Table of Contents

[Executive Summary 2](#_Toc100479634)

[Requirements Gatherings 3](#_Toc100479635)

[Understanding Your Application 3](#_Toc100479636)

[Understanding Your Environment 4](#_Toc100479637)

[The WebLOAD Tool 4](#_Toc100479638)

[Understanding the WebLOAD Architecture 4](#_Toc100479639)

[Understanding the USF WebLOAD Architecture 5](#_Toc100479640)

[How to Create your First Tests 6](#_Toc100479641)

[WebLOAD Recorder 6](#_Toc100479642)

[WebLOAD Console 9](#_Toc100479643)

[WebLOAD Analytics and the WebLOAD Dashboard 9](#_Toc100479644)

[WebLOAD Analytics 9](#_Toc100479645)

[Webload Dashboard 10](#_Toc100479646)

[Performance Metrics 13](#_Toc100479647)

[Advanced Topics 14](#_Toc100479648)

[Parameterizing Scripts and using Test Data 14](#_Toc100479649)

[Parameterizing Scripts 14](#_Toc100479650)

[Test Data 16](#_Toc100479651)

[PMM 17](#_Toc100479652)

[Adding Tomcat jmx 17](#_Toc100479653)

[Adding JDBC 19](#_Toc100479654)

[Game Days 22](#_Toc100479655)

[Appendix 1 – Glossary of Terms 23](#_Toc100479656)

[Appendix 2 – Challenges and Risks 24](#_Toc100479657)

[Appendix 4 – Contacts 25](#_Toc100479658)

[Appendix 4 – Helpful Resources 26](#_Toc100479659)

# Executive Summary

This document comes handy for understanding where to start with load testing, how to setup a load testing environment for an application and how to execute the tests and analyze the results.

It will cover what you will need to know about the application you intend to test so that you are prepared to create test cases.

It will introduce the WebLOAD tool that the Student Information Services team uses to test the Banner system. You will learn generally how the software works, where it is installed, and how to access it.

Finally, it will walk you through the general process of recording tests, executing them, and running them at scale to understand how your application performs and how you might improve that performance.

It is intended to be an introduction to these concepts so you can get started. It will include links and references to more in-depth documentation provided by the vendor that will be useful once you are familiar with the basics.

Throughout the document, we will also be using “Registration Process” and “Student Schedule Search (SSS)” as our Use case. Please refer to [Registration Process recording](https://usf.box.com/s/j3359ub5zbp2jzwkwt4uuy01aaj7b0ym) to understand the process of registrations.

# Requirements Gatherings

In order to use WebLOAD to test your application, you will need to be prepared with some specific information about it, be familiar with the process you intend to test, and understand the test cases you intend to create. You also need to have that application deployed to the environment where you plan to test it.

## Understanding Your Application

1. **Get the URL of your application**

*Example 1. – Student Schedule Search*

Student Schedule Search is load tested in PPRD environment at: <https://oasisnpa.it.usf.edu/pls/pprd/bwckschd.p_disp_dyn_sched>

Choosing a term and hitting submit is considered one transaction.

Filling out the form and hitting class search is a second transaction.

To perform load testing, we need to record a script using application URL of the environment in which load testing needs to be performed.

1. **Understand the Process:**

A clear understanding of process with details like choosing options and navigating through pages helps in creating good real time scenarios. We need the knowledge of the process and an understanding on what set of actions can be considered as a transaction. This helps in dividing the whole process into multiple *transactions*.

1. **Prepare detailed scenarios of the functionality**

The execution of the process in real time will always be different from how we navigate through the process while recording the script. Understanding this help create better *Agendas* by including sleep times in the scripts when creating them.

Knowing how the application is used in real life will allow you to create *load profiles* that simulate realistic demand. If you have logs or other metrics that indicate how many visitors use your system and how that demand fluctuates over time, you will be able to simulate that in your WebLOAD *templates*.

*Example 2. – Student Registration*

The registration process URL is: <https://oasisnpa.it.usf.edu/pls/pprd/twbkwbis.P_wwwlogin>

To simulate the situation where many students are already logged in before registering, we include sleep times in the agenda.

To simulate the scenario where many students register during the first minute their registration window opens, we create a load profile that starts high and tapers off.

1. **Prepare test cases for your scenarios**

Once you understand the processes you want to test and the types of load you would like to generate to test them, prepare a list of intended test cases for those scenarios.

Writing them down with a standard naming convention will make it easier to keep them organized. You can use that naming convention to save the agendas, templates, and test *sessions* that WebLOAD creates.

## Understanding Your Environment

*Example 3. – Naming Conventions*

The registration process, had 3 scenarios:

1. Student Schedule Search using default search options named as “SSS\_ALL\_SUB\_TS1”
2. Student schedule search by choosing three different subjects named as “SSS\_CUSTOM\_SUB\_TS2”
3. Registration Process named as “STU\_REG\_TS1”

Scenario sheet: - <https://usf.box.com/s/p8txyndintrk1rwobahz1mm37fxjd88q>

1. **Prepare a load testing environment to match production**

Load testing an application means testing if system can keep up with the traffic. Testing production environment is risky as production database has Realtime data. However, we need to test the system capacity. Hence, we need to create another environment with same system configurations as production.

Decide what environment you will test against and then compare the specs of that environment to your production environment. Make them match as closely as possible and understand the limitations of the tests when you are not able to test the actual system.

1. **Deploy your application to the test environment**

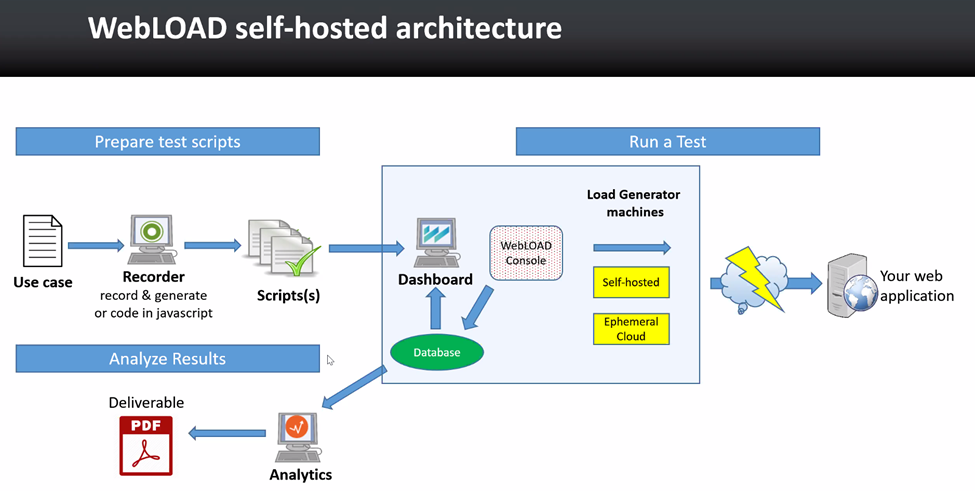
Deploy the latest version of application which is either production ready or which is currently in production, to your test environment.

