmark Test:

Benchmark test will give a repeatable set of quantifiable results which from current and future releases. These results has to be compared with baseline test results.

8) Capacity Planning Test:

By forecasting the future usage, whatever the sequence of test we are conducting is called capacity planning test.

Process:

We have to conduct load test, stress test to identify application breaking point. Speak with BA people to understand what is the growth of the business and plan for capacity planning.

Refer graph below

9) Spike Test:

We have to test the application behavior under abnormal conditions.

Refer graph below

10) Volume Test:

Volume test is the test to verify the application behavior under huge amount of load.

Example: Interfaces and Batches

11) Scalability Test:

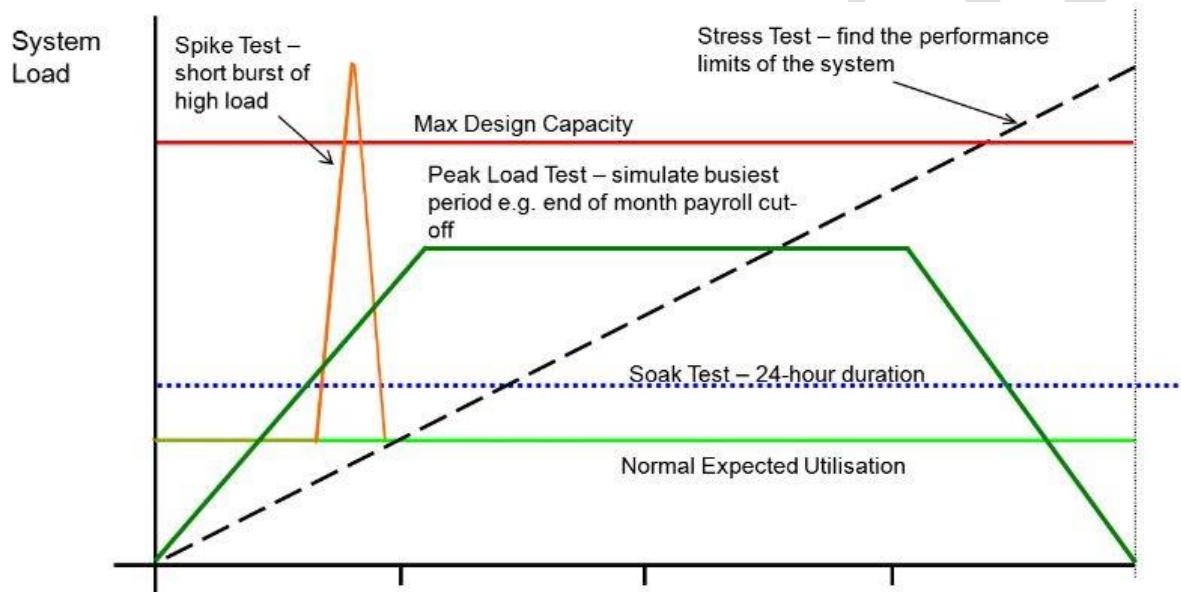
Scalability Test is the testing of an application to measure its capacity. Scale up or scale down using horizontal and vertical techniques.

We can scale down application in two ways:

- 1) Vertical scale down: Adding the resources to same node (CPU, memory, cores)
- 2) Horizontal scaling: We can add multiple nodes to existing system.

12) Network Latency/Wan Emulation Test:

To simulate the network latency (delay) we used HP Shunra (Network Virtualization) and conducting virtualization test.



Q: If the controller crashes before collating the result, what is your approach to recover the file?

Solution: If the controller crashes before collating the result, we can recover the result file.

Process:

- ➔ Go to load generator
- ➔ Select details
- ➔ Copy the path of the log
- ➔ Connect to LG through RDP
- ➔ Navigate to same path
- ➔ Copy “.eve” and “.log/.map” files
- ➔ Paste them into controller result path

ANALYZER

Analyzer file Extension is “.lra”

- 1. Cross results option:** Allow you to compare two “.lrr” files as part of benchmarking test.
- 2. Section explorer:** Contains “.lrr” path, period, duration, average throughput hits per second, total throughput, hits per second, transaction response time and status code.
- 3. Graphs:** Allow you to add and delete the graphs.
- 4. Properties:** Allow you to exclude/ include think time and generate percentage response time.
- 5. Controller output message:** Controller error message will be displayed which will be helpful to analysis.
- 6. User data:** Allow you to write something.
- 7. Raw data:** Based on the request we can pull the raw data and send to architecture people to analysis purpose.
- 8. Graph data:** Will give raw data for graph.
- 9. Legend:** To make you understand which color is indicating which measurement.
Scale: Indicating number of measurements in graph.
- 10. Granularity:** Time difference b/w two saturation points.
NOTE: minimum Granularity for throughput and Hits per 5 second.
For all remaining graphs 1 second
- 11. 90th percentile:**
90 percent of the transactions are completing with in this limit.
Step1: Write all the response times in ascending order.
Step2: Take out 10% of values from below.
Step3: Which will be the highest value consider as 90% response time.

Note1: We have to report only 90% response time to client.

Note 2: Based on client requirement we can generate 80%, 85%, 90% ...etc.

12. Reports:

By default we can generate doc report, Html report, crystal report, PDF report.

Reporting:

Once the test got completed, I will export the response times to excel and I will prepare a comparison report.

Comparison report:

It compares 90th percentile response times with baseline response times of previous test results and I will maintain a RAG (Red Amber Green) status.

In some other tab, I will copy merged graph to understand the test results.

Apart from comparison report, we will prepare a quick analysis summary which contains objective, scope, how we designed the scenario, test environment, observations in terms of resource utilization, high response times and controller, web server logs.

I will send a mail to get AWR, NMON reports for future analysis.

I will prepare a PPT by analyzing all the supporting files (AWR, NMON ...) by mentioning objective, observations, environment comparison, high response transactions, root causes to present to the stake holder.

Q: How to generate the reports?

A:

- Choose reports
- Navigate to report template
- Click on generate report
- Save on required format.

Q: How to create a template?

A:

- Tools
- Navigate to templates
- Create new template
- Click on save and close button.

Q: How to add existing template to current session?

A:

- Open “.lra” file
- Go to tools
- Choose templates
- Select specific template
- Apply to session.

Analysis Process:

Q: What is your approach to analyze the statistics?

(or)

What is the process you are following to identify the bottleneck?

A:

Once the test got finished, I am going to compare derived statistics with expected statistics. If both are not comparable, then I will start the process to find out the root cause.

- Client side statistics analysis (analyzer, throughput, Hits per second, Response time)
- Server side statistics analysis. (Hardware and OS level statistics).
- Application side statistics (Methods, I/O operations, DB, EJB, Packages etc..)
- Configuration setting analysis (Current limit, Connection limit, Thread limits..etc..)

1. Client side statistics analysis:

- Running Vusers.
- Connections.
- Hits per seconds.
- Throughput.
- Error per second.
- Response time.

Merging: We can merge the graph in 3 ways.

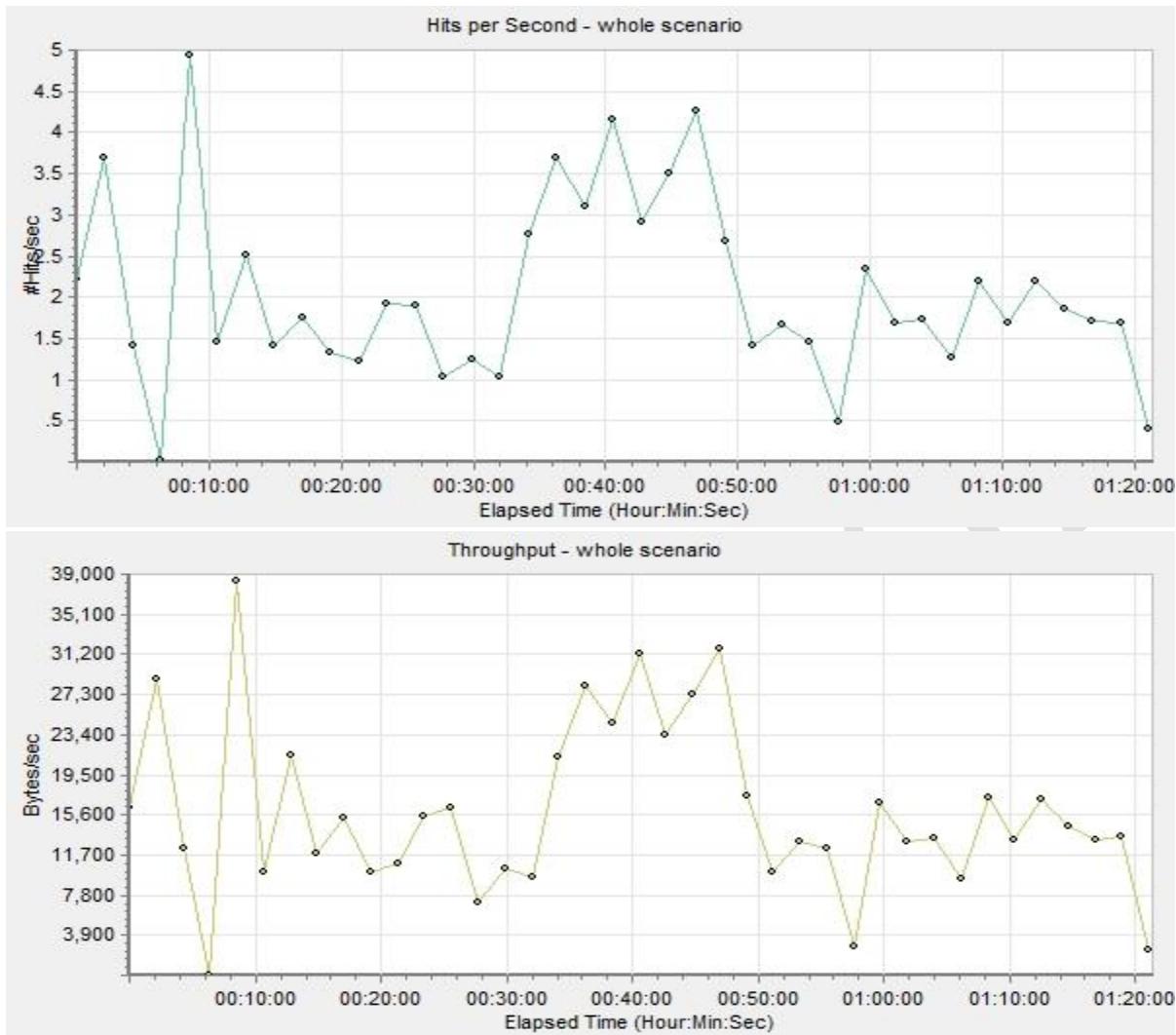
Overlay Graph

Tile graph

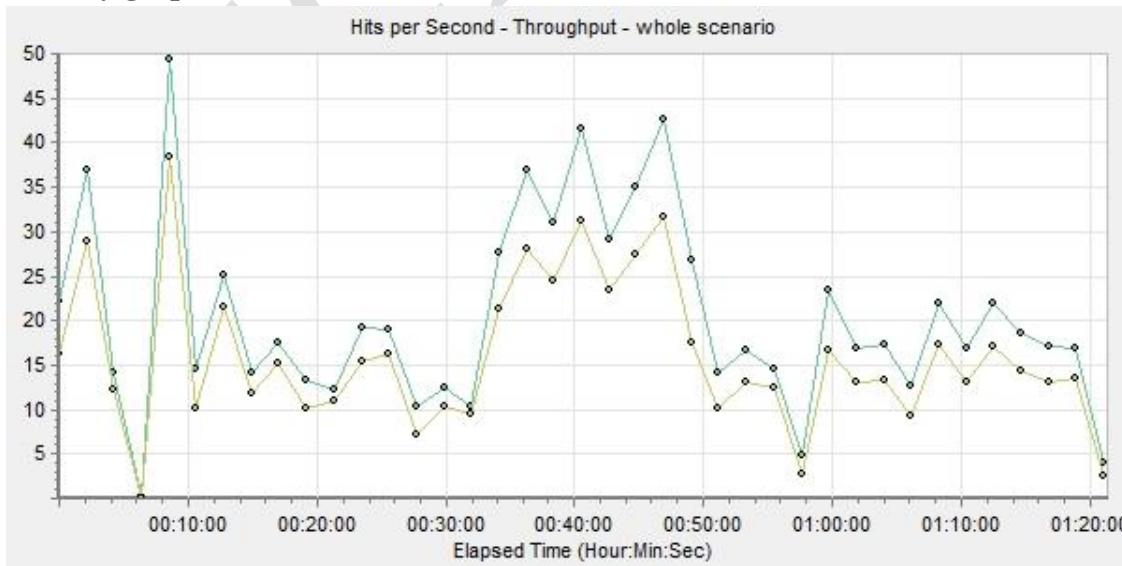
Correlate graph

1. Overlay graph:

It plots a graph with two Y-axis by sharing one X-axis.



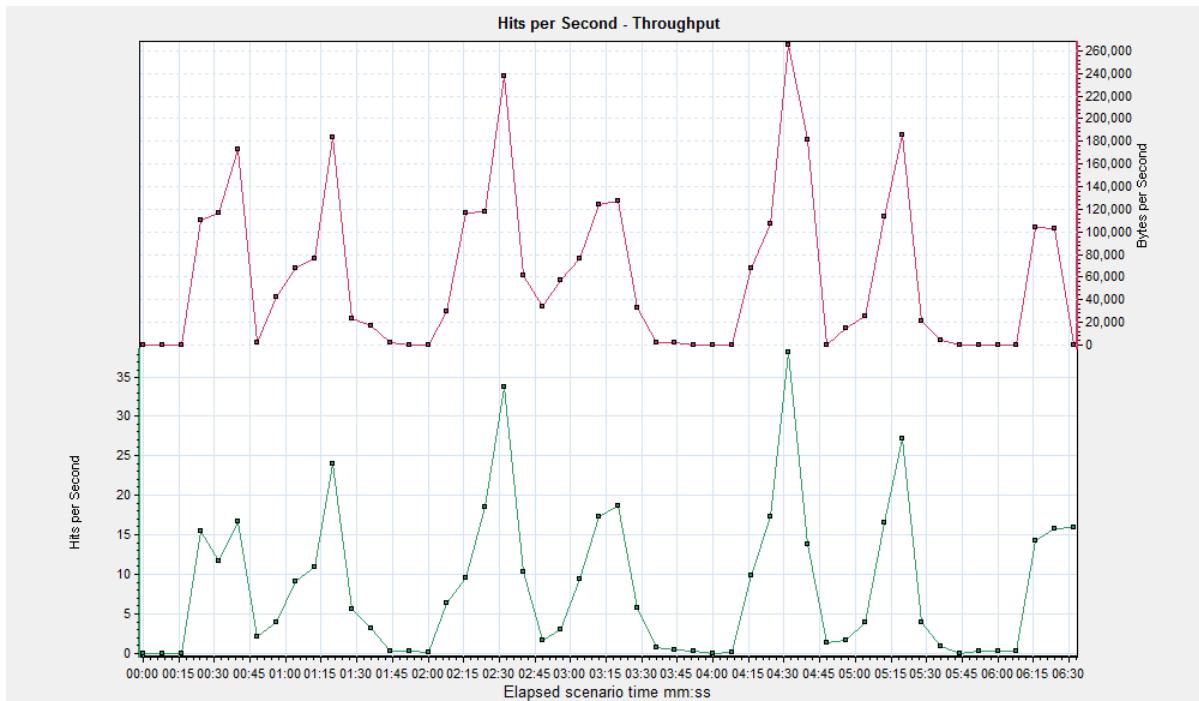
Overlay graph:



2. Tile Graph:

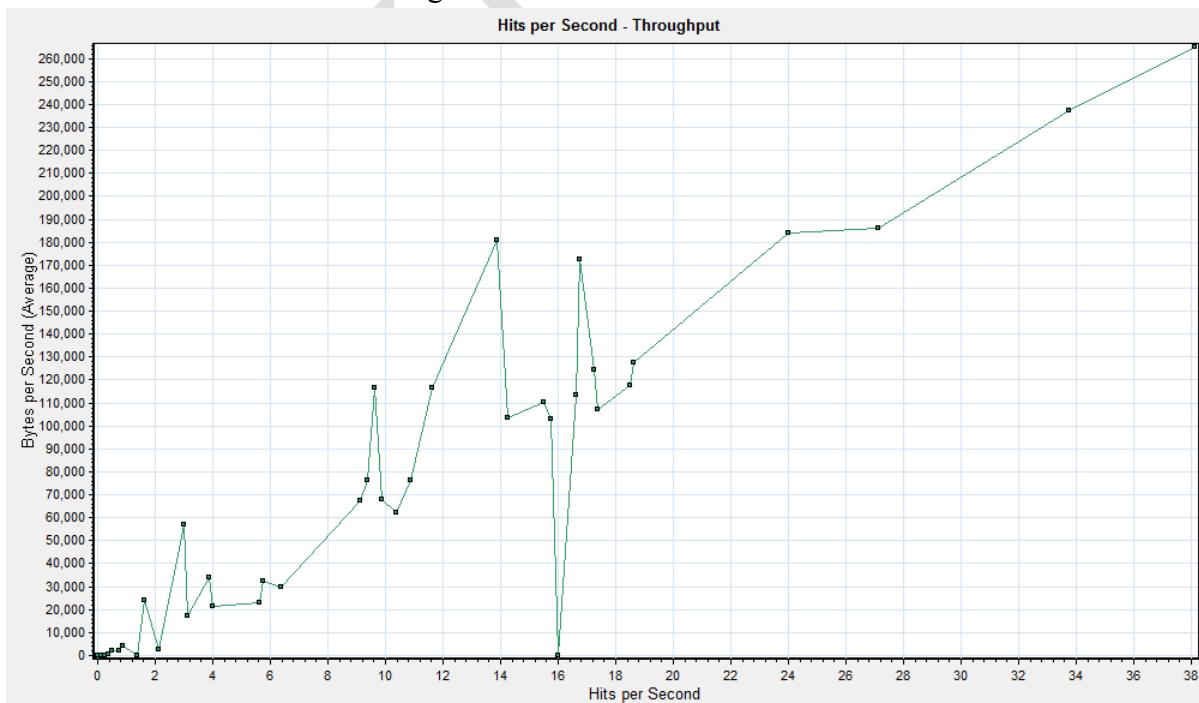
View the contents of two charts that shares a common X-axis one-above the others.