# The WebLOAD Tool

## Understanding the WebLOAD Architecture

Webload has multiple components including

1. Webload recorder: This is used to record the scripts and edit scripts. Recorder needs to be installed on tester’s desktop. Recorder doesn’t need license. Hence any number of users can install the recorder. Please check webload [documentation](https://usf.box.com/s/7efti0jdpknh6e7gjjondjpdke48ooaj) for more information.
2. Webload Console: A console is used to create templates which determines the load size, load profile etc. Webload console can be installed on a server per license. Test scripts developed in a recorder needs to be uploaded to console server or the console server can hold recorder as well and scripts can be recorded and tested in console server. Please check webload [documentation](https://usf.box.com/s/gxj5pqt685p7jxk8bbsv98vvjmuvpiq7) for more information.
3. Webload analytics: Webload Analytics is used to customize the graphs that from load session and generate the customized reports. Webload analytics is installed on Console server itself. Please check webload [documentation](https://usf.box.com/s/nh0h5b5rna6b6urtv2fnk0ti8b7ppma4) for more information.
4. Webload dashboard: Webload dashboard is used to watch the metrics of load test session live as well as post the completion. It is also used to schedule the load sessions. To use dashboard, it needs to be started first. Then the dashboard is accessible at *localhost:3000* on that server. Please check webload [documentation](https://usf.box.com/s/nh0h5b5rna6b6urtv2fnk0ti8b7ppma4) for more information.
5. Load generators: Load generator is a machine that creates load on our system. There can be n number of servers with webload load generators installed and load generators doesn’t need license.



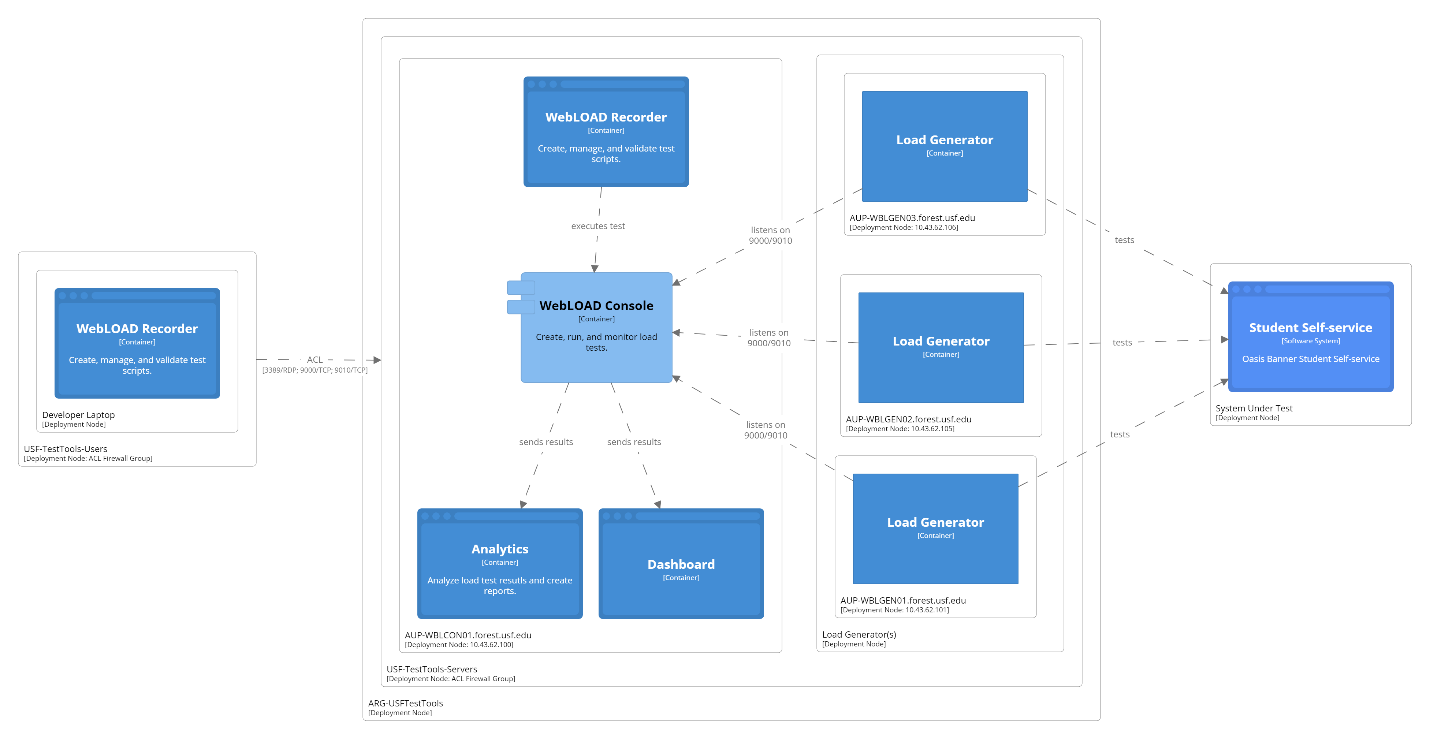
## Understanding the USF WebLOAD Architecture

USF has a license for WebLOAD that is valid until April 2023. It is licensed for:

1. A single WebLOAD Console Server (aup-wblcon01.forest.usf.edu)
   1. The number of testers using this server or systems tested by this server are unlimited
   2. The number of load generators used by the console server is unlimited
2. 5000 Virtual Clients
   1. Up to 5000 Virtual clients can be used at one time
   2. A virtual machine can likely generate a load of 500-700 users at once

The WebLOAD software is installed on a console server with several load generators that are used to create stress on the system under test. These are hosted in Azure.

|  |  |
| --- | --- |
| Server | Description |
| aup-wblcon01 | Console server – Hosts WebLOAD Software |
| aup-wblgen01 | Load Generator – Hosts TestTalk service |
| aup-wblgen02 | Load Generator – Hosts TestTalk service |
| aup-wblgen03 | Load Generator – Hosts TestTalk service |



Developers will need access to the Console server to record agendas and run tests. To get access you will need:

1. Create a ticket with Jeremy Radwan to create an account for you on the aup-wblcon01 server.
2. Create a ticket to be given access to the aup-wblcon01 server from your local machine (RDP access on port 3389).