Ex: Hits per second and throughput



3. Collate Graph:

Plot two Y-axis of two charts against each other's.



Scenario1:**Relation between Hits per second and throughput.**

- Both should be directly proportional, if not

Cause1: That could be a network bandwidth issue.

Cause2: Web server might have issue.

Cause3: Application itself having the issue.

Note: Hits are increasing and throughputs are not increasing due to the application issues. We are receiving exception page which impacts high hits low throughput.

Scenario2:**Relation between running users and hits per second?**

- Both should be directly propositional, if not application itself having a problem (or) application is not responding well.

Scenario3:**Relation between throughput and response time.**

- Both should be inversely proportional.

Note: As per the market standard both should be inversely proportional based on boundaries (if you are testing pages).

Scenario4:**Relation between running users and connections**

- Both should be directly proportional, if not

Cause1: Connection limit issues in the web server.

Cause2: Number of treads limit reached threshold point in the web server.

Controller output message for above issue:

1. Users permanently or prematurely shutdown
2. Web server log max client error.

WEB PAGE DIAGNOSTICS:

Using Web page diagnostics graph we drill down the issues related to component, network, and server level issues (break down drill).

Component Break down graph:

Which will allow you to analyze component level issue any one of the component getting delay to download which will be reported as an issue.

Time Taken For First Buffer Graph (TTFB):

If the TTFB is high then the problem is with server or application.

If the TTFB is low and the page response time is very high then that is a network issue.

Q: What is the difference between absolute graph and relative graph?

A: Absolute graph plots the graph based on the system time. Relative graph plots the graph based on elapsed time.

Server side Statistics analysis:

Monitoring CPU, Memory, and Disk Utilizations is called server side analysis.

Note: For all windows OS based machines CPU, memory utilization should not cross 80%. For all UNIX, Linux, Red hat OS based machines should not cross 90%.

Q: Difference between the simultaneous users and concurrent users?

A: Whoever connects to the application (or) whoever keeps the load against the application those users consider as concurrent users.

Whoever performs the same transaction in same time those users will be considered as simultaneous users.

Note: concurrent users always greater than (>) simultaneous user. To conduct the test with the simultaneous user we have to design rendezvous test.

Thread Dump:

Whenever application is not performing well, we used to analyze thread dump.

Thread Dump is a snapshot of thread status.

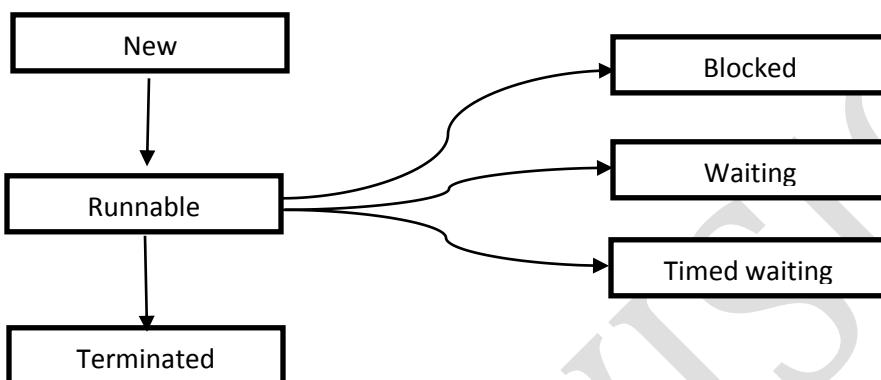
There are two types of threads:

1. Demon Threads: Which are invoked by OS and hardware level

2. Non Demon Threads: Which are created by program.

Example: public static void main (string args [])

Thread Status:



Thread contention:

Thread contention is a status in which one thread is waiting for a lock which is locked by some other thread.

Deadlock:

Deadlock is a situation where one or more threads are waiting for other resources which are locked by some other threads.

Thread synchronization:

This code will allow the threads to use multiple resources by multiple threads.

In Java, every object has one monitor. At any point of time, only one thread get lock on monitor. Other threads will wait until monitor will be released.

Note: Thread dumps will be published with the help of jstack or jvisualvm.

As part of analysis, we are going to identify which threads are in blocked status and waiting status.

If you find any blocked threads, verify for which stack it is looking for, who locked on expected stack, copy these details and post to developer.

If you find waiting threads, we have to find which methods are executing for particular thread (wait (), park (), sleep ()) and copy details and post to developer.

If multiple threads are trying to get locks on stacks which is locked by some other threads causes deadlock threads. Apart from above analysis, we do have internal thread analyzer tools.

Using these tools, we are deducting the deadlocks and thread level issues. For infrastructure level, we have to take CPU sampling, which thread is utilizing more CPU and report the same.

Memory Dump:

Memory dump is a snapshot of memory utilization statistics in a particular time period.

Whenever you received OOM (Out of memory) exception or memory leakages we have to take the memory dump to find out the root cause.

We do have two types of memories

1. Stack memory: Static variables will be loaded into stack memory

2. Heap memory: Dynamic variables will be loaded into heap memory.

Memory dump contains below information:

- Objects
- Classes
- Class Names
- Class loader information
- Fields
- Primitive fields
- Garbage collection roots
- Thread level data
- Stacks

To understand the memory dump, we should drill down how many threads required for our application, how much memory required by each and every thread will give detailed idea about memory footprint (static)

Memory dump will give static memory and dynamic memory information (user session) for analysis purpose.

Using memory dump, we can identify which object, class are running for more time, we can copy them and send to the developer.

If you are not able to analyze the memory dump manually, we can use memory dump analyzer tools to identify the issue.

Memory dump extension is “**.phd**” or “**.hprof**”

Difference between Load Runner versions:

Load runner versions	Changes
1. LR 9.0 – LR 9.5	<ul style="list-style-type: none">Protocol advisorRDP and RTE protocol enhancementCorrelation Studio enhanced with new rule
2. LR 9.5 – LR 11.0	<ul style="list-style-type: none">AJAX truclient introducedResult export option in analyzerMobile protocol introduced<code>web_reg_save_paramex()</code> introduced
3. LR 11.0 – LR 11.5	<ul style="list-style-type: none">AJAX truclient protocol started working in IEEntire user interface of vugen got changedNetwork virtualization and service virtualization introduced.Solution explorer introduced.LR developed in sharp IDE
4. LR 11.5- LR 12.01	<ul style="list-style-type: none">Mobility protocol enhancedSAP mobile protocol introducedLR became lightweight application
5. LR 12.01- LR 12.5	<ul style="list-style-type: none">Customized protocol introduced where you can create your own protocolJunkin plugins are introduced

Profiling Tools:

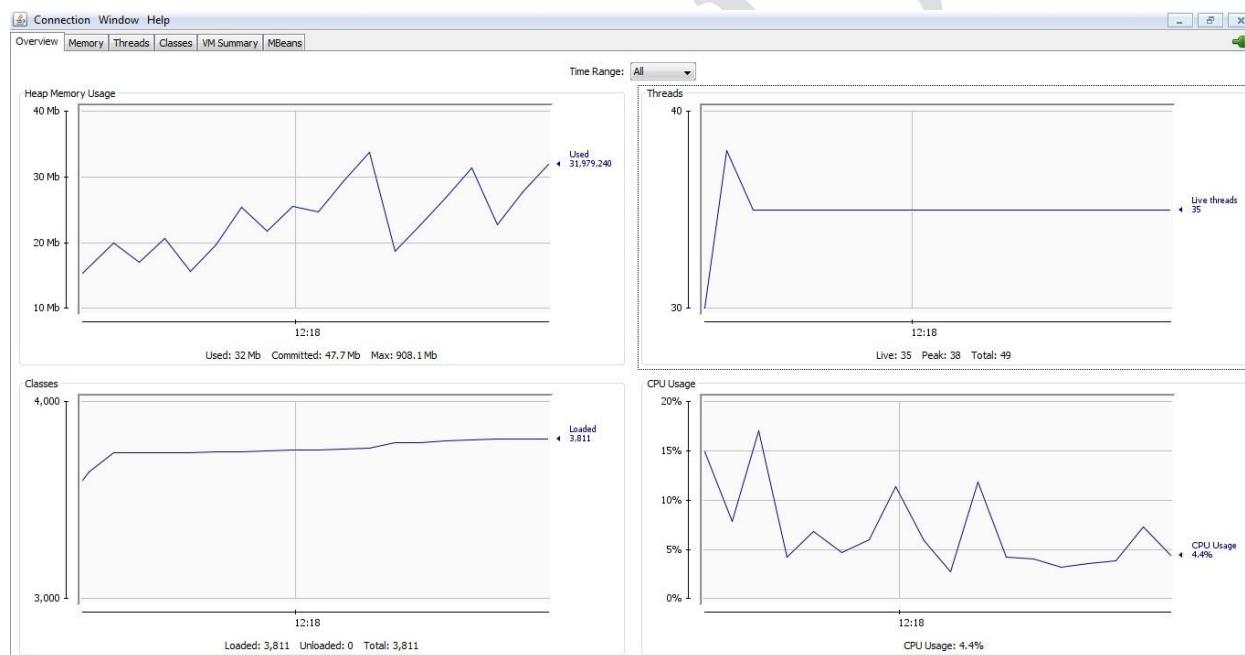
1. JConsole:

JConsole is a default profiling tool for JVM. To leverage the services we have to install JDK. Here We can only view Hardware Statistics such as CPU, memory, threads, classes By default we can view CPU, memory, threads, classes as part of garphs.

Steps to invoke JConsole:

- My Computer
- Program Files
- Java
- bin
- JConsole

Overview Of JConsole:



2. JVisualVM:

JVisualVM is a default profiling tool for JVM. To leverage the services we have to install JDK.

Steps to invoke JVisualVM:

- My Computer
- Program Files
- Java
- bin
- JVisualVM

We can Monitor

- 1.Local Machine
- 2.Remote Machine

Steps to invoke Remote Machine:

- Go to Remote tab
- Add ProcessID
- Add JMX connections by providing Port Numbers

1. Overview:

Here we can view how much memory?, XMS & XMX settings, JVM version, JRE Version

Here we can check JVM Arguments & System Properties

2. Monitor:

By default we can View CPU,Memory,Classes,threads graphs

Here we can perform GC and we take heap dump

Heap dump extension is ".hprof"

Here we can check memory leakage with the help of heap metaspace graph.

3. Threads:

Here we can view thread status like Running(), Sleep(), Park(), Waiting()

And we can take thread dump for Analysis Purpose.

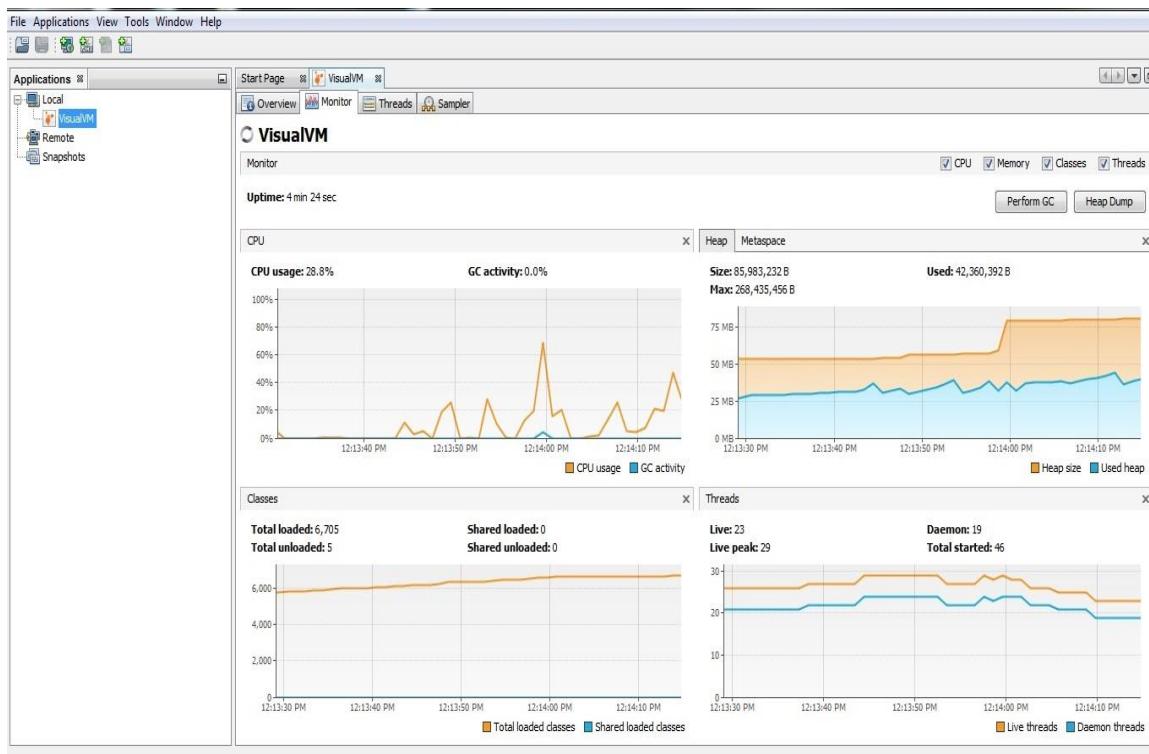
Note 1: For thread dump analysis copy the thread dump and paste it in any online thread analyzer tool.

Note 2: We can't copy Thread dump from remote machine to local machine and vice versa.

4. Sampler:

Here we can view how much CPU & Memory Utilized by each & every thread as part of JVisualVM.

Overview Of JVisualVM:



3. JMC(Java Mission Control):

JMC is a upgraded version of JVisualVM. Its has more Graphical User Interface(GUI).

By default we can view JVM CPU usage, used java heap maemory, total live thread count, used physical memory as part of GUI

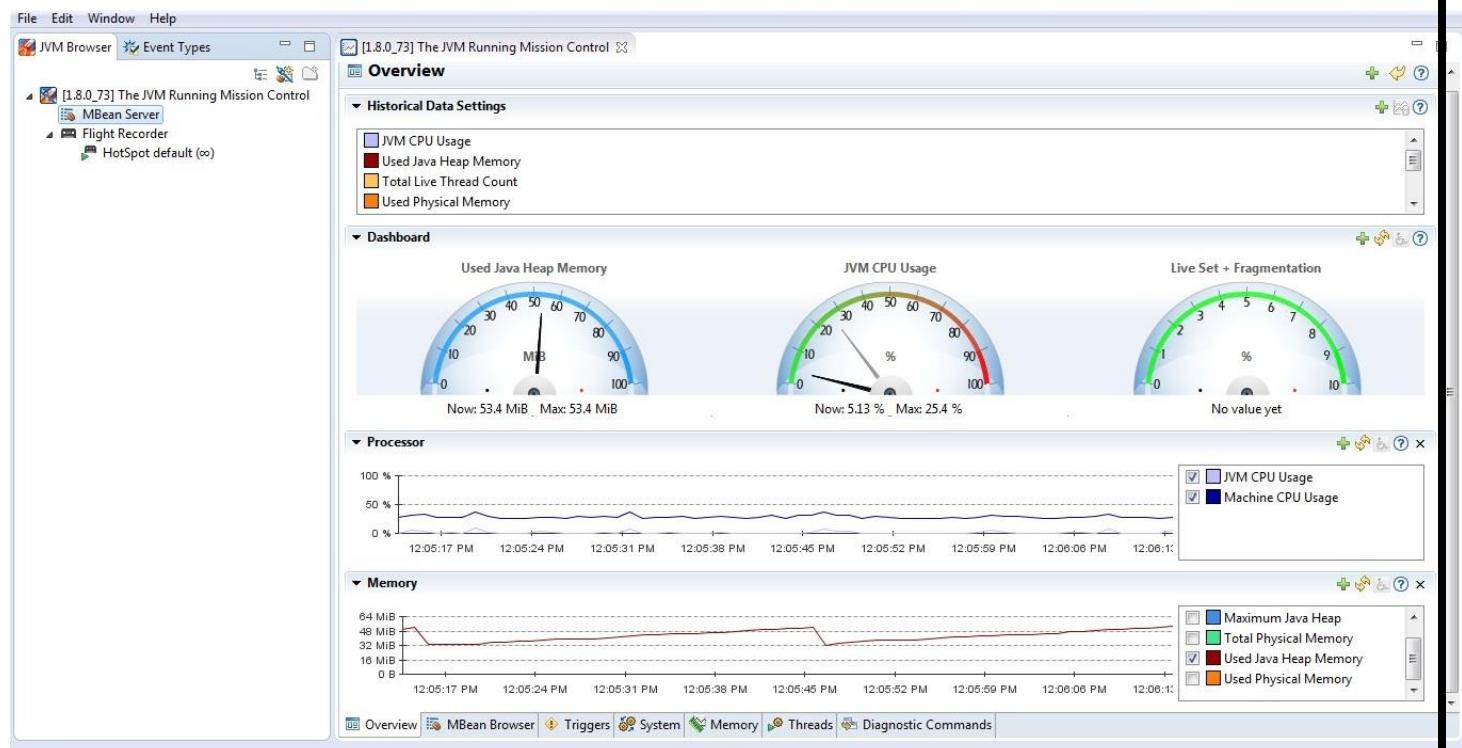
Also we can view processor & memory as part of graphs.

Here also we can analyze thread and memory status same as JVisualVM.

Steps to invoke JMC:

- My Computer
- Program Files
- Java
- bin
- JMC

Overview Of JMC:



Monitoring Tools

1. Site scope:

- It is an agent less monitoring tool or online monitoring tool.
- It is a server monitoring tool, to monitor any kind of servers.
- At a time, we can monitor 120 servers.
- We have to configure the web, app and DB to site scope.
- We can monitor for every time interval at any point of time from anywhere in the world.
- By using this, we can monitor windows, UNIX and ax-box.
- We need to install the site scope in your machine.
- We need to configure server in site scope.
- We can add monitoring profiler in site scope.
- It is a product of HP, and latest version is 11.52.

Launch the site scope URL and provide credentials after that we are able to see the following tabs

- Monitor – Here we can add counter to monitor which server you want.
- Remote Server – In this, we can configure win, UNIX servers.
- Templates
- Preference
- Server statistic
- Tools

Configuring server in site scope

1. Go to Remote server tab – select windows or Unix
2. For windows, provide the following details
 - Name
 - Description
 - Server IP or Server Name
 - Provide credentials i.e, username and password
 - Method – WMI (web method invocation)
 - Click on save
3. For UNIX provide the following details.
 - Name
 - Description
 - Server IP or server name.
 - Provide credentials i.e. username and password.
 - Provide OS.
 - Method –SSH (secure socket host)
 - Click on save

In monitor tab we can do add counter, quick reports, current status and history.

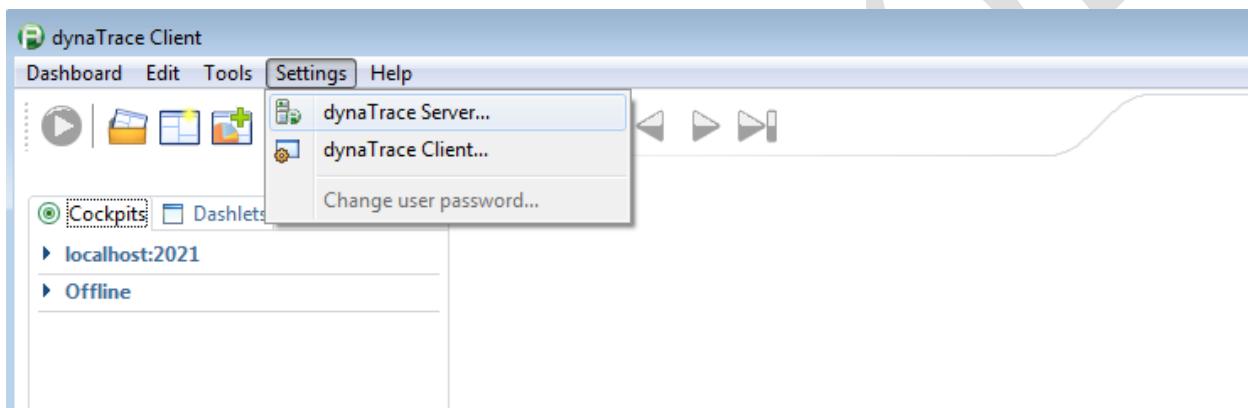
Adding servers:

- Open Site Scope
- Monitor
- Remote servers
- UNIX remote systems

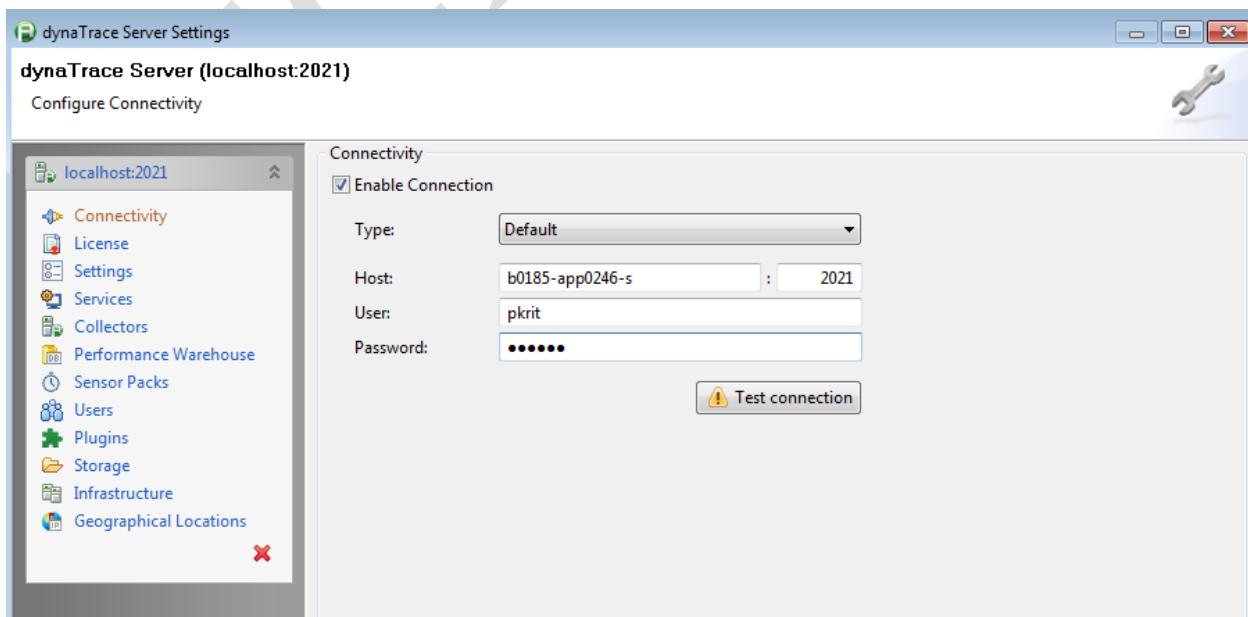
2. Dynatrace:

Step1: Login to dynaTrace server via dynaTrace client 5.5 installed on your desktop using your NTID and password.

First open the dynaTrace client 5.5 and click on “Settings” → dynaTrace Server as shown below



Step 2: You will get a new window where you are asked to enter the server details and your credentials.



Click on “Test connection”. Once the connection is established click on click on “Apply” and then “OK”

There are 2 dynaTrace servers configured for TPT environment and they are b0185-app0246-s and b0185-app0247-s, so ensure that you login to the server where your project profile has been created.

The below table contains the server and the system profile of Emerging Business applications.

S1 no	Application	Dynatrace server	System Profile
1	Interlink	Not Applicable	Not Applicable
2	Encompass	B0185-app0247-s	XH_EncDashBoard
3	Northlight	B0185-app0246-s	DR-Northlight
4	Drivewise	B0185-app0246-s	F9DRVWISE-8-TPT
5	NQ2/MyAccount/Allstate.com	B0185-app0246-s	IMT

Once you launch dynatrace, you can select the system profile on which you wish to work/monitor.

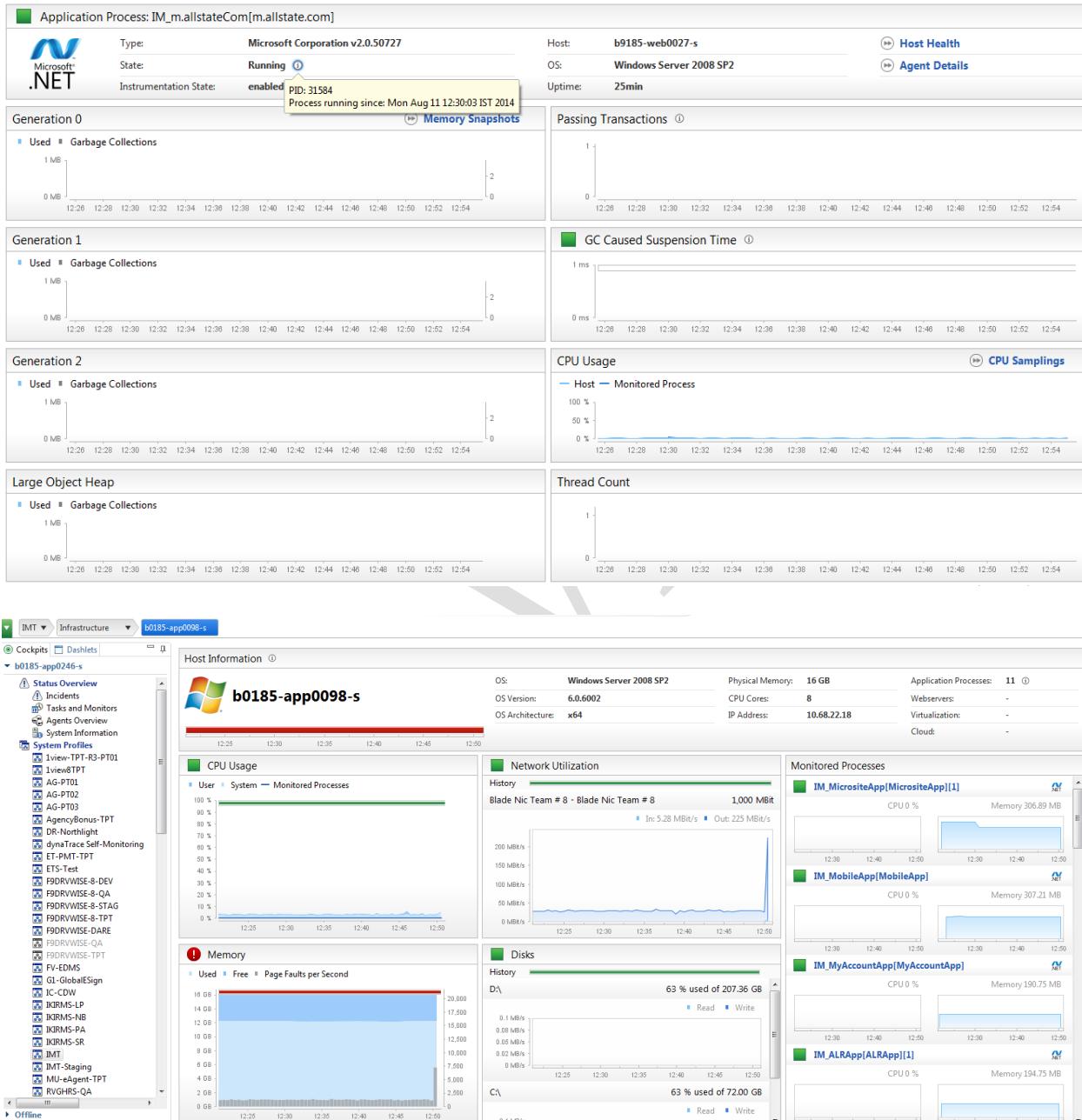
Each profile has servers configured for monitoring. IMT profile has 4 servers (2 web and 2 app servers) configured. Here we can monitor the infrastructure and the applications alternately by selecting the desired one. In infrastructure we can monitor the servers performance, the CPU & Memory utilized etc.

The screenshot shows the Dynatrace Infrastructure module. On the left, there's a sidebar with navigation links like Cockpits, Dashlets, Status Overview, and System Profiles. Under System Profiles, it lists profiles such as 1view-TPT-R3-PT01, 1view8TPT, AG-PT01, AG-PT02, AG-PT03, AgencyBonus-PTP, DR-Northlight, and dynaTrace Self-Monitoring. The main area displays four host cards. Each card has a green header bar with the host name and a yellow Windows logo. Below the logo, it shows the OS (Windows Server 2008 SP2) and OS Version (6.0.6002). At the bottom of each card is a blue 'Host Health' link.

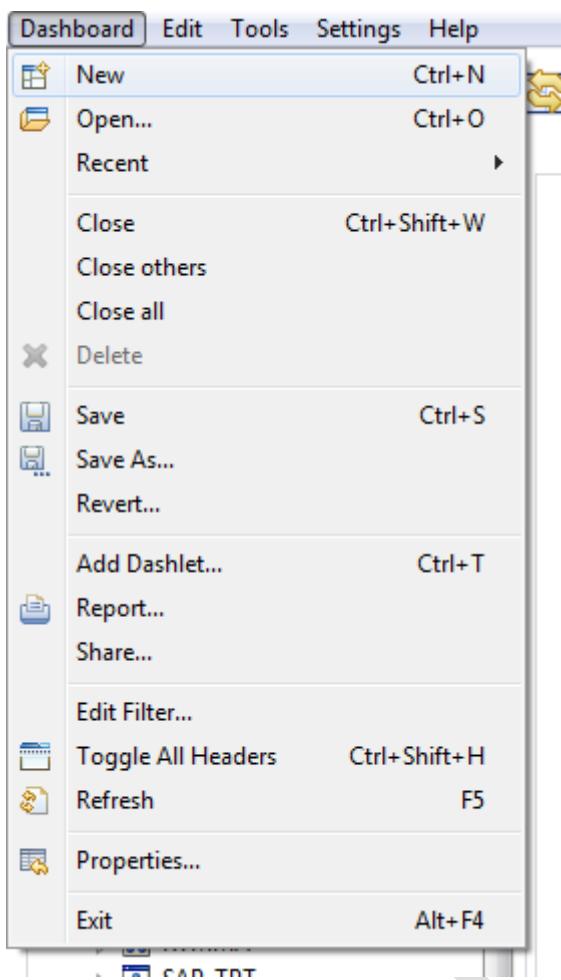
You can also select any specific server and view further details like the applications running on each of these servers, the OS and its version running on the server, number of CPU cores, IP Address of the server, number of applications processes running and also their type.

This screenshot shows the detailed host information for b0185-app0098-s. It includes sections for System, Performance, and Applications. The System section lists the OS as Windows Server 2008 SP2, OS Version as 6.0.6002, and OS Architecture as x64. The Performance section shows Physical Memory at 16 GB, CPU Cores at 8, and IP Address at 10.68.22.18. The Applications section shows 11 Application Processes, with a breakdown of Java: 0, .NET: 11, and Native: 0. A tooltip for the Java count indicates 'Java: 0'.

You can further select any application running on the server. Here you can see application process specific details like process id associated with this process, Garbage Collection occurred at various generation levels, Suspension time caused due to Garbage Collection, CPU Used by this application process and also thread count.



To monitor the application i.e. the exceptions thrown during the test, errors generated, webservices being called, webrequests, response times, graphs for CPU consumption at the process level, we have to create a new “Dashboard”. In the menu bar, click on Dashboard → New to create a new Dashboard.



In the next screen select either of “Use for Analysis” or “Use for Monitoring” as per requirement.



Use for Analysis

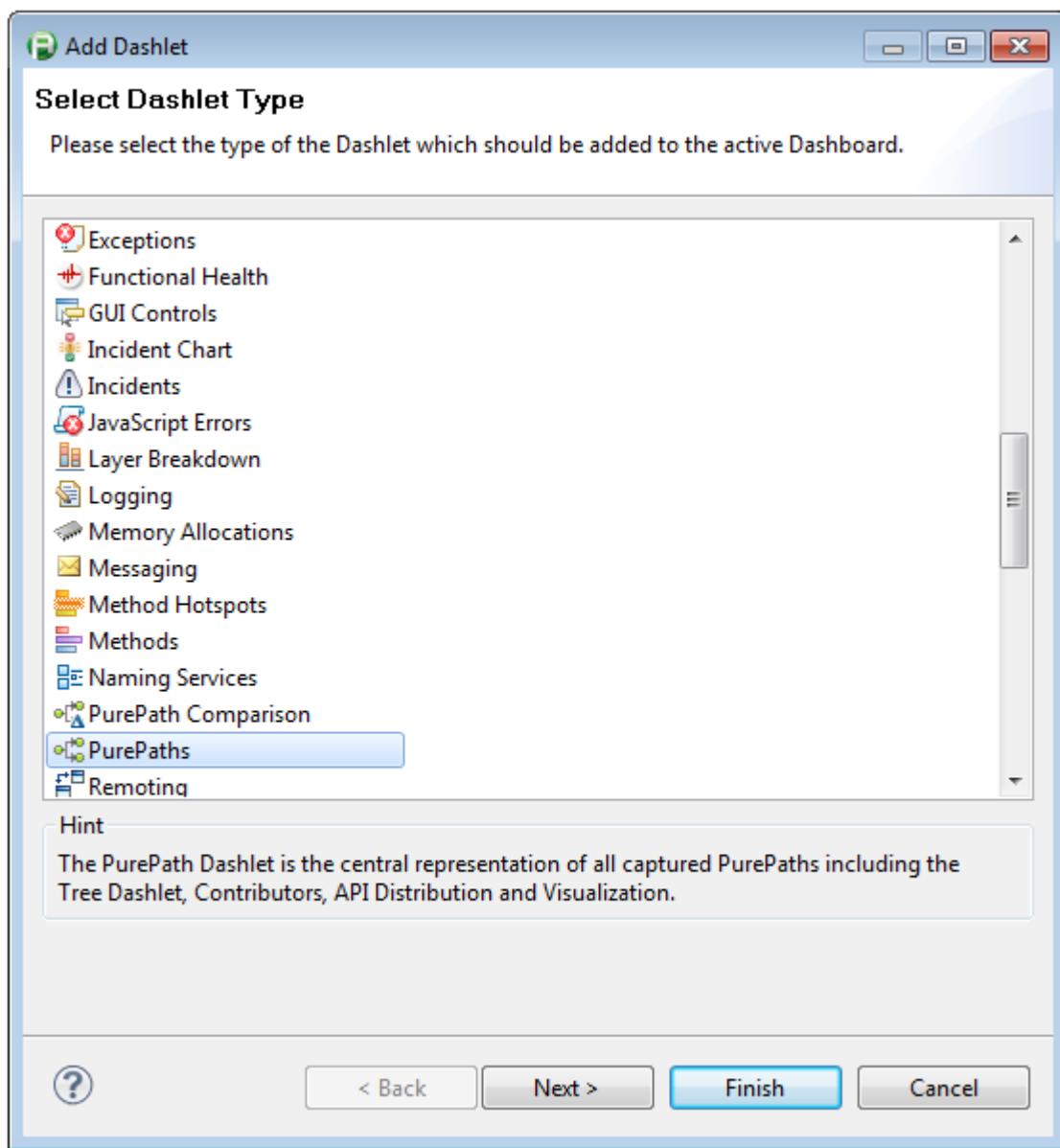
Dashboard settings will be preconfigured for typical analysis use. These settings can be changed later on.