If a new load generator is required, you will need:

1. Create a ticket for *Jeremy Radwan* to create the server. (Example: <https://usfjira.atlassian.net/browse/HELP-115339> )
2. Then we need/needs to add it/them to the USF-TestTools-Servers Firewall group – (Example: <https://usfjira.atlassian.net/browse/HELP-115512> )

The WebLOAD software is installed on the ‘D:’ drive of the aup-wblcon01 server. However, there are some local settings that are useful. These will be mentioned in the sections below.

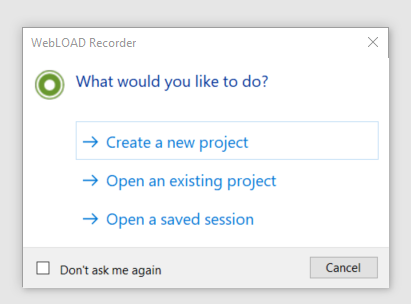
# How to Create your First Tests

After you understand the application you want to test and have access to the WebLOAD software at USF you are ready to start recording your first agendas and running your first tests. You start by recording a test with the WebLOAD Recorder. Then you execute the test with the WebLOAD Console. The results are available in WebLOAD Analytics and the Dashboard. These steps will be briefly described in the following sections.

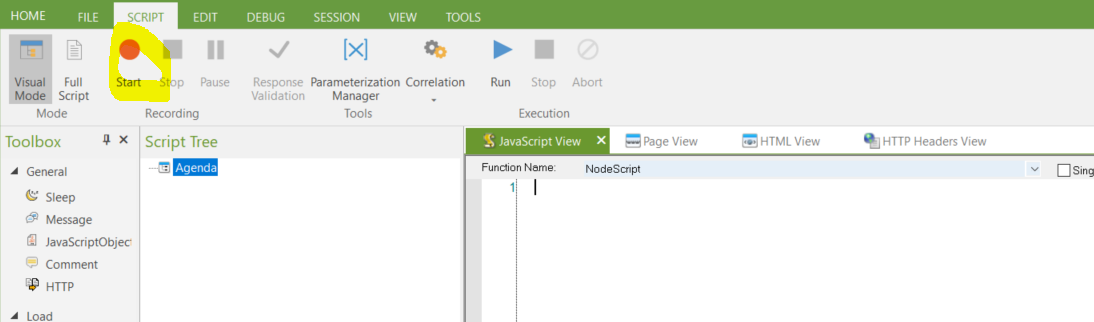
## WebLOAD Recorder

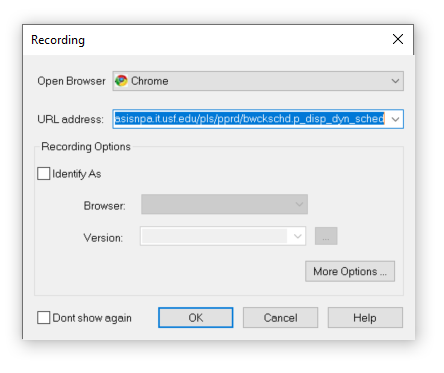
1. **Record the scenario/agenda**

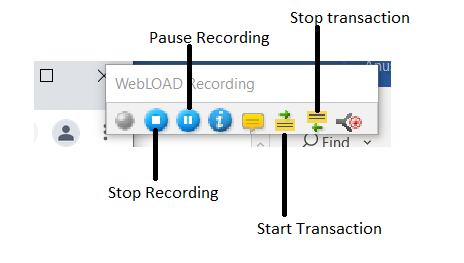
Open WebLOAD recorder and create a new project.



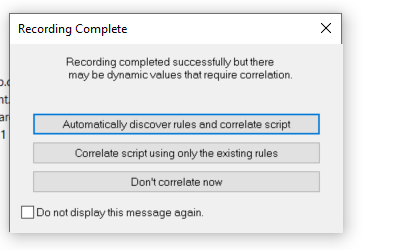
Hit start and give URL and browser. Browser is preferably Chrome.



  
Use “Start transaction” button to start recording a transaction. Use “Stop transaction” to stop recording a transaction. Use “Stop recording” button to end recording the script.