Use for Monitoring

Dashboard settings will be preconfigured for typical monitoring use. These settings can be changed later on.

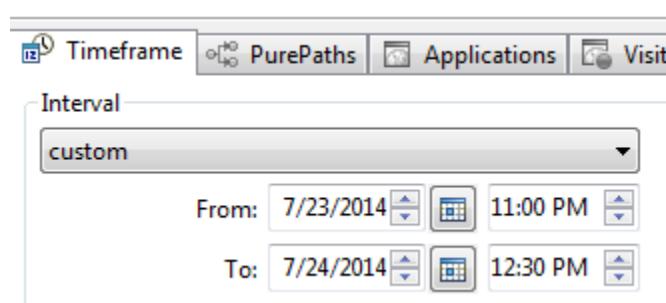
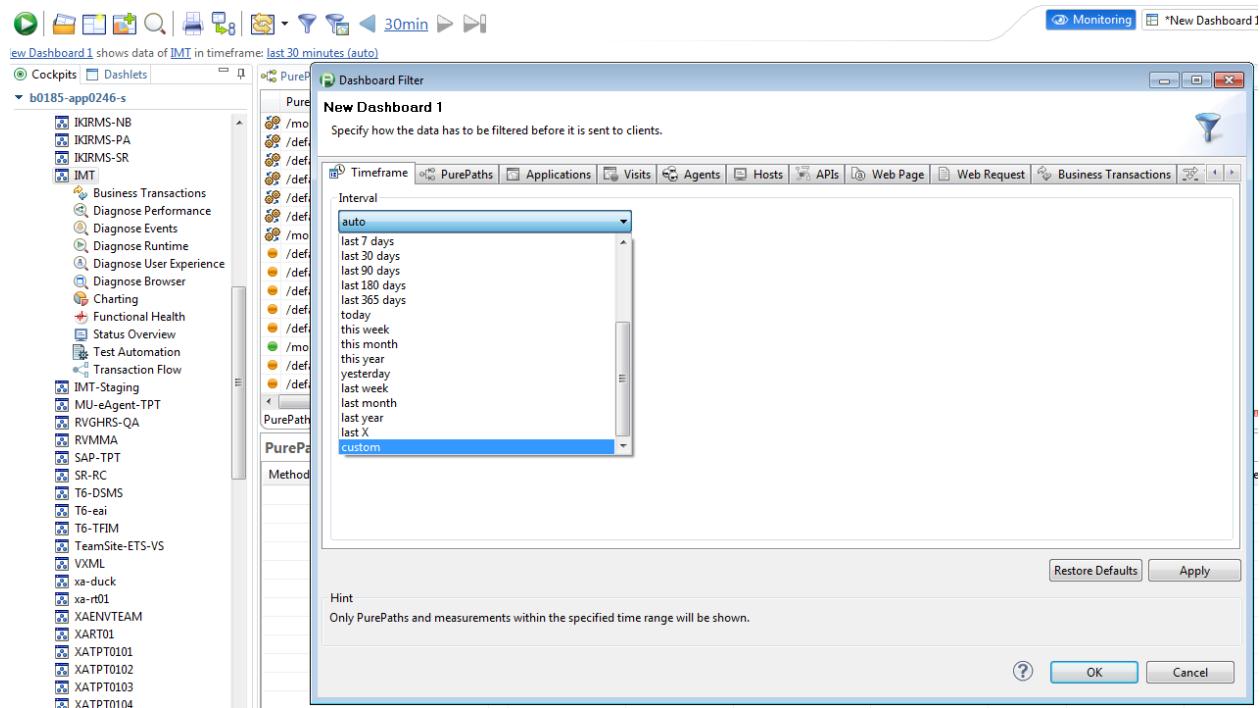
After this you will be asked to select a “Dashlet” for your dashboard. Initially you can select any of the dashlet among the various options. Later you can add new dashlets, delete existing dashlets.



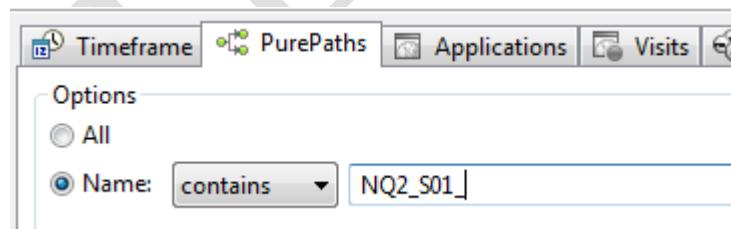
In the dashlet, details specific to the dashlet for the last 30 minutes will be present. You can further alter the time duration for which you want view data. You can also select a past date/time during which a test would have executed a week ago.

Please Note: Since the dynatrace client is configured on your local desktop, the client would consider your local desktop time for fetching any details stored on the dynatrace server. So, if a test is executed during EDT time (assume a test was executed during 23/07/2014 13:28:49 - 23/07/2014 14:55:01 EDT) and your desktop is configured in IST time zone, then to fetch the

details of the test, you must convert the test execution time in EDT to IST(23/07/2014 10:58:49 PM - 24/07/2014 12:25:01 AM IST) before fetching the sessions.



You can also fetch details specific to a particular transaction by entering the purepath name for the specific transaction.



Further filtering of data can be done specific to a particular application etc.

It is possible to drill further down from a purepath to an exception thrown when the pure path is executed, or from an exception to the webservice that throws the exception, or the webrequests

executed for a particular transaction. Just right click on the exception/purepath/webservice/webrequest and drill down as shown below.

The screenshot shows the PurePaths dashboard interface. A context menu is open over a row in the main table, specifically for the entry 'NQ2_S15_Auto_FQ_1D1V_QP_Options_IL_T08_SLA4...'. The menu items include:

- Details...
- Drill Down
- Show in Dashboard
- Add Snapshot to Dashboard
- Export...
- Store Session...
- Result Limit (PurePaths)
- Select PurePath for comparison
- Compare PurePaths
- Find... (Ctrl+F)
- Select all
- Deselect all
- Copy (Ctrl+C)
- Refresh Dashlet (Ctrl+F5)
- Dashlet Properties...
- Toggle Header (Ctrl+H)

The main table displays several rows of purepath data, each with a green circular icon, a name, a response time (e.g., 5020.58 ms), a breakdown chart, a size (e.g., 3540), and an agent (e.g., IM_purchaseA). The breakdown chart for the selected row shows 'cpu' (blue) and 'io' (orange) components.

The right side of the interface features a sidebar with various navigation links:

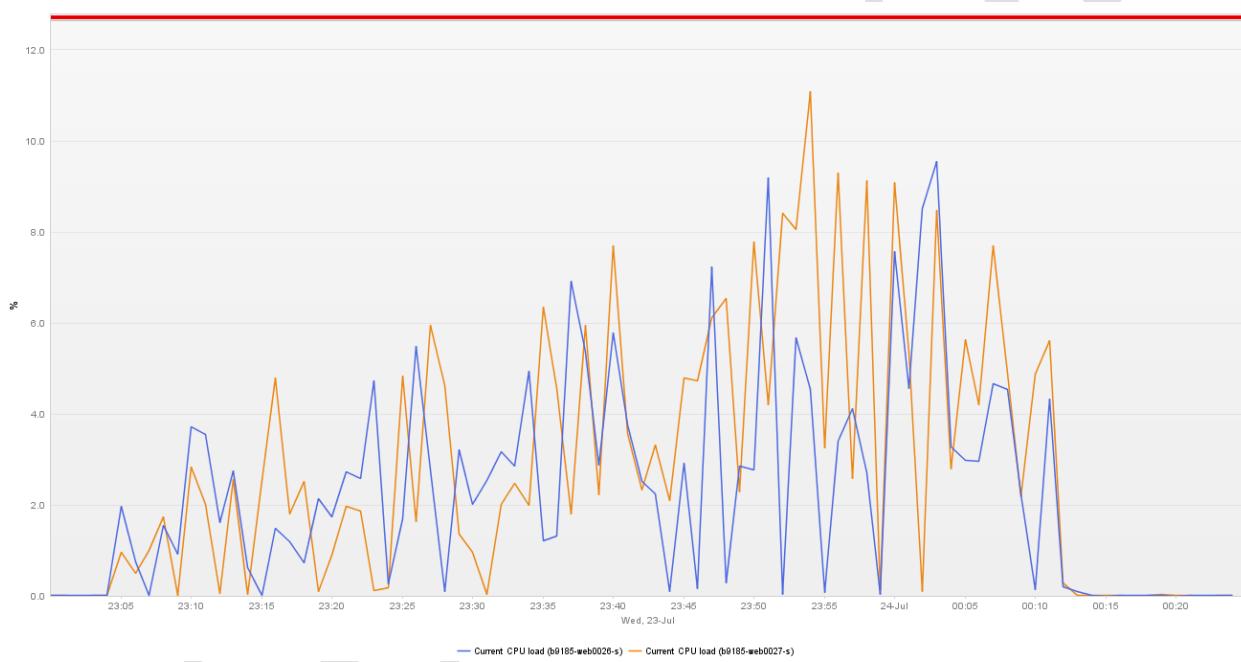
- Transaction Flow
- Agent Breakdown
- API Breakdown
- Business Transactions
- Components
- Database
- Database Hotspots
- Entry Points
- Messaging
- Method Hotspots
- Methods
- Naming Services
- PurePaths
- Remoting
- Response Time Hotspots
- Sequence Diagram
- Tagged Web Requests
- Web Requests
- Web Services
- Exceptions
- Logging
- Runtime Suspensions
- Browser PurePaths
- JavaScript Errors

Also, when you select a purepath, a purepath tree in the window below shows the flow of the purepath, the requests executed, the DB calls, webservices request and response details along with the time consumed for each of these actions to execute.

PurePath Tree (showing only relevant nodes)								
Method	Argument	Exec Total [ms]	Breakdown	Class	API	Agent	Elapsed Time [ms]	
ProcessRequestNotificationHelper(IntPtr, IntPtr, IntPtr, int)	/auto/Auto/Overl...	5657.36	io (91%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.00	
DisposeHandler(HttpContext, IntPtr, RequestNotificationStatus)		0.01	io (100%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.07	
ProcessRequestNotification(IntPtr, IntPtr, IntPtr, int)		0.15	cpu (98%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.09	
ProcessRequestNotificationHelper(IntPtr, IntPtr, IntPtr, int)	/auto/Auto/Overl...	0.06	cpu (94%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.18	
DisposeHandler(HttpContext, IntPtr, RequestNotification)		0.00		PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.23	
ProcessRequestNotification(IntPtr, IntPtr, IntPtr, int)		0.12	cpu (97%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.25	
ProcessRequestNotificationHelper(IntPtr, IntPtr, IntPtr, int)	/auto/Auto/Overl...	0.03	cpu (87%)	io	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.34
DisposeHandler(HttpContext, IntPtr, RequestNotification)		0.00		PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.36	
ProcessRequestNotification(IntPtr, IntPtr, IntPtr, int)		0.13	cpu (97%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.38	
ProcessRequestNotificationHelper(IntPtr, IntPtr, IntPtr, int)	/auto/Auto/Overl...	0.08	cpu (96%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.42	
DisposeHandler(HttpContext, IntPtr, RequestNotification)		0.00		PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.50	
ProcessRequestNotification(IntPtr, IntPtr, IntPtr, int)		0.18	cpu (98%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.52	
ProcessRequestNotificationHelper(IntPtr, IntPtr, IntPtr, int)	/auto/Auto/Overl...	0.12	cpu (97%)	PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.56	
DisposeHandler(HttpContext, IntPtr, RequestNotification)		0.00		PipelineRuntime	ASP.NET	IM_purchaseAllsta...	0.68	

It is also possible to generate graphs for various system resources consumption during the test.

You have to use the “Charting” dashlet to generate graphs.



Please read the below section on how to save/store the dynatrace sessions for a particular duration at a shared location.

Every project that has been configured on dynaTrace has a unique profile name.

Select the profile name for which you want to export the session. The profile name can be found under “System Profiles”. For Encompass Dashboard, the profile name is XH_EncDashboard.

When you expand the Profile, you can see various tabs as shown below.

The screenshot shows the Dynatrace interface with the 'Analysis' tab selected. The left sidebar lists various system profiles and a session storage section. The 'XH_EncDashboard' profile is expanded, revealing a detailed list of monitoring and diagnostic tools. The right panel displays a list of PurePaths, with the first item being '/foragen' repeated 12 times.

You can go to any of the desired tab to see the corresponding details.

When you are exporting the session, you have to right- click on the profile, and select “Session storage” → “Export Session”.

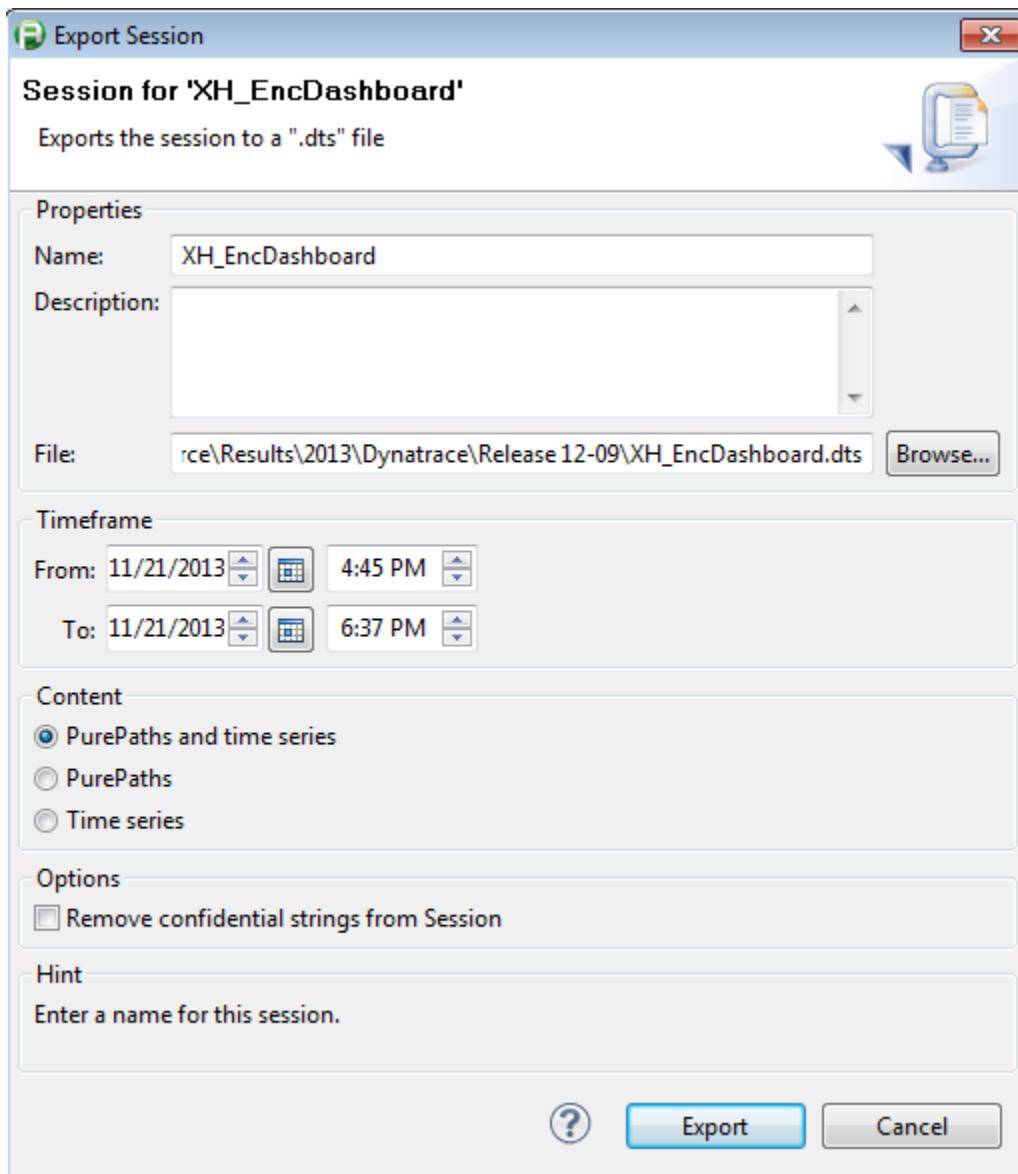
The screenshot shows the dynatrace interface with the following details:

- Top Navigation:** Cockpits, Dashlets, Method Hotspots, PurePaths, Exceptions, Errors, Tra.
- Left Sidebar:**
 - b0185-app0247-s**
 - Status Overview:** Incidents, Tasks and Monitors, Agents Overview, System Information
 - System Profiles:** ALB_Benefits, BW-Siebel, dynaTrace Self-Monitoring, E8-EESS, EncClaims, NGTPT-D <Def>, NGTPT-E, NGTPT-F, PCIPS-TPT, PMT-Staging, XH_EncDashboard
 - XH_EncDashboard:** Business Tr., Diagnose P., Diagnose E., Diagnose R., Diagnose U., Diagnose B., Charting, Functional, Status Overview, Test Autom., Transaction
 - Session Storage:** Session Storage, Hot Sensor Placement, Synchronize Discovery Data On Collectors, Disable System Profile, Export System Profile..., Rename System Profile..., Active Configuration <Default>, Details..., Find..., Select all, Deselect all, Expand All, Collapse All, Copy (Ctrl+C), Paste (Ctrl+V), Remove, Delete, Refresh Dashlet (Ctrl+F5).
 - Bottom Status:** Offline, 9 rows found (1 row selected)
- Right Panel:**
 - PurePath:** /foragents/logout.jsp, /foragents/callwrapper.jsp, /foragents/logout.jsp, /foragents/callwrapper.jsp, /foragents/logout.jsp, /foragents/callwrapper.jsp, /foragents/services/customer_inquiry/search_results..., /foragents/customer inquiry XMIM.jsp
 - Session Storage Context Menu:** Store Session..., Export Session... (highlighted in blue).
 - Only relevant nodes:** Argument

Once you select “Export Session” you get a new window as shown below where you are asked to enter the file path where the session has to be stored. The file will be stored with “.dts” extension.