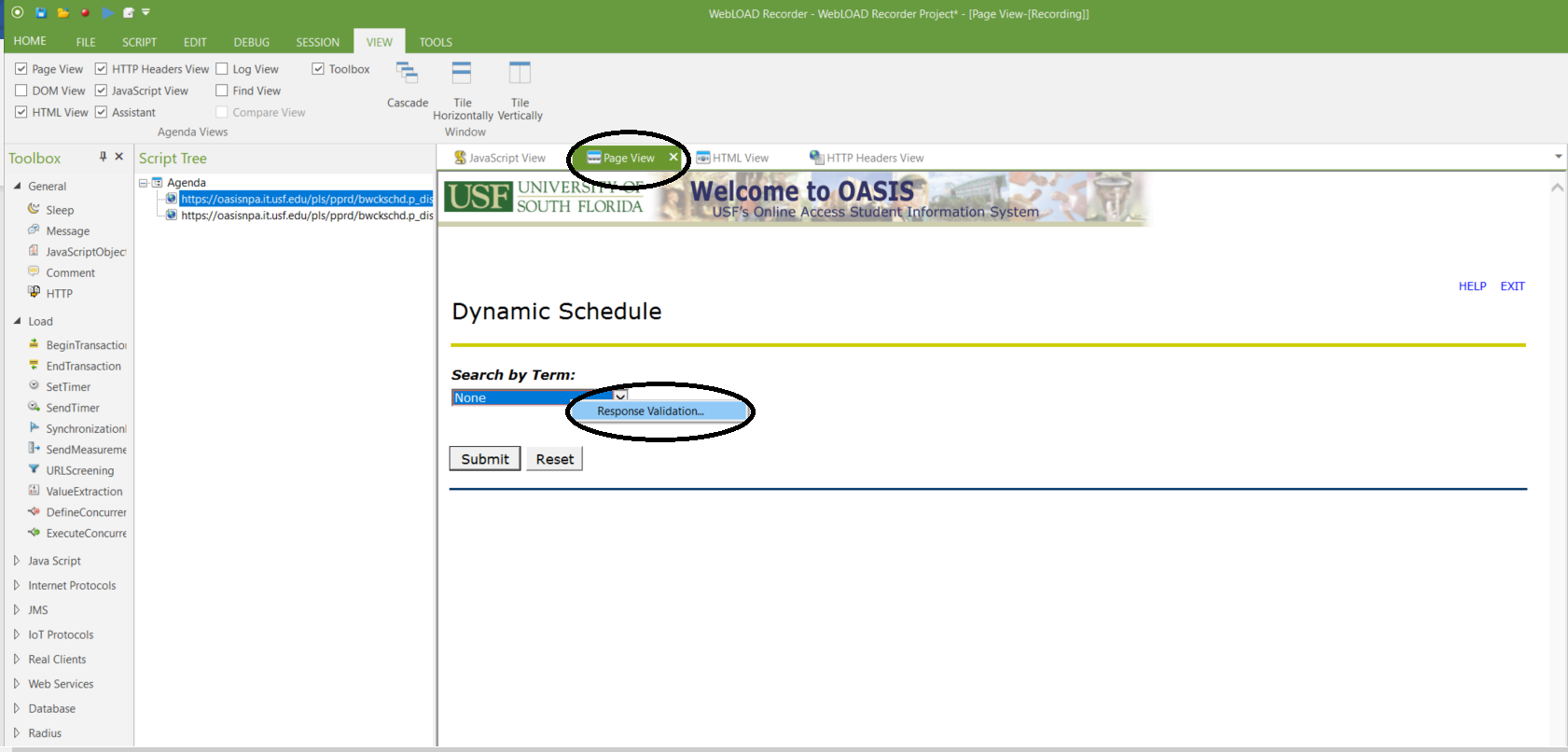


Once the script recording is stopped, we see correlation notification on screen. Click on “Automatically discover rules and correlate script”. For more information on correlation, check Chapter 7 of [WebLOADIDEUserGuidePRO.pdf](https://usf.box.com/s/7efti0jdpknh6e7gjjondjpdke48ooaj)

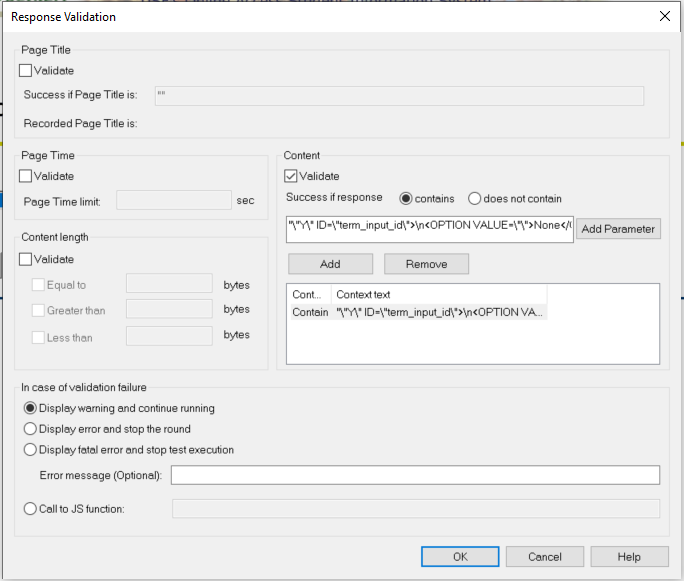


1. **Include validations**

Validations helps us during analysis to understand the exact reason of transaction. To add validation logic, go to “Page view” and right click on the page where validation is required.



Add required validations in this screen. To explore more about validations please check Chapter 8 of [WebLOADIDEUserGuidePRO.pdf](https://usf.box.com/s/7efti0jdpknh6e7gjjondjpdke48ooaj)



## WebLOAD Console

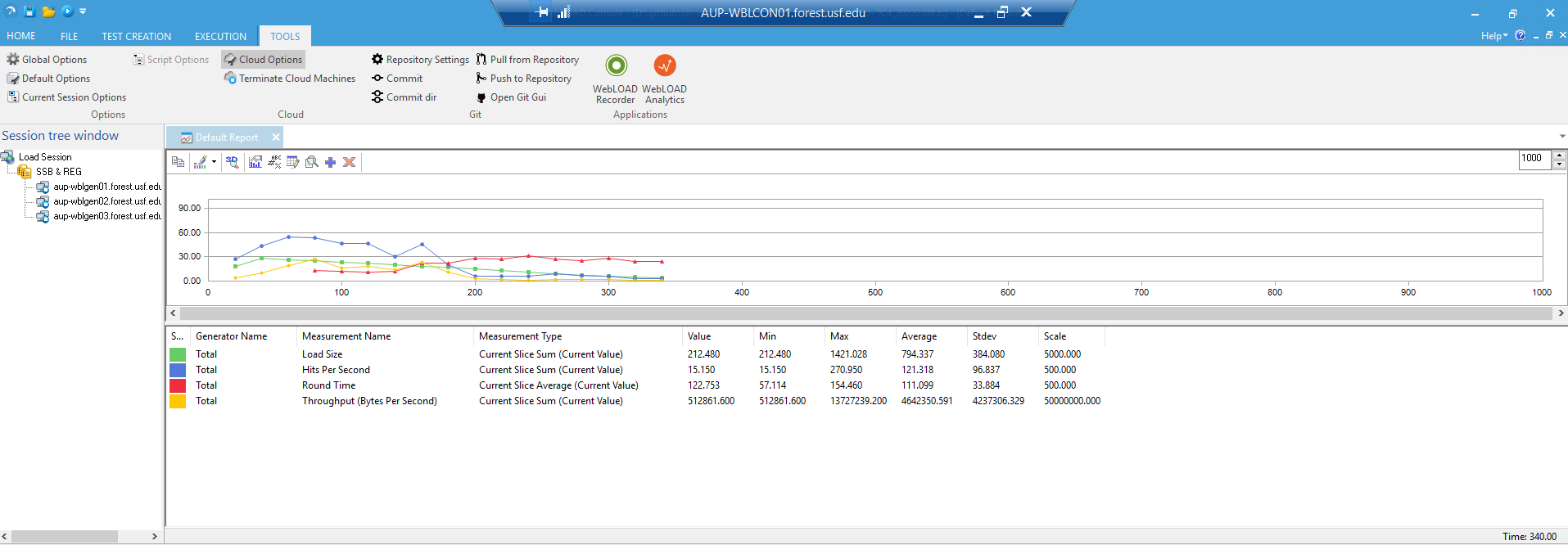
1. Prepare test templates by determining number of virtual users to generate and the load profile
2. Make sure to play with test data and virtual users’ options available both in recorder as well as Console to use test data effectively
3. There is an option to create a mix of scripts in console to test different scenarios at the same time
4. Execute the saved templates on game day
5. Save the session to create reports in future using analytics