Also, enter the duration of the test execution to store the session under Timeframe.

Note: The duration of the test executed in controller is in CST timezone and if you are exporting the dynaTrace session and the time zone of your machine is not in CST, then convert the corresponding CST time to your local time zone and then export the session(because the client is configured to your local time zone).



Under “Content” select “PurePaths and time series” or anything as per your requirement and finally click on “Export”.

3. HP Shunra:

HP Shunra is created by Shunra, acquired by HP.

Before 11.0, we used to call it as WAN Emulator.

After 11.0 onwards, we used to call it as Network Virtualization.

Steps:

- Install Shunra in Controller.
- Install Shunra agent in Load Generator.

Prerequisites: We have to create “.ntx” file based on the locations.

Steps to browse .ntx files:

- Open the controller
- Design the scenario
- Go to LG
- Click on details
- Choose network virtualization option
- Choose professional option
- Browse .ntx files

Once the test is finished, filter the response times based on the response times and report the same.

Steps to use Shunra in controller:

- Design the scenario in controller
- Click on network virtualization/location virtualization
- Choose per group
- Create profiles by configuring the .ntx files
- Assign the profile against every script
- Conduct the test

Open .lra file, filter the response times based on host names and report the same.

4. Perfmon:

It is a default monitoring tool to monitor windows operating systems based servers.

- Start
- RUN
- Type Perfmon
- Click ok.

Process:

- Right click on counter logs
- New log settings
- Provide the output file name
- Add servers & objects along with counters
- Define time interval
- Provide the output file name and Schedule settings.

CVS/VSS (Concurrent Versioning System/Visual Source Safe)

All these tools supports change management system. It facilitate end users to work on current version of file.

Step 1: Once you create a file, you have to check in to CVS

Step 2: Your teammate will checkout same file into his local machine and implement to changes.

Step 3: He has to check in modified file into CVS.

Note: At any point of time, only one person can work on current version of file.

Bottlenecks

Bottleneck is an issue which causes delays Response Time.

We have to find bottlenecks in below areas:

1. Network
2. Hardware
3. Code/Application
4. Configuration Settings

1. OACore issue: Oacore process the requests and consumes the memory from JVM heap.

Scenario 1: I have an application called oracle R12/oracle apps/Ebiz which has to support 300 users, expected response times are below 5 seconds. While ramping up the users, application performance is degraded at 100 user load. Even application itself crashed at 150 user load.

Process: Initially we started analyzing client side statistics and found that issue with server side. Then we analyzed OS/hardware/code/DB but we did not find anything. So moved to DB layer, finally we found an issue with JVM settings which is a oacore settings (memory management configuration).

After analyzing oacore settings, we recommended them to tweak that setting from oacore = 1 to oacore = 3 which resolved the issue.

2. Cursor Limit Issue:

We have an application which has to support 500 users

Solution: We started test with 500 users but application got crashed at 150 user load and application performance degraded at 100 user load.

We have started analyzing client side, OS, hardware, memory, method, level statistics which didn't provide any clue. Then we started analyzing configuration settings in HTTPD file in Tomcat Apache server and database statistics. Then moved to database configuration settings.

We found an issue with cursor limit which is causing application crash.

Process: We used dynatrace profiling tool to monitor all the layers of application including configuration settings as part of testing while ramping up users, cursor limit usage reported 100% at 120 user load. Then we asked DB team to provide the cursor limit setting which is reported 50. By default cursor limit is 50 which can accommodate only 100 users. We recommended them to change 50 to 2000 as per oracle SGA, they can tweak to 1000 for that application.

After deploying new build with 1000 cursor limit supported 500 users without any performance degradation.

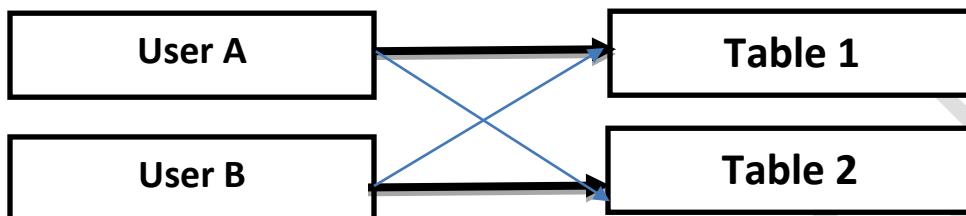
3. Dead locks:

Dead lock is a situation which causes longer wait or makes transaction failure.

Ex: if the user A locked on Table 1 requested lock on Table 2, which is locked by user B. Both are not willing to release the locks which will cause failure transaction.

With the help of DBA we can find Dead locks.

Dead locks occur due to DB poor design.Oratrace can be used to find Dead locks in AWR Reports.



4. To find out the best server for the application

Client asked me to find out which is the best server for their application

Objective: Maxout the resources in both of the servers to find out which is best for their application. We need to find out at what user load, at what hits per second and at what level of throughput servers are maxingout the resources.

Process: Initially we deployed the application in both the servers and conducted test independently with multiple users, we received max client error without reaching objective. Then we received error at controller “Server has shut down the connection prematurely”. Then we suggested install JVM to monitor at server level.

Again we conducted one more test with the help of JVisualVM monitoring support we found error message in the logs which is related to configuration settings.

Server is within minsparethread of maxclient consider raising the maxclient settings.

From the above observation, we found issue with configuration settings, asked them to provide Apache configuration settings.

After tweak the above settings for multiple times, we achieved objective of the test.

Conclusion: We reached the performance degradation point at 1500 user load for web logic server and 2000 user load for Tomcat Apache.

We recommended client Tomcat Apache is the best server for their application.

5. Memory leakage:

It is a concept deals with memory management to execute any program or task we require Memory.

Memory has to be de-allocated from unused variables or un-referencive variables by GC Engine.

If the memory is not de-allocating from un-referncive variables causes memory leakage.

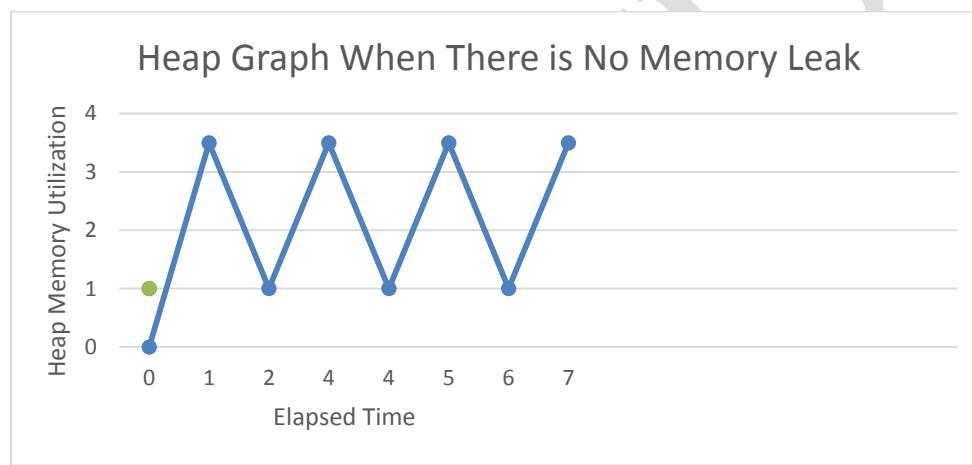
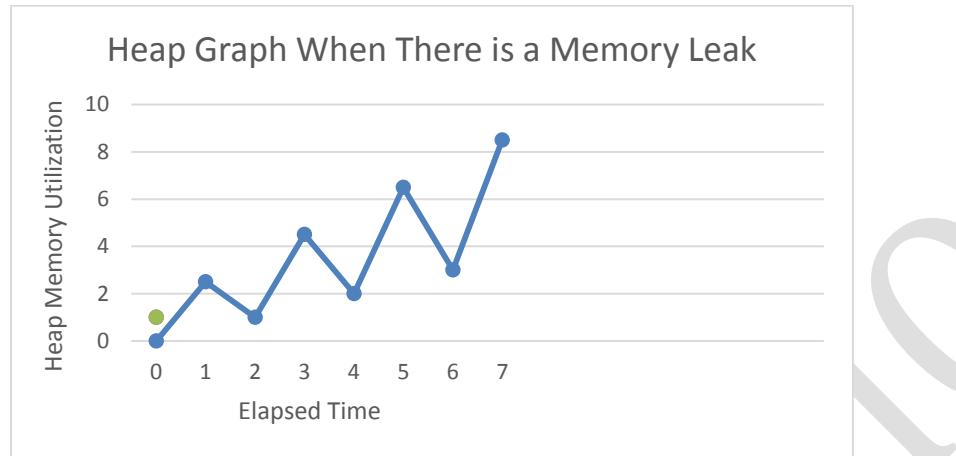
To find out memory leakage issue we have to conduct availability test.

GC Engine:

It is a mechanism automatically loads in to the system and de-allocates memory from un-returncive objects.

Note 1: How often GC Engine is loading in to the system.

Note 2: How long it is taking de allocating memory from the un-referencive objects.

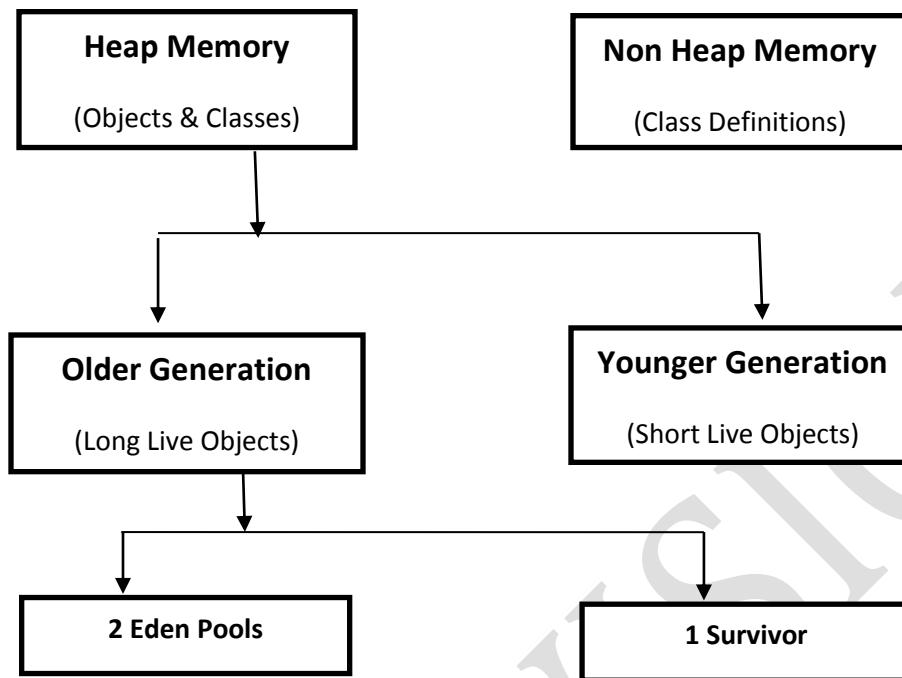


Heap graph should be in teeth pattern, and it should not be an upper trend graph.

We have various types of memory pools.

1. Heap memory
2. Non-Heap memory

Heap memory is having multiple memory pools



Garbage Collector Mechanism:

Garbage Collector is a method which will invoke automatically based on XML settings.

GC loads as

Half GC: Deallocates memory from younger generation

Full GC: Deallocates memory from both generations.

6. Thread limit:

We found an issue server reached the max number of simultaneous connections in server log.

Cause: Not enough threads in Apache.

Statistics: Due to the connection pool settings, application is not able to handle simultaneous connections created by 150 users.

Recommendations: Increase the connection pool settings from 10 to 16.

7. Heap memory issue in Tomcat:

In one of my business object application, we received an error “web intelligence is busy contact BO admin”.

Cause: High CPU utilization triggered by JVM process.

Recommendation: Upgrade the heap memory in template.

8. Database error occurred:

Database error received like “unable to bind configuration objects” and “unable to extend temp segment by 128 in table space temp_BO”.

Cause 1: Oracle SGI architecture

Cause 2: Java version

Recommendation: We received the oracle patch and applied the same even though we found an issue with oracle JDBC protocol violation.

AD team pull the logs and shared with oracle team received the new patch and used new ODBC jar file to check whether java 6 is backward compatible or not.

9. High CPU utilization due to I/O operation:

Process: My application reported high CPU utilization as part of drill down, we segregate the CPU utilization into messaging level and user level.

User level CPU utilization reported very high again we drilled down into I/O operations, method, EJB(Enterprise Java Bean), package level.

Cause: We found that one of the I/O utilizing more CPU.

Recommendation: We recommended AD people to verify whether really that I/O operation is required and when it is utilizing more CPU.

10. Private Byte Issue:

Process: In one of my SAP application, users are causing memory dumps as per SAP architecture, they configured private bytes 4mb memory. If the users are trying to retrieve more than 4mb memory, then the users will be pushed to run mode to debug mode. If SAP system allocates more memory, then the user will come back to run mode. To execute the report if the system is not allocating the memory, the users will be pushed to error mode which causes the memory dump.

Recommendation: Recommended SAP people to reconfigure private byte settings.

11. High CPU utilization:

To perform any action, we required processor time. If the processor is not available (100% utilized) to perform the task, it causes the application failure or impacts the transaction response time.

Recommendation: Upgrade the processor.

Counters:

1. Ideal CPU
2. System CPU
3. User CPU
4. Privileged Time
5. Processor CPU
6. CPU interrupts
7. Consumption of operation
8. Kernel mode CPU logs

Note

By default, every thread privileged CPU time will be allocated.

12. High Memory Utilization:

To perform any action, we required memory. High memory utilization impacts the application performance and transaction response time.

Recommendation: Upgrade the memory.

Counters:

1. Available Memory
2. Committed Bytes
3. Cached Bytes
4. Page faults per second
5. Page reads per second
6. Page writes per second

13. OOM (Out of memory) Exception:

In one of my application, we received OOM exception.

Process: As part of analysis, memory utilization reported 50% still causing OOM exception

Cause: Server is running on 8 GB ram with 32bit OS

Recommendation: We recommended them to upgrade the server from 32bit to 64bit to resolve the issue.

Note: 32 bit supports only 4 GB RAM

64 bit supports upto 1TB RAM

14. High physical disk queue length:

Disk queue length should be minimal.

In one of my application, we found high disk queue length which is causing high response times.

Physical disk queue length should not cross twice of the spindles.

Counters:

1. Disk read time
2. Disk write time
3. Average Disk bytes (Read and Write)

15. Processor queue length:

It should be minimal.

Verification: It should not cross twice of the cores.

Database bottlenecks:

1. Low buffer hit ratio:

Process: Generated AWR report and found low buffer hit ratio which is reported 60%. BHR should be more than 95%.

Cause: Due to allocation of low memory to the buffer memory, triggers low BHR.

Recommendation: Recommended DB architect people to increase buffer memory allocation by redesigning oracle SGA.

2. Low index utilization

Low index utilization can't monitor by AWR. With the help of DBA, we found index utilization issues and asked them to provide tuning opportunities in terms of index utilization.

Cause: Low index utilization causes delayed response time.

Recommendation: Redesign the index.

3. Full table scans

With the help of DBA, we found there are lot of full table scans.

Cause: Full table scan causes delayed response time.

Recommendation: Redesign the index and implement the index of which tables are frequently accessed.

4. High hard parses and soft parses

Cause: AWR reported high monitor and CPU parsing time which indicates that there are lot of hard parses happening.

Process: Whenever we fire a request, query tried to retrieve the information from buffered memory. If the data is not available in buffered memory, then it has to retrieve from the tables. To retrieve the data, it has to load as hard parsing or soft parsing.

Hard parsing:

- Loading into shared pool
- Syntax verification
- Verify the table authorization
- Transformation of sql queries from complex to simple
- Preparing execution plan
- Executing the sql query
- Fetch data from table

Soft parsing:

Except step 1, it will perform all above steps.

Recommendation:

- 1) Alter the shared memory pool size.
- 2) Verify the binding variables in query.

5. High I/O operation queries

Drill down

Why queries are utilizing more CPU for any I/O operation?

How much CPU utilized by the I/O operation?

Recommendation: Recommended fine tune the queries of I/O operations.

6. High Elapsed running queries

Cause: AWR reported high elapsed running queries which causes the delayed response time.

Process: We gone through the queries which contains lot of binding variables, inner queries which causes the delayed response time.

Recommendation: Recommended DB people to rewrite the query and tuning the inner queries and binding variables.

7. High CPU utilized queries

Cause: AWR reported high CPU utilized queries which causes the delayed response time.

Process: Once we gone through the AWR report which contains more number of high CPU utilized queries, we segregated that into DB level and machine level.

Recommendation: Recommended DB people to tune the query which causes high CPU utilization.

8. DB time and DB CPU time

Cause: AWR reported high DB time and DB CPU time which causes the delayed response time. It should be minimal.

Recommendation: Recommended DB people to redesign the DB architecture.

COUNTERS

1. JVM COUNTERS:

Process CPU Time:

Indicating the total amount of CPU time consumed by JVM.

Garbage Collector time (Garbage Collection Time):

Indicates the cumulative time spent on garbage collection and total number of innovations (invokes).

Current Heap Size:

Indicates the no of kilobytes (Kb) occupied by objects.

Free memory:

Available memory in the heap.

Garbage collector interval Time: time difference between garbage collection cause.