## WebLOAD Analytics and the WebLOAD Dashboard

WebLOAD tests store the statistics from their runs in a database. These session statistics can be viewed and analyzed in a number of ways, with Analytics and the Dashboard being the two most prominent.

### WebLOAD Analytics

A WebLOAD Load session file gets created for every load test performed. This Load session has all the data about system performance. We can open analytics directly from console under tools or open analytics as a separate application.



From a saved session it is possible to add or delete the graphs to create a customized report with metrics we are concerned about.

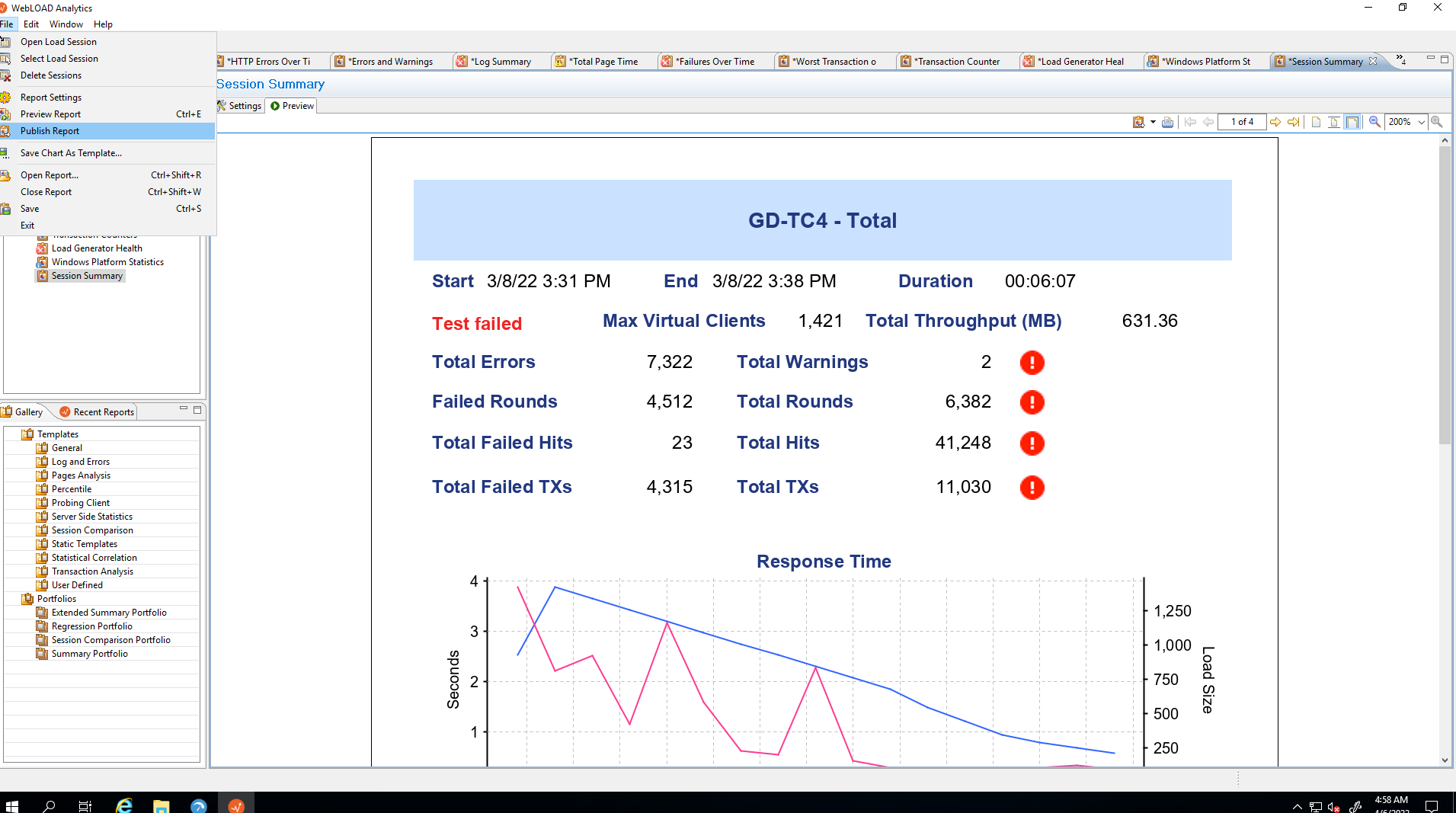
We can use metrics from a saved session and create our own templates changing the parameters. Also, we can compare between sessions. For more information on working with templates and how to do it, refer to [WebLOADAnalyticsUserGuide.pdf](https://usf.box.com/s/nh0h5b5rna6b6urtv2fnk0ti8b7ppma4)

1. **Save and publish the reports to all concerned parties**

WebLOAD Analytics enables you to publish a chart in the following formats:

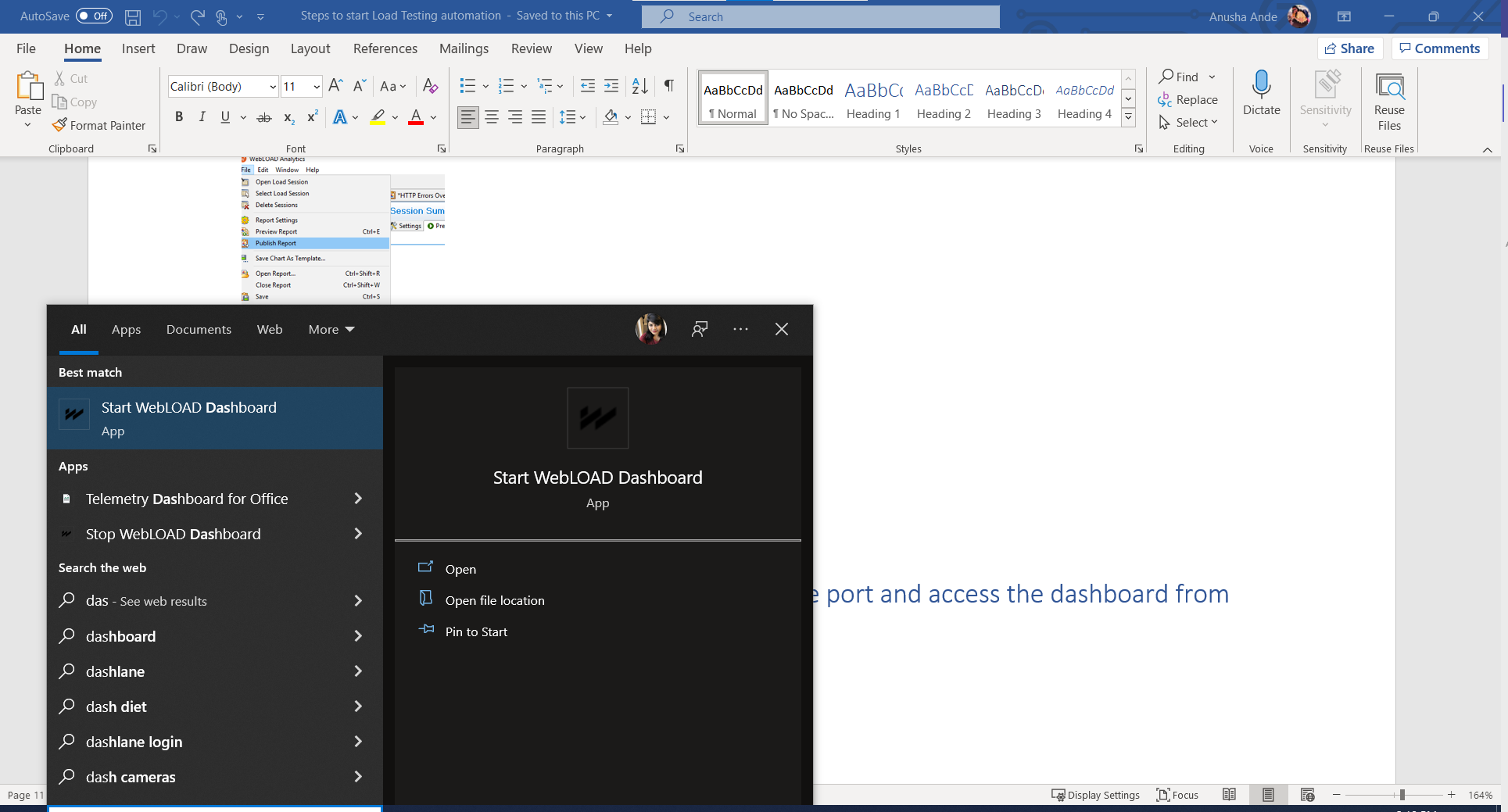
* Doc (Microsoft Word).
* PDF (Adobe Reader).
* XLSX (Excel).

To publish a chart, go to file and click o publish.



### Webload Dashboard

1. **Start the Dashboard**

Search for “start WebLOAD dashboard” in windows search 

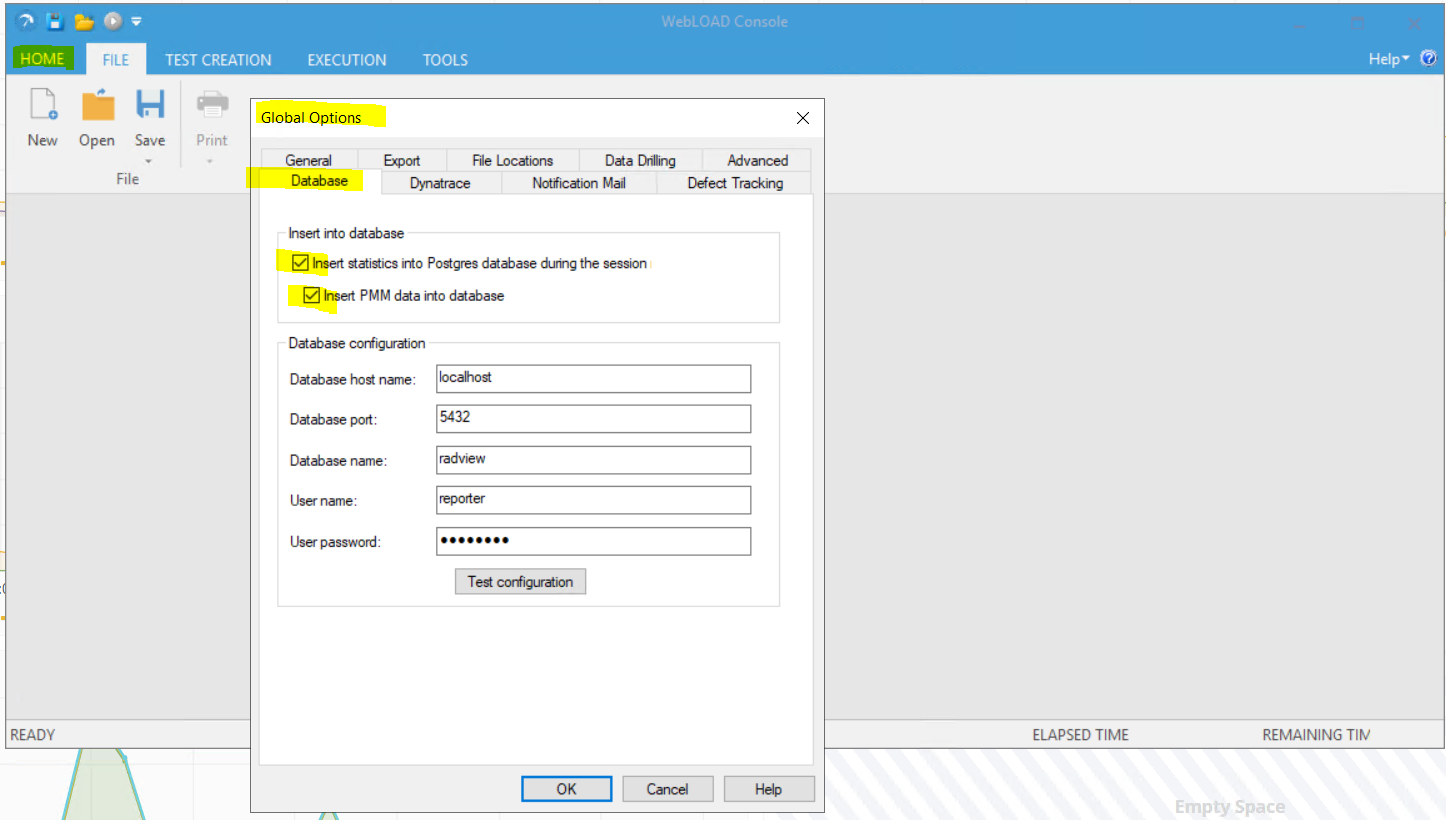
1. **Open dashboard on local host port 3000**