2. CLR (Common Library Runtime) Counters:

- **Exception through per second:** Indicating number of managed code exceptions thrown per second.
- **Timing GC:** Time spent on garbage collection.
- **.net CLR memory heap size:**
- **.net CLR total committed bytes**
- **.net CLR large object heap size**

3. WEB SERVER COUNTERS:

Apache:

- **CPU load:** Percentage of the CPU consumed by apache server.
- Request per second: The total number of request per second served by apache.
- Bytes per second(throughput)
- Busy workers: Number of active threads serving the request.
- Ideal workers: Number of inactive threads in the apache.

IIS SERVER (INTERNET INFORMATION SERVICES):

NOTE: IIS is a web app server for .Net based application.

- Bytes sent per second.
- Bytes received for second.
- Current connections.
- Request per second disconnection ratio.
- Number of request queued.
- Number of requests rejected.

Anonymous Users:

Indicating anonymous http connections in the particular tab.

4. APPLICATION SERVER COUNTERS:

Web logic server:

- **Execute the thread total count:** Indicating the total number of threads assigned to queue.
- **Pending request current count:** Indicates number of pending request in queue
- **Queue length:** number of request in the priority queue.
- **Throughput:** number bytes received per second.
- **Exception count:** It should not cross 20.
- **Connections current count:**
- **Transactions roll back total counts:** it should not cross 5.

WEBSPHERE (IBM WAS) COUNTERS:

- **Concurrent request:** Number of requests are concurrently processed.
- **Service time:** Response time for a servlet request.
- **Active count:** Number of active threads in the system.
- **Connection pool size:** Number of threads in the pool.

5. Network Counters:

- **Connection established:** indicating connections success ratio.
- **Connections failure:** Percentage of connections failure.
- **Through put:**
- **Network latency (delay):**
- **Pocket loss:**

6. Disk Counters:

Disk read per second:

Disk writes per second: rate of write operations on the disk.

Average disk queue length: Number of read and write request that were queued for selected disk during the sample interval.

Disk time: percentage of the elapsed time that the selected disk was busy with serving read and write request.

Note:

Average disk queue length should not cross to for every disk.

Split I/O's Second: Measures the rate of I/O split due to file fragmentation.

Free space: Display the percentage of total available space.

7. Database Counters:

- We can monitor oracle 10g versions using DB stats reports above 10g.
- AWR reports.

Note:

Even DBA can generates the DB trace (or) oracle trace report to identify dead locks and full table scan.

Oracle counters (for all java based applications):

- Buffer hit ratio
- Full table scan
- Indexing
- DB time
- DB CPU time
- Hard parses and soft parses
- Top 5 time taken events(in a particular durations)
- Physical read
- Physical writes
- High CPU utilized query's
- High memory utilized query's
- High I/O's utilized query's

SQL Server Counters (.NET based applications):

Using sql profile we can monitor DB activities.

Navigations:

- Open SQL server
- Choose new option
- Choose create new profile.

Counters:

Buffer cached hit ratio.

Transactions for second.

Log cached hit ratio.

Page read per second.

Page writes per second.

SAP HANA Counters:

Using SAP HANA studio we can monitor SAP HANA.

SAP HANA built on column and row based technology.

But purely work on column based.

Protocols

1. SAP Web Protocol:

SAP web protocol same like as Web HTTP/HTML protocol. If the application is developed as a ECC portal, net weaver portal, dynapro portal. We have to use SAP Web protocol. If the LR failed to record the objects using Web HTTP/HTML then only we can use SAP protocol.

The SAP-Web Vuser script typically contains several SAP transactions which make up a business process. The business process consists of functions that emulate user actions. For information about these functions, see the Web functions in the Function Reference.

Note: You can generate a SAP - Web Vuser script by analyzing an existing network traffic file (capture file). This method may be useful for creating Vuser scripts that emulate activity on mobile applications.

Common correlation values in SAP Protocol:

1. SAP_exit_sid
2. SAP_context_id
3. SAP_securid
4. Window_id
5. Event queue [WD1101]

How to handle window Id?

Window id is a 13 digit time stamp which will generates in millisec.

Ex: windowed=144333121987

Challenges (or) Scripting Technique:

In one of the ECC portal SAP_exit_sid capturing as ABCD11234566789PLNNO.

Whenever I am trying to convert the above value HTML to URL, URL to HTML. Which is not happening properly.

To overcome the above scenario. I have to search for a character in the captured value, replace with expected character.

To find and replace a character, we written the “C” code which will automatically search for string and replaces with other string. Once it replaced we are converting to the Lr variable & substituting whenever we required.

The following example shows a typical recording for an SAP Portal client:

Example:

```
vuser_init()
{
    web_reg_find("Text=SAP Portals Enterprise Portal 5.0",
        LAST);
    web_set_user("junior{UserNumber}",
        lr_decrypt("3ed4cfe457afe04e"),
        "sonata.hplab.com:80");
    web_url("sapportal",
        "URL=http://sonata.hplab.com/saportal",
        "Resource=0",
        "RecContentType=text/html",
        "Snapshot=t1.inf",
        "Mode=HTML",
        EXTRARES,
        "Url=/SAPPortal/IE/Media/sap_mango_polarwind/images/header/branding_image.jpg",
        "Referer=http://sonata.hplab.com/hrnp$30001/sonata.hplab.com:80/Action/26011[header]"
        , ENDITEM,
        "Url=/SAPPortal/IE/Media/sap_mango_polarwind/images/header/logo.gif",
        "Referer=http://sonata.hplab.com/hrnp$30001/sonata.hplab.com:80/Action/26011[header]",
        ENDITEM,
        ...
        LAST);
```

The following section illustrates an SAP Web and SAP GUI multi-protocol recording in which the Portal client opens an SAP control. Note the switch from **web_xxx** to **sapgui_xxx** functions.

Example:

```
web_url("dummy",
    "URL=http://sonata.hplab.com:1000/hrnp$30000/sonata.hplab.com:
    1000/Action/dummy?PASS_PARAMS=YES=&dummyComp=dummy=&
    Tcode=VA01=&draggable=0=&CompFName=VA01=&Style=sap_mango_polarwind",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=http://sonata.hplab.com/saportal",
    "Snapshot=t9.inf",
    "Mode=HTML",
    LAST);
```

```
sapgui_open_connection_ex(" /H/Protector/S/3200 /WP",
 "",  
 "con[0]");  
sapgui_select_active_connection("con[0]");  
sapgui_select_active_session("ses[0]");  
/*Before running script, enter password in place of asterisks in logon function*/  
sapgui_logon("JUNIOR{UserNumber}",  
 "ides",  
 "800",  
 "EN",  
 BEGIN_OPTIONAL,  
 "AdditionalInfo=sapgui102",  
 END_OPTIONAL);
```

2. SAP GUI Protocol:

SAP GUI is a client server application SAP logon pad version is 7.3

The SAP GUI Vuser script typically contains several SAP transactions which make up a business process. A business process consists of functions that emulate user actions. Open the **Step Navigator** to see each user action as a Vuser script step.

The following example shows a typical recording of a SAP GUI client. The first section, **vuser_init**, contains the opening of a connection and a logon.



Note: The Open Connection step uses one of the connection names in the SAP Logon **Descriptions** list. If the specified connection name is not in the list, the Vuser looks for a server with that name.



In the following section, the functions emulate typical user operations such as menu selection and the setting of a check box.



The final section, **vuser_end**, illustrates the logoff procedure.



When recording a multi- protocol script for both SAP GUI and Web, VuGen generates steps for both protocols. In the Script view, you can view both **sapgui** and **web** functions.

The following example illustrates a multi-protocol recording in which the SAP GUI client opens a Web control. Note the switch from **sapgui** to **web** functions.

Prerequisites to work on SAP GUI protocol:

- I. We have to install SAP logon pad.(VUgen as well as in LG's)
- II. Configure in the connection string is SAP logon pad.
- III. Enable client side , server side scripting.
- IV. Change the notifications “ask to allow”.

Configure Connection String:

- I. Get the connection string from client.
- II. Right click => add new entry.
- III. Change connection type to “Group/server selection”.
- IV. Click finish button by providing description ,messege server , SAP router.
- V. By importion .ini file also we can configure the connection.
 - ⇒ Goto the SAP logon options , navigation to the accessibility & scripting => check enable scripting.
 - ⇒ **Server side scripting :**
Login to application
Using R2 level T-code we can enable the server side scripting. “RZ11”.

Example:

```
sapgui_tree_double_click_item("Use as general WWW browser, REPTITLE",
  "shellcont/shell",
  "000732",
  "REPTITLE",
  BEGIN_OPTIONAL,
  "AdditionalInfo=sapgui1020",
  END_OPTIONAL);

...
sapgui_set_text("",  

  "http:\\\\yahoo.com",
  "usr/txtEDURL",
  BEGIN_OPTIONAL,
  "AdditionalInfo=sapgui1021",
  END_OPTIONAL);
```

3. Web Services Protocol:

If the application is communicating with the help of 3rd party services (XMB/Rest Ful XMLs/JSONs) we have to use web service protocol.

If the application is developed in SOAP. We can develop the scripts in 2 ways using below protocols.

1. Web HTTP/HTML.
2. Web Services.

1. Web HTTP/HTML Protocol:

Pre-requisites:

1. We have to request the project team to provide the WSDL file for business flow.
2. Validate the WSDL file in SOAP UI tool.

SOAP UI (Versions 3.5/4.6):

- I. Open the SOAP UI.
- II. Create new project.
- III. New Project.
- IV. Provide the project name.
- V. Provide the WSDL File Path.
- VI. Click on OK button.

Select the XML in request plan. Execute the XML by providing authentication details and method type.

Verify the response whether it is correct or not by validating HTTP status code & XML.

1) Web HTTP/HTML Protocol:

We can create the same request using Web (HTTP/HTML) protocol. You need take the xml request as shown in the following and place in the web custom request.

You need to keep this soap request in the web request body. As shown below and you can capture all the response using correlation function. You can also add check point using web_reg_find. The url should end with .asmx as shown in the request.

- i. Copy the web_custom_request() syntax to the Vugen.
- ii. Copy the end path URL, and paste into web_custom_resuest() url.
- iii. Copy the XML, and paste into web_custom_request() body.

- iv. Use add_web_add_header() information for authentication and header information.

```
POST /globalweather.asmx HTTP/1.1
Host: www.webservicex.net
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.webserviceX.NET/GetWeather"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001
<soap:Body>
    <GetWeather xmlns="http://www.webserviceX.NET">
        <CityName>string</CityName>
        <CountryName>string</CountryName>
    </GetWeather>
</soap:Body>
</soap:Envelope>
```

XML Request Example for Web Custom Request

```
web_reg_find("Text/IC=New Delhi",
LAST);

web_reg_save_param_ex(
    "ParamName=Web Service Response",
    "LB=",
    "RB=",
    SEARCH_FILTERS,
LAST);
```

```
web_custom_request("Weather SOAP Request",
    "URL=http://www.webservicex.net/globalweather.asmx",
    "Method=POST",
    "TargetFrame=",
    "Resource=0",
    "RecContentType=text/xml",
    "Referer=",
    "Mode=HTML",
    "EncType=text/xml; charset=utf-8",
    "Body= Your Request should be here as shown in the following image"  

LAST);
```

```

web_custom_request("Weather SOAP Request",
    "URL=http://www.webservicex.net/globalweather.asmx",
    "Method=POST",
    "TargetFrame=",
    "Resource=0",
    "ReqContentType=text/xml",
    "Referer=",
    "Mode=HTML",
    "EncType=text/xml; charset=utf-8",
    "Body="

    |<soap:Envelope xmlns:xsi=\\"http://www.w3.org/2001/XMLSchema-
instance\\" xmlns:xsd=\\"http://www.w3.org/2001/XMLSchema\\" xmlns:soap=\\"http://schemas.xmlsoap.
org/soap/envelope\\\">
        "<soap:Body>
            "<GetWeather xmlns=\\"http://www.webserviceX.NET\\\">
                "<CityName>Delhi</CityName>
                "<CountryName>India</CountryName>
            "</GetWeather>
        "</soap:Body>
    "</soap:Envelope>",
    LAST);

```

Web Custom Request with Web Services

Note: Correlation, parameterization capturing and is same like web HTTP/HTML protocol.

2) Importing the service directly to vugen:

I have taken weather web service as example shown in this link.

- i. Open the Vugen.
- ii. Choose the web services protocol.
- iii. Choose the SOA tool.
- iv. Choose the Manage services.
- v. Import the services by clicking on “Import Buffer”.
- vi. By providing the URL of the services.
- vii. Provide the connection details if it require.
- viii. Click on add service call.
- ix. Provide the operation name, Input & Output arguments.

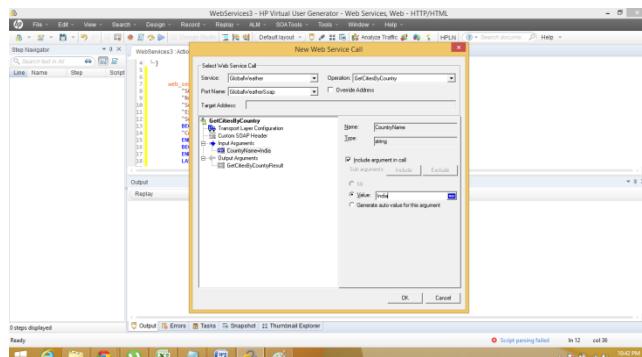
(or)

Steps:

1. First open the vugen and select Web Services protocol.
2. Click on Manage Services on top nav bar and Click on Import and give the WSDL URL which is usually ends with wsdl

http://www.webservicex.net/globalweather.asmx?WSDL.

Now Click on Add Web Service Call on top nav bar. Give the Input Arguments and Leave the out put arguments empty. And click on ok. As shown in the following image.



Input Arguements in Web Services

It will create a script in loadrunner as shown below. You can perform the steps one by one in this way for all the web services steps. You can use lr_xml_find and lr_xml_get_values to validate the page.

```
web_service_call( "StepName=GetCitiesByCountry_101",
  "SOAPMethod=GlobalWeather|GlobalWeatherSoap|GetCitiesByCountry",
  "ResponseParam=response",
  "Service=GlobalWeather",
  "ExpectedResponse=SoapResult",
  "Snapshot=t1396977083.inf",
  BEGIN_ARGUMENTS,
  "CountryName=India",
  END_ARGUMENTS,
  BEGIN_RESULT,
  END_RESULT,
  LAST);
```

Enhancements:

To capture the dynamic value from the above request. We used below request

```
Lr_xml_get_values("XML={response}",           // input argument
  "valueparam=outputparam",                   //output argument
  "query=/sendSMSResult",                   //query string like LB/RB
  LAST);
```

Verification Points:

```
Lr_xml_find("XML={response}",  
"value=name is",  
"query= //sendSMSresult",  
LAST);
```

3) By invoking the SOAP UI through Vugen:

1. Open the Vugen.
2. Invoke the SOAP UI by providing SOAP UI file path.
3. Then execute the all the requests under project.
4. Above process will allow you to record the objects with the help of SOAP request.

Note:

Correlation we can capture the dynamic values with the help of lr_xml_get_values().

4. Web click & Script Protocol

Web click & script protocol records the actions which are performed against the browser (or) it will record the only browser specific actions. There is no correlation values in Vugen script.

Functions:

- 1) **web_browser()**: Performs an action on a browser.
- 2) **web_edit_field()**: Enters data for a text field/ input purpose
- 3) **web_image_submit()**: Emulates a user clicking on an image that fires a submit request.
- 4) **web-image_link()**: Emulates a user clicking on an image that is a Hyper text link
- 5) **web_list()**: Select an item from a list control/Drop down list
- 6) **web_radio_group()**: selects one button from a radio button group.

Click & Script:

- 1) Records the browser specific actions
- 2) Correlation is not required
- 3) Recommended for GUI based apps.
- 4) Records the objects in terms of X.Y co-ordinates
- 5) Identified some deviations in response times when compared with HTTP Protocol

HTTP/HTML

- 1) Records the communication b/w client & server.
- 2) Correlation is required
- 3) Recommended for any web apps which is communicating in HTTP Protocol
- 4) Records the objects in terms of GET & POST Requests
- 5) It will give accurate Response times.

Record time options:**Recording**

- 1) **GUI Level:** It will generate a step for every user actions
- 2) **HTML Level**
- 3) **URL Level**

Challenges and Enhancements:

- While recording your script password will be present in encrypted format, we have to change to normal format. Ex: set value=jojo; set value= bean
- All the clicks will be recording in screen co-ordinates, we have to convert them to action format.

NOTE: If image co-ordinates are changing causes script failure.

Advantages:

- 1) Correlation is not required
- 2) Easy to understand & easy to maintain the script

Disadvantages:

- 1) Response times are slightly different from actual
- 2) It will generate object for every user action.

Pre requisites (or) Precautions while recording the script:

Which objects (Fields) would like to parameterize those fields should be modify while recording.

Case study 1:

I have an application which is developed in Java, my business scenario having 10 requests. Every Jsp page having 100 fields which are filling based on my previous input.

Solution: in the above scenario correlate 100s of values/ fields for every page is a different process.

To avoid conducting/ implementing correlation I switched from HTTP/HTML protocol to click & script protocol.

Case study 2:

I have an application in that application having multiple tabs. As per the business flow I have to move Tab1 to Tab2 which is not a server call.

Solution: In the above scenario Tab2 action is not a server call.

To perform continue button I have to navigate Tab1 to Tab2 which is not possible in HTTP/HTML protocol. So I moved to Click & script protocol.

Case study 3: If the application having GUI interface we can use Click & script protocol.

5. RTE (Remote Terminal Emulator) Protocol

RTE: If the application developed in Unix environment we have to use RTE Protocol.

Ex: Client server apps and cursor based apps (or) Min frame apps.

Recording Options:

- 1) Configuration
- 2) RTE: RTE having two options to handle synchronization. It will generate 2 automatic functions like
 - Te_wait_cursor()
 - Te_wait_text()

Options:

1) Cursor

2) Prompt

Cursor: It instructs the vugen to generate Te_wait_cursor() function to handle synchronization issue.