Now we should be able to access webload dashboard at <http://localhost:3000> . Use username as “admin” and password as “admin” to access the dashboard.

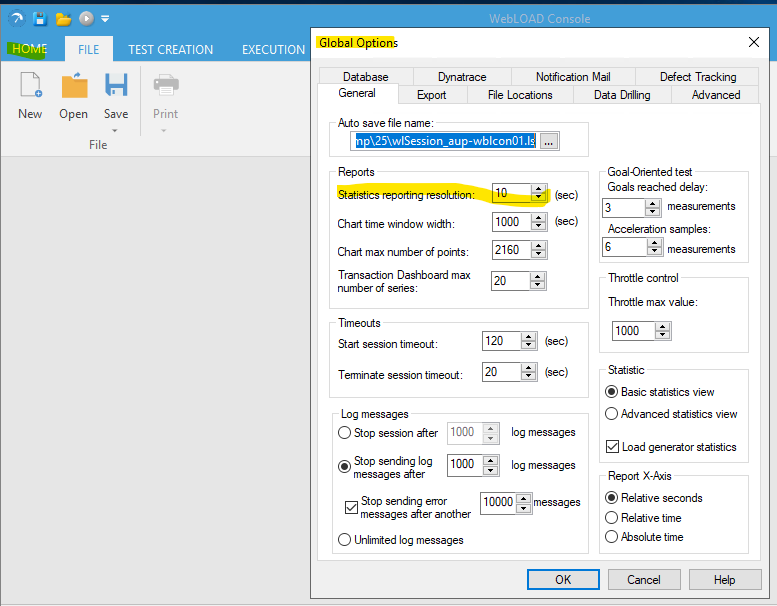
1. **Monitor the results live during test**

There are few settings to do in console server to be able to monitor the test results live in dashboard.

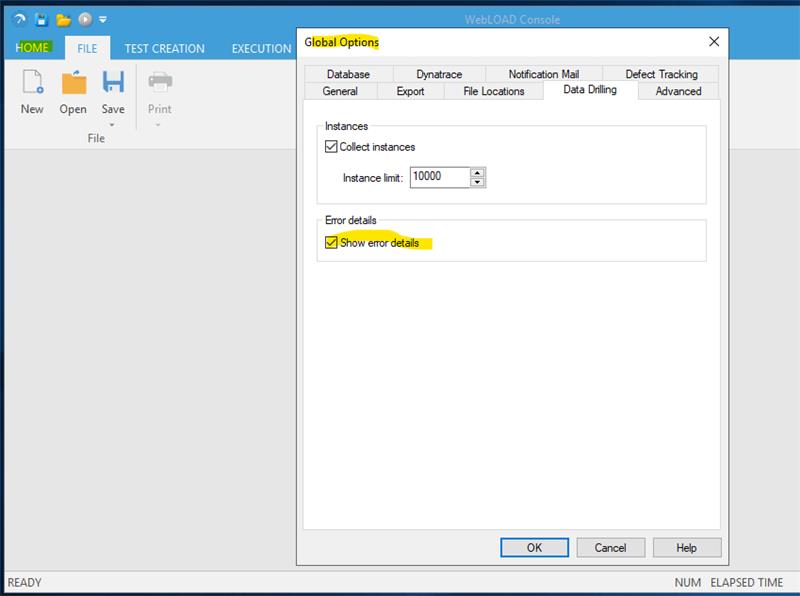
First one is to check “insert into databases” options. Home -> Global options -> Database -> check the two boxes under “insert into databases” as shown below.



Second one, “Statistics reporting resolution” setting is defaulted to 20 in the tool. This helps in collecting more data. Radview recommended this setting to be 10.



Third one is to check the “Show error details” box under data drilling.



1. **Compare with different sessions.**

We can compare data from different sessions by simply choosing the sessions that we want to compare on dashboard



For more information on dashboard read [WebLOADDashboardGuide.pdf](https://usf.box.com/s/ndr9eeqjtmlhlwhwgvpdutnthxnbu4s8)

### Performance Metrics

There are several load testing performance metrics that show how a web application or website is performing. Of those metrics we considered to monitor *Response Time, Page Time and HTTP response status* as we kept seeing some behavioral improvement in the graphs with better resources.

* Response Time

Response time refers to the time that elapses between when a user hits a button on the webpage until the full data received from the server at his end. This is also called End-to-End response time. As a performance tester, you can see response time as a single unit. But the response time figure is the summation of many other small units or individual responses. A performance tester may ignore the component level response time, but for a performance engineer, it is must to understand, analyses and tune the component level response time to optimize the End-to-End response time.

* Page Time

Page Load Time represents the amount of time it takes for a page to show up on your screen. Also, we can say that it is a time when all the components (items) of a page get displayed on the browser for user interaction. It depends on 3 things:*Server Processing Time:* How quickly the servers serve the request and send response back to the client.