Te_wait_cursor() function instructs script execution flow wait for the cursor to be appeared at a specific location in terminal window.

Prompt: It instructs the vugen to generate Te_wait_text() function to handle synchronization issue.

Te_wait_text() function instructs script execution flow wait for the text to be appeared in a designated location.

NOTE: Above 2 functions automatically generate based on your record time options. Both the functions purpose is to handle synchronization.

Advantages: No correlation

Process to record a simple script:

File → New Script → Choose RTE protocol → Start record → Invokes the terminal Session → choose communication tab → click connect button → provide session type → Host name and port number → ok

Functions:

- 1) **Te_connect():** Connects the terminal emulator to the specified Host.
- 2) **Te_type():** sends a string to the terminal emulator (or) Input purpose.
Ex; Te_type("Username"):
- 3) **Te_wait_cursor():** Waits for the cursor to appear at a specified location in a terminal window.
- 4) **Te_wait_text():** Waits for a text string appear at a designated location.
- 5) **Te_wait_sync():** It instructs the vuser to wait until receives a response from server.

- 6) **Te_get_cursor_pos()**: Returns the current location of the cursor in the terminal screen.
- 7) **Te_set_cursor_pos()**: Sets the position of the cursor on the terminal screen.
- 8) **Te_getvar()**: Returns the value of the RTE system variable.

Challenges:

1) Synchronization issue.

Initially we used Record time options (prompt and cursor) to overcome the synchronization issue. But we fail to run the test in controller because of the synchronization issue.

Solution: We commented all Te_wait_cursor () & Te_wait_text() functions and written synchronization function called Te_wait_sync ().

2) To capture some text from on screen

Scenario: I have a scenario/situation to capture ASN number which is generated from server and I have to pass the same value in next request.

Solution: Using Te_get_text_line() function captured onscreen text passed it in which request we required.

2).Error handling:

Solution1: We can use Te_wait_text()/ Te_wait_cursor() functions to overcome /to handle the Exception pages.

Scenario: Every response should be validated before perform new action. We have to verify whether we receive a correct page or exception page.

Solution: To overcome this issue I developed an external function which will verify whether we received a correct page (or) Error page (or) warning page (or) Invalid page etc...

6. Citrix ICA Protocol

Citrix_ICA Protocol: If the application deployed in citrix environment we have to use citrix_ICA protocol. But my current project deployed in citrix. Before accessing this application we have to connect to the citrix env. Through citrix protocol.

Steps to Access the application:

- 1) Access the citrix Env. Through URL
- 2) Provide credentials & access the application which is published in citrix Env.
- 3) Perform the business scenario against the application.
- 4) Log off from the application and from citrix.

Pre requisites:

- 1) Install citrix_ICA agent in vugen as well as in LG machines (Same version of citrix)
- 2) Use same resolution (Screen) in vugen as well as in LG machines.
- 3) Avoid mouse clicks while recording the business scenario.

Functions:

- 1) **ctrx_nfuse_connect()**: connects to a citrix server via an NFUSE portal.
- 2) **ctrx_sync_on_window()**: waits until a window is created (or) become active.
- 3) **ctrx_mouse_click()**: Emulates a mouse click on a citrix client sent to a citrix server.
- 4) **ctrx_wait_for_event()**: It is a synchronization function that waits for an event to occur.
- 5) **ctrx_get_window_name()**: Retrieves the name of the active window.
- 6) **ctrx_type()**: Emulates a typing alpha numeric keys.
- 7) **ctrx_key()**: Emulates a non-alpha numeric keys.(Tab,uparrow,dowm arrow...)
- 8) **ctrx_sync_on_bitmap()**: waits until a bitmap appears.

Challenges:

1) My test got failed in controller

Cause: citrix_ICA agent version is different in LG & vugen machines.

2) My mouse click objects were failed in controller.

Cause: Resolution is different in LG & vugen machines.

3) Synchronization issue:

Case study 1: Synchronization issue is the biggest issue in citrix protocol. To overcome synchronization issue we have 4 ways.

- i) Based on window name
- ii) Based on bitmap value
- iii) Based on wait for an event function
- iv) Based on sync text

i) Based on window name:

I recorded a script for every action it generayed `ctrx_sync_window()` function. Which will wait for the expected window for a specified time.

To perform new action we have to verify whether the correct window appeared or not. For this verification i used function `ctrx_get_window_name()` to capture the active window name.

From the above if the active window name is expected window name then we can continue to the next action. Else we have to instruct the vuser to wait until specified window name to be appeared.

ii) Based on the Bitmap value:

Using bitmap hash value we can overcome the synchronization issue.

Using `ctrx_get_bitmap_value()` we can capture hash value of any image which can compare with expected Hash value.

iii) Based on wait for an event function:

`ctrx_wait_for_event()` function wait for a specified event to occur.

iv) Based on sync text():

waits until specified text is displayed around the specific position.

4) Challenge: To capture the specific text from on screen (or) capture dynamic value from screen

Solution: using `ctrx_get_text()` function to capture on screen text to handle text verification points (or) to capture dynamic value.

7. Oracle NCA Protocol

I have an application web interface is interacting with data base via oracle forms for this scenario I used Web+ Oracle NCA Protocol.

Functions:

1. **nca-connect-server ()**: Establishes the connection to Oracle NCA DB server.
2. **nca-set-window ()**: indicates the name of the active window.
3. **nca-obj-type ()**: sends keyboard input to an object.
4. **nca-edit-set ()**: sets the contents of an edit objects.
5. **nca-button-press ()**: Activates the specified push button.

Challenges:

1. Correlation for web objects.
2. NCA objects are recording in the form of ID's instead of object names.

1. Solution:

We have to append the record = Names to the URL. So that it will record objects as names.

Ex:

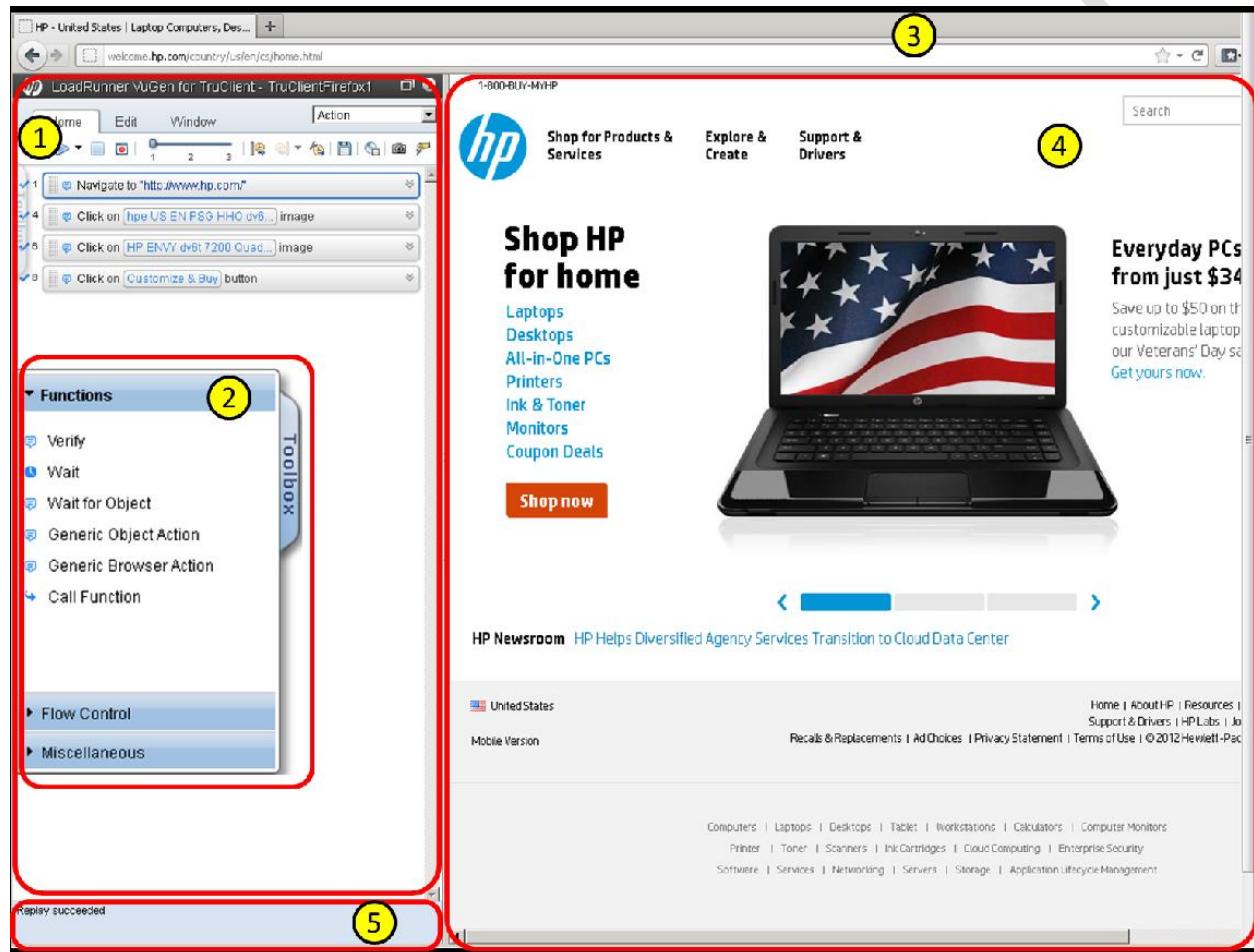
- 2: Set record= name is start up file.
In oracle form server find start up file called Base.html.
- 3: Set value in form web file and html start up file.

8. The Ajax TruClient User Interface

The Ajax TruClient user interface is made up of the following sections:

1. TruClient Sidebar. The heart of the interface, contains all the tools you need to develop your Ajax TruClient scripts.

2. TruClient Toolbox. To enhance your TruClient script by adding steps. The toolbox opens and closes by clicking on the tab, and moves by dragging it up or down.



3. Browser Navigation Bar. Enter the URL of the application for which you are developing a script.

4. Application Browser Window Develop and replay your script interactively.

5. TruClient Sidebar Status Pane. Displays status details about the active action in the TruClient Sidebar.

The following lists some of the main differences between the Ajax TruClient protocol and other VuGen protocols:

The script is visible in VuGen's script view but it is read only. The script is created and modified in the **TruClient sidebar** in the browser (FF or IE).

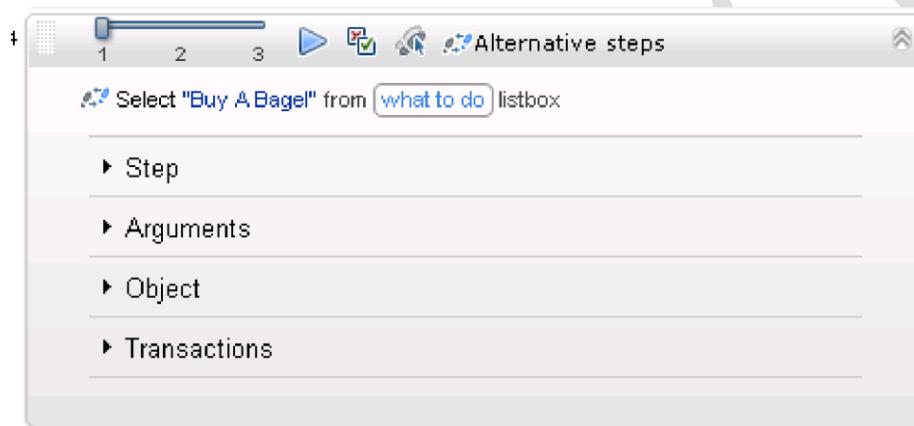
Ajax TruClient scripts are asynchronous. Steps do not have to wait for previous steps to complete. Each step defines an **End Event** which defines the point at which subsequent steps are allowed to start running.

Ajax TruClient scripts are recorded on the user level, therefore there are no correlations however the main challenge becomes object identification.

Most of the tasks involved in recording, replaying, and modifying scripts are done using the Ajax TruClient Sidebar for Internet Explorer or Mozilla Firefox.

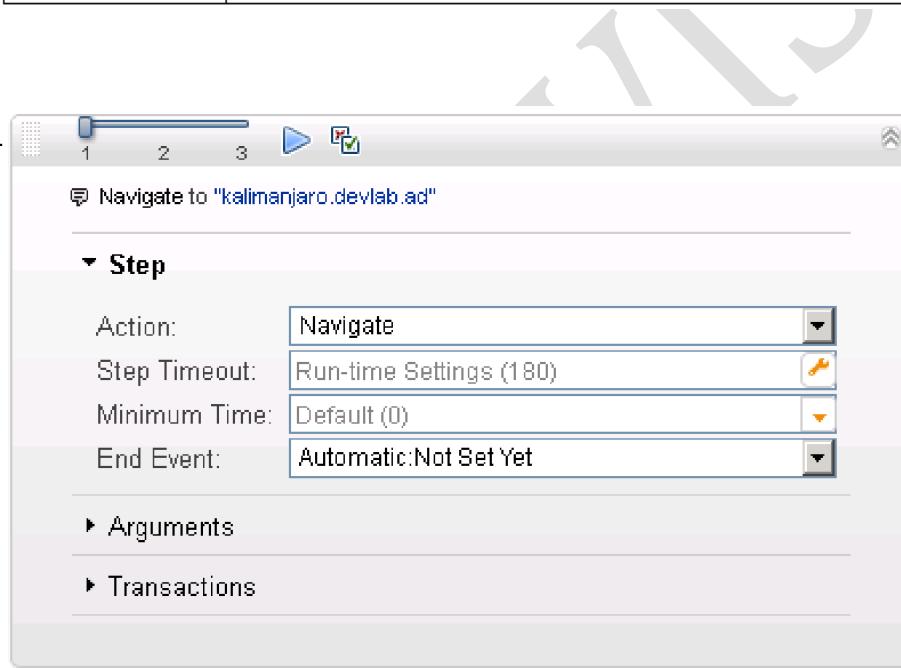
Ajax TruClient Step Structure

Ajax TruClient steps are comprised of a number of sections. The sections and elements within each section vary depending on the type of step.



User interface elements are described below:

UI Element	Description
	Drag Step. Enables you to arrange the order of your script by dragging the step to a different location.
	Expand Step. Displays the individual components of the steps which include step, argument and object.
	Script levels selector. Allows you to view and modify the script level of the step. For more information, see " Ajax TruClient Script Levels" on page 488.
	Replay. Replays this step only.
	Disable/Enable Step. Steps that are disabled are not replayed. This feature allows you to temporarily remove steps from the script without deleting them.
	Alternative Steps. This icon indicates a step which can be redefined in alternative ways. To redefine the step, click the icon, select the desired step definition, and click Back. For more information, see " Ajax TruClient Alternative Steps" on page 487.



Step	<ul style="list-style-type: none"> Action. The action that defines the step. The list of relevant actions is determined by the object roles. Object Timeout. If the object does not appear before this time in seconds, the step returns an error. Step Timeout. If the End Event is not reached by this time in seconds, the step returns an error. The way the script behaves when such an error occurs can be configured in the Run-Time settings dialog box. Minimum Time. The least time in seconds that the execution of the step will take. The value of this field can be either 0, "as recorded" or another manually set number.
Arguments	Contains step arguments. These arguments differ for different step actions and roles. For a list of the step arguments, see " Ajax TruClient Step Arguments " on page 525.
Object	<ul style="list-style-type: none"> Roles. The functions that Ajax TruClient understands about an object. This information is read-only and is updated dynamically depending on how the object is used during recording. The list of available step actions is defined by these roles. Name. A logical name for the object. This does not affect replay and can be modified to enhance readability. ID Method. The method of identifying the object. <ul style="list-style-type: none"> Automatic. Ajax TruClient's default object identification method. If this method does not successfully find the object during replay, click the Improve Object Identification button, reselect the correct object from the application, and replay the script again. XPath. Identifies the object based on its XPath expression that defines the object in the DOM tree. When you select this option, the next edit box in the display is labeled XPath and enables you to select an XPath to define the object. See below for details. JavaScript. JavaScript code that returns an object. When you select this option, the next edit box in the display is labeled JavaScript and enables you to define a JavaScript to define the object. See below for details. Related Objects. Tool to enable Ajax TruClient to identify a target object in relation to an anchor object. For details, see "How to Resolve Object Identification Issues" on page 492.
Transactions	Allows you to create, modify, and view transactions. For more information, see " How to Enhance Ajax TruClient Scripts " on page 505.

Ajax TruClient - Developing a TruClient Script How to Develop Ajax TruClient Scripts

This task describes the basic steps involved in interactively developing an Ajax TruClient script.

1. Create an Ajax TruClient script from the VuGen toolbar.

2. Configure the General Browser Settings

The Browser Configuration settings allow you to configure settings that apply to all Ajax TruClient scripts.

The settings are imported to new scripts as they are created.

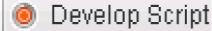
To edit these settings click the **Ajax TruClient General Settings** button from the Record toolbar in the

VuGen main window and select the **Browser Settings** tab.

3. Configure the Run-Time Settings

In VuGen configure the Run-Time settings before recording and performing a load test. To open the Run-Time settings dialog box, click **F4** or select the **Run-Time Settings** node from VuGen's **Solution Explorer**.

4. Start developing the script

Click  **Develop Script** to open an interactive recording session in the **TruClient Sidebar** for either the Mozilla Firefox browser or the Ajax TruClient browser for IE.

5. Record interactively

Navigate to the desired starting website and click  **Record** button . All of your actions will be recorded and displayed in the **TruClient Sidebar** on the left as you perform your business process. You can stop recording by selecting the **Stop** button  . You can continue recording from any point in the script.

To record into different section of the script, right-click a step and select **Record > Record after** or **Record > Record before** to begin recording new steps into the desired location in the script. If you are recording into a group step, select **Record > Record into**.

6. Replay the script

It is strongly recommended that you replay the script at least two times, correcting any errors that occur during the process. After two successful consecutive replays, you can move on to the next step.

During interactive replay, Ajax TruClient will animate each step progress according to its real running progress by filling the step bar. The step running progress can be for example: finding the test object, performing the action, or synchronizing the step on its end-event. In addition, script action details can be seen the **TruClient Sidebar Status Pane**.

7. Enhance the script

You can enhance your script in a number of ways such as inserting parameters, transaction, loops, and verification steps.

8. Stop developing

Click the Save button  to save the script. Close the browser window. Restore the VuGen window.

9. Replay the script in Load Mode

Before you run the script in a Controller scenario, run the script in VuGen's load mode. From the VuGen main window, click the **Replay** button  to replay the script in Load Mode. Progress can be monitored in the Replay log. The browser does not open, and snapshots are not displayed.

10. General Tips Regarding Successful Interactive Replay

Do not resize the browser between record and replay and during replay. This can cause objects to move and interfere with TruClient's ability to locate them.

Do not switch between applications during interactive replay. Keep the browser in focus.

Note: This is especially important when the Related Objects feature is used, as resizing may change the relative position of the objects.

Note: Any customizations (such as bookmarks) that you make within this instance of Firefox will not be saved globally. This is because VuGen opens each script in a unique Firefox profile. If you want to use Firefox for any use other than creating this script (e.g. browsing the internet), we recommend that you open an additional Firefox window.

Actions



User interface elements are described below:

UI Element	Description
	Record. Starts recording the script. Additionally, you can use the arrow to specify whether to record before or after the selected step.
	Play. Replays the script. Additionally, you can use the arrow to specify whether to play the selected step only, or to run the script step by step. Running the script step by step pauses the replay after each step. For more information, see "How to Debug Ajax TruClient Scripts" on page 490.
	Stop. Stops recording or replaying the script.
	Toggle Breakpoint. Toggles breakpoints on the selected step.
	Script Level. Modifies the script levels that are visible and replayed in the script. For more information, see "Ajax TruClient Script Levels" on page 488.
	Start/End Transaction. Inserts a starting or ending point for a transaction.
	Transaction Editor. Opens the Transaction Editor, allowing you to define new transactions and modify existing ones.
	Save. Saves the script.
	General Settings. Opens the Ajax TruClient General Settings dialog box. For details, see "Ajax TruClient General Settings Dialog Box" on page 472.

Functions



To access	Select Functions from the drop down list on the TruClient Sidebar
Relevant tasks	"How To Create and Use Function and Libraries" on page 512

User interface elements are described below:

UI Element	Description
	Enables you to select the active library.
	New Library. Enables you to create a new library.
	Import Library. Import a function library from an xml file.

Ajax TruClient Edit Tab

This tab enables you to cut, copy, and paste steps and data in Ajax TruClient scripts.



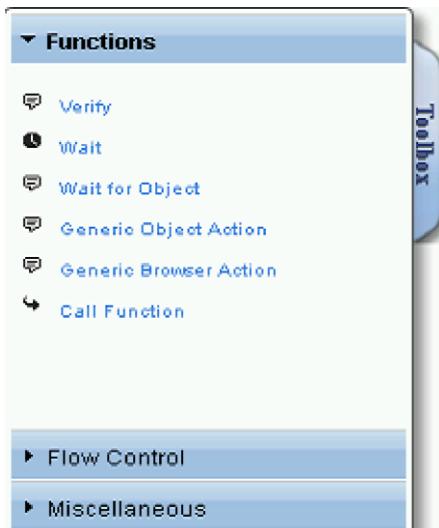
To access	Select the Edit tab from the TruClient sidebar .
-----------	---

User interface elements are described below:

UI Element	Description
	Cut the selected data or step.
	Copy the selected step or data.
	Pastes before the selected step.
	Pastes after the selected step.
	Pastes into the selected step.
	Deletes the selected step
	Opens the Find dialog box, allowing you to search the script for steps by step name or number. Note: You can create a step that is a group of sub-steps. If you are searching for step that is a sub-step, you need to specify both the step group number and sub-step number. For example, enter 4.1 to search for the first sub-step in the fourth (group) step .
	Go to the specified step.

Ajax TruClient Toolbox

The toolbox enables you to add steps to Ajax TruClient scripts. The toolbox opens and closes by clicking on the tab and moves by dragging it up or down.



User interface elements are described below:

UI Element	Description
Functions	<ul style="list-style-type: none">Verify. Verify that an object exists in the application.Wait. Wait for a specified number of seconds before continuing with the next step.Wait for Object. Wait for an object to load before continuing with the next step.Generic Object/Browser Action. Blank steps that can be inserted and manually configured.Call Function. Insert a custom function in the script. For details, see "How To Create and Use Function and Libraries" on page 512.

Flow Control	<ul style="list-style-type: none"> • For Loop. A logical structure that repeats the steps contained in the loop a specified number of times. • If Block. A logical structure that runs the steps contained in the block if the condition is met. • Add else. Click the Add else link to add an else section to your If block. If the condition is not met, the steps included in the else section run. • Remove else. Removes the else section from the If block. <p>Note: If the else section contains steps and you click Remove else, the steps are deleted. Copy and paste them into the main body of your script to save them.</p> <ul style="list-style-type: none"> • If exists. A combination of “If Block” and “Verify”, a logical structure that runs the steps contained in the block if the condition on a property of the selected object is met. • If verify. A logical structure that runs the steps contained in the block if the selected object exists in the application. • Break. Causes the loop to end immediately without completing the current or remaining iterations. • Continue.. Causes the current loop iteration to end immediately. The script continues with the next iteration. • Catch Error. Catches an error in the step immediately preceding and runs the contents of the catch error step. For more information, see " How to Enhance Ajax TruClient Scripts" on page 505. • Exit. Exits the iteration or the entire script depending on the specified setting.
Miscellaneous	<ul style="list-style-type: none"> • Evaluate JavaScript. Runs the JavaScript code contained in the step. • Evaluate JS on Object. Runs the JavaScript code contained in the step after the specified object is loaded in the application. • Evaluate C. Runs the C code contained in the step. • Comment. A blank step which allows you to write comments in your script.

How to Debug Ajax TruClient Scripts

This task describes different options to debug an Ajax TruClient Script.

View Replay Errors in the TruClient sidebar

If any steps failed during replay, they are marked with an error  icon. Hover the mouse over these icons to view descriptions of the errors.

Run the Script Step by Step

You can run your script step by step to view the replay more slowly and in a controlled manner. To run the script step by step, select the down arrow from the replay button in the browser and select **Replay step by step**. Repeat this procedure after each step to continue the step by step replay.

View the Replay Logs

In the VuGen Output Pane, you can view details of your script's replay. Select **Output Pane > Replay** or the **Output Pane > Browser Replay** logs.

Insert Toggle Breakpoints

Breakpoints instruct the script to stop running during a replay when in interactive mode. They can be used to help debug your script. To insert a toggle breakpoint, select the desired step and click the Breakpoints  button.

Debug Scripts Using Snapshots

You can use the snapshots generated during replay to debug scripts by viewing the snapshots of the failed step(s).

1. Select the  button from the **TruClient Sidebar** and select the **Interactive Options** tab. Set the **Replay Snapshots Generation** to **On Error**.
2. Replay the script from the **TruClient Sidebar**.
3. Look in the **Output Pane > Replay** or the **Output Pane > Browser Replay** logs for errors. Note the step numbers of the steps that had errors.
4. To view the snapshots from the **TruClient Sidebar**, select a step with an error, and select the

- Button.

(Or)

To view the snapshots from VuGen, select **View > Snapshot Pane**.

You now have a group of snapshots in which errors occurred in the script.

Modify and view script levels

Sometimes, steps that were recorded and are necessary for replay are placed in levels 2 and 3. In this case, you need to manually modify the level of those steps to level 1.

To modify the script's replay level, drag the slider in the toolbar to the desired level. Dragging the slider to level 3 displays and replays the steps on levels 1, 2, and 3.

To move a step to a different level, open the step and click on the step section. Move the slider to the desired level. If the step is part of a group step, both the group step and the individual step must be modified.

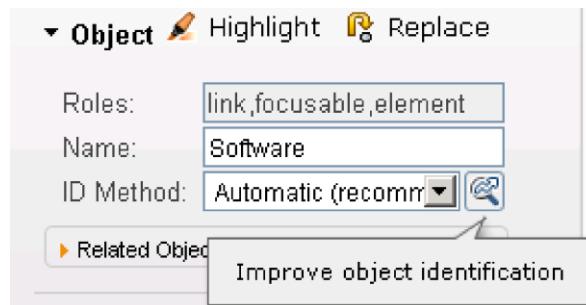
Insert Wait steps

Sometimes a script will fail to replay because an object in a step is not available when the previous step has finished. You can resolve this by inserting **Wait Steps** into your script which pause the script replay before continuing to the next step. There are two different types of **Wait Steps**:

The **Wait** step pauses the script for a defined amount of time before continuing to the next step. The **Wait for Object** step pauses the script until a specified object is loaded before continuing to the next step.

Improve Object Identification

If the Highlight option fails, use the Improve Object Identification



Replacing an object

If you selected the wrong object during recording, or an object has permanently changed you can replace it with a different object without replacing the step. This effectively resets the step, deleting changes made to the original step such as relations. Expand the step, select **Object**, and click the

Replace button . Select the new object and replay the script. Replace Object will tell Ajax TruClient that the object currently referenced in the step is incorrect.

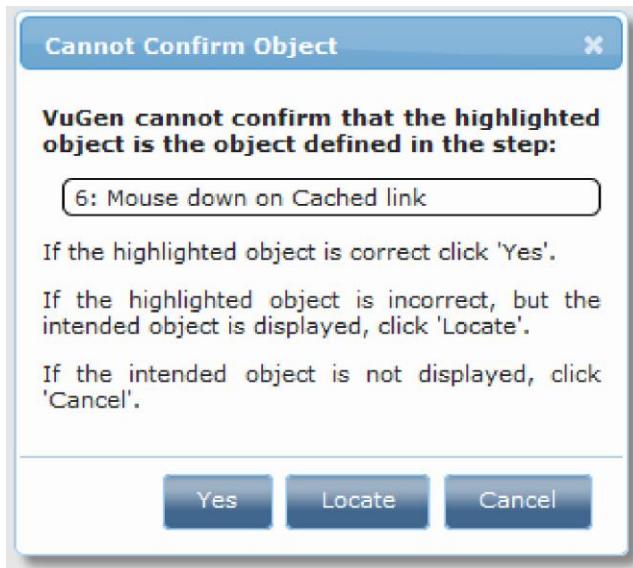
Ajax TruClient will remove any current knowledge of the object and learn the object you select from scratch.

Therefore, you should only use the Replace Object option if the object you used during recording was the wrong one.

Cannot confirm object

Ajax TruClient suspects a specific object to be the desired object but it cannot be positively identified. The suspected object is highlighted on the screen, and the following assistant dialog box

Appears:



Options:

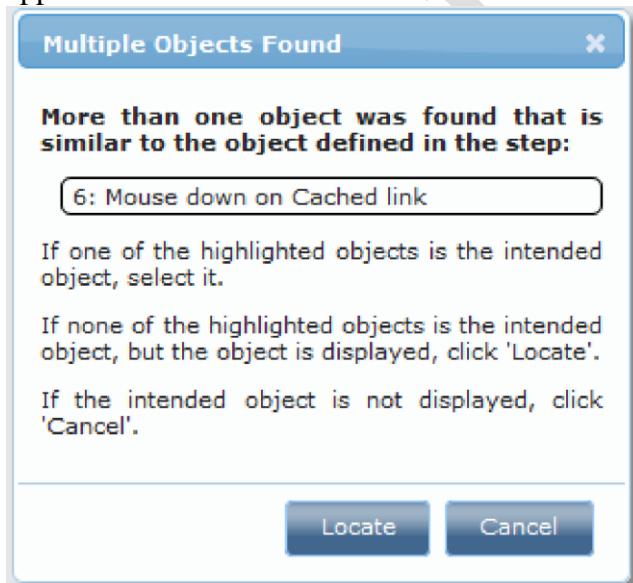
Yes. The suspected (and highlighted) object is the correct object.

Locate. The suspected object is not the right object. You will need to identify the correct object in the application.

Cancel. Stop the replay.

Multiple objects found

Ajax TruClient found several objects that match the identification of the desired object. All suspected objects will be highlighted on the screen, and the following assistant dialog box will appear:



Options:

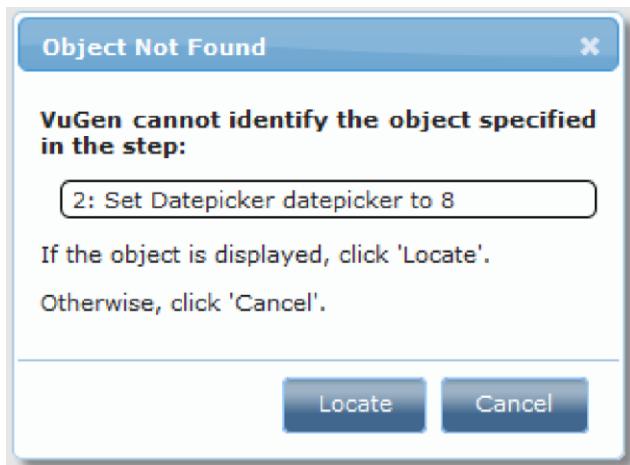
The correct object is one of the marked objects. Click the object in the application to specify the correct one.

Locate. The suspected object is not one of the highlighted objects. You will then need to highlight the correct object in the application.

Cancel. Stop the replay.

Object not found

Ajax TruClient could not find the desired object. The following assistant dialog box will appear:



Options:

Locate. Locate the object. You will need to highlight the correct object in the application.

Cancel. Stop the replay.

How to Select a Random Option from a List

Set the Ordinal argument to 0. TruClient will automatically select a random option from the list. For example, let's assume you have an auto-complete list that shows a list of cities based on the typed text. You've currently selected the second option and the step is: Select option #2 from City autocomplete.

Ajax TruClient Functions and Function Libraries

A TruClient function is a group of steps, such as a login, that you define as a function. Functions are stored in libraries that can be reused multiple times in a script. Each library can contain multiple functions.

Each library can be either local or global. A local library can be accessed by the script that created it. A global library can be accessed by all Ajax TruClient scripts. Additionally, a global library can be saved on the network and shared between many users.

How to Use External Functions in the Script

Add your JavaScript and C functions to the JS-Function.js and C-functions. c files that are part of the script and appear in the left navigation pane in VuGen.

JavaScript functions can be called directly from the Ajax TruClient script, as all arguments and parameters support JavaScript. You can also add an Evaluate JavaScript or LR.evalC step from the Toolbox for this purpose.

To call C functions add an Evaluate C step from the Toolbox.

Alternative Steps

Alternative steps allow you to view instances in which there are multiple ways to perform the same action in a step. If Improve Object Identification fails, try using one of the alternative steps.

For example, you may be clicking on an option in a drop down list in which the text changes based on some value.

Modify the Object Identification Method

You can modify the way Ajax TruClient identifies the object by modifying the object identification method in the Object section of the step properties. The following options are available:

Automatic. Ajax TruClient's default object identification method. The Automatic method allows Ajax TruClient to use its internal advanced algorithms to locate the object. If this method does not successfully find the object during replay, click the Improve Object Identification button and replay the script again.

XPath. If Automatic identification fails, even after using Improve Identification or Related Objects (described below), try using the XPath identification method. This method identifies the object based on an XPath expression that defines the object in the DOM tree.

Click the dropdown arrow next to the **XPath** edit box to select a suggested XPath for the object. You can manually modify the suggested path. To revert to one of the original expressions generated by Ajax TruClient, select one of the options from the drop-down again.

For example, if you need to select the first search result, regardless of the term being searched for, using XPath identification may help.

JavaScript. JavaScript code that returns an object. For example: document.getElementById("SearchButton") returns an element that has a DOM ID attribute of "SearchButton".

Using the JavaScript identification method you can write JavaScript code that references the returned document and can use CSS selectors and other standard functions.

For example, the page returned by the server contains multiple links with the same "title" attribute (search results) and we want the script to randomly click on one of the available links.

Object identification for this case, using the JavaScript identification method, may look something like this:

```
var my_results = document.querySelectorAll('a[title="SearchResult"]');
random(my_results);
```

AWR REPORTS

Why we are using AWR report?

A: To analyze the database utilization.

Hard parsing:

In hard parsing, it will follow below steps.

- DB (Database) will fetch sql query from hard disk.
- It will check for syntax and also for Symantec
- It will parse the sql query.
- It will keep in the shared pool memory and fulfil the request.

Soft parsing:

- It will simply fetch passed sql query from shared pool memory and it will fulfil the request.
- If percentage is 91% then it is good
- If percentage 100% then we need to increase shared pool memory size.

How to generate a report?

Sol:

- Select any monitor which we added.
- Right click on monitor.
- Select "reports" and select "Quick" It will display quick report pop up window.
- Select "thresholds" and select "options" according to our requirement.
- Select "general" ->Graph.
- Navigate to "filter" and schedule settings.
- Select the report period.
- Select the report type (html/text/xml).
- Click on generate report file.

AWR report (Automatic Workload Repository):

- To communicate with remote server we need to install putty.
- In putty we need to provide remote server host name (or) IP address and we need to provide remote server credentials.
- We have to enter a command {sqlplus"/as sysdba";}
- Click on enter which will prompt to sql prompt.
- In sql prompt we need to enter below one:
SQL>@\$ORACLE_HOME/rdbms/admin/awrrpt.
- Click enter which will ask for "Report_type".

- Provide report type as html.
- Click enters which will ask for number of days.
- Provide number of day → Click enter.

It will show "snapshot ID's".

And ask for begin snap_ID → Provide begin snap id and click enter.

And Ask for end "snap_ID".

Provide end snap_ID and click enter → It asks for report name.

Provide the report name with extensions (.htm1) → click enter → exit.