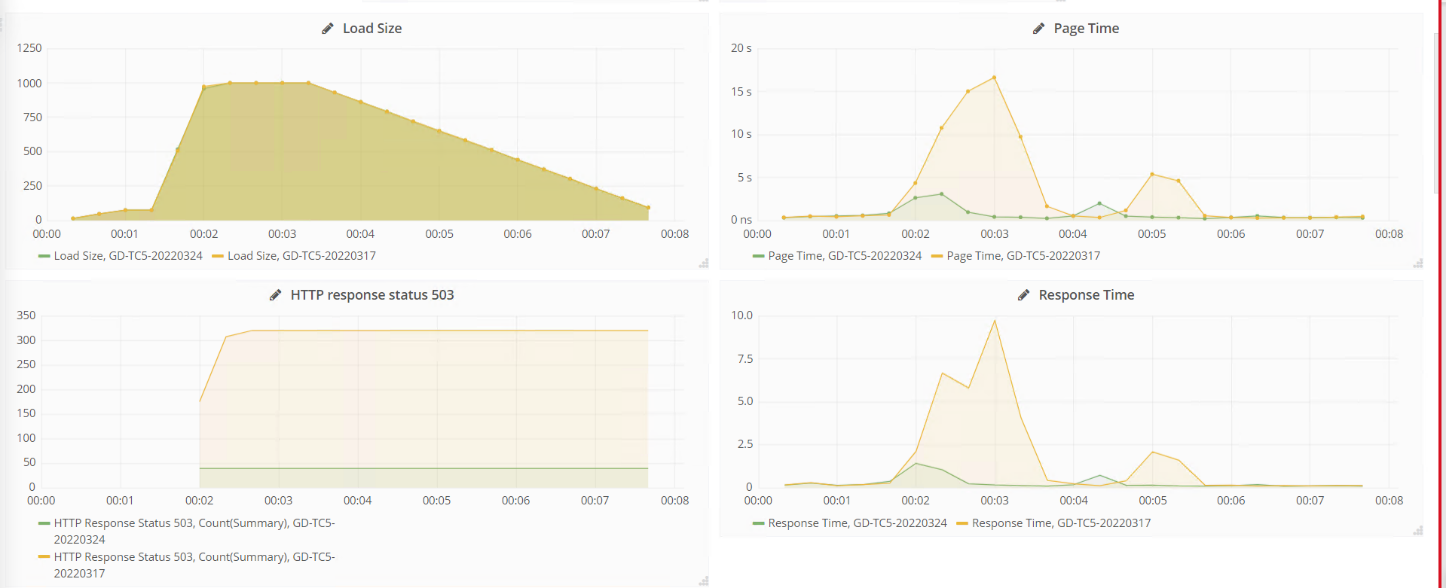
*Network Latency:* How quickly the communication channel transfers the information from client to server and server to client.

*Component Load Time:* How quickly it takes for the browser to parse and execute the document (available in cache or coming from server) and render the page to make it available for user interaction.

* HTTP Response status

A HTTP response is made by a server to a client. The aim of the response is to provide the client with the resource it requested or inform the client that the action it requested has been carried out; or else to inform the client that an error occurred in processing its request.  HTTP response status codes indicate whether a specific HTTP request has been successfully completed. If the HTTP request is successfully completed the response code will be 200. If the client sees an error page with HTTP response code 503, then it indicates that the system failed to respond to the HTTP request because of heavy traffic.

Here is a sample graph from dashboard that shows the difference in all the metrics from first game day to second game day where the database server size is increased from Standard\_D32s\_v3 to *Standard\_D64s\_v3*



Please refer to page 85 Appendix D. Understanding the Templates of [WebLOADAnalyticsUsersGuide.pdf](https://usf.box.com/s/nh0h5b5rna6b6urtv2fnk0ti8b7ppma4) for more information about performance metrics.

# Advanced Topics

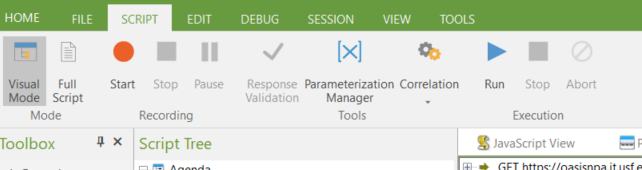
The topics above cover a basic run through load testing with WebLOAD. However, real world scenarios are likely to include the need to parameterize your scripts and to stream test data to it while it runs. You are also likely to want to capture more data than the defaults that WebLOAD provides. These topics are covered in brief below.

## Parameterizing Scripts and using Test Data

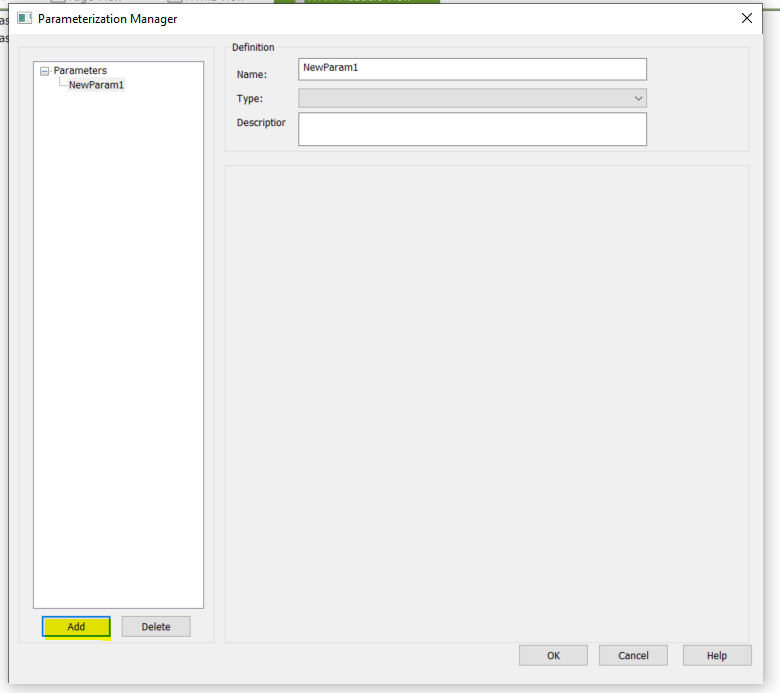
### Parameterizing Scripts

1. **Parameterize the script**

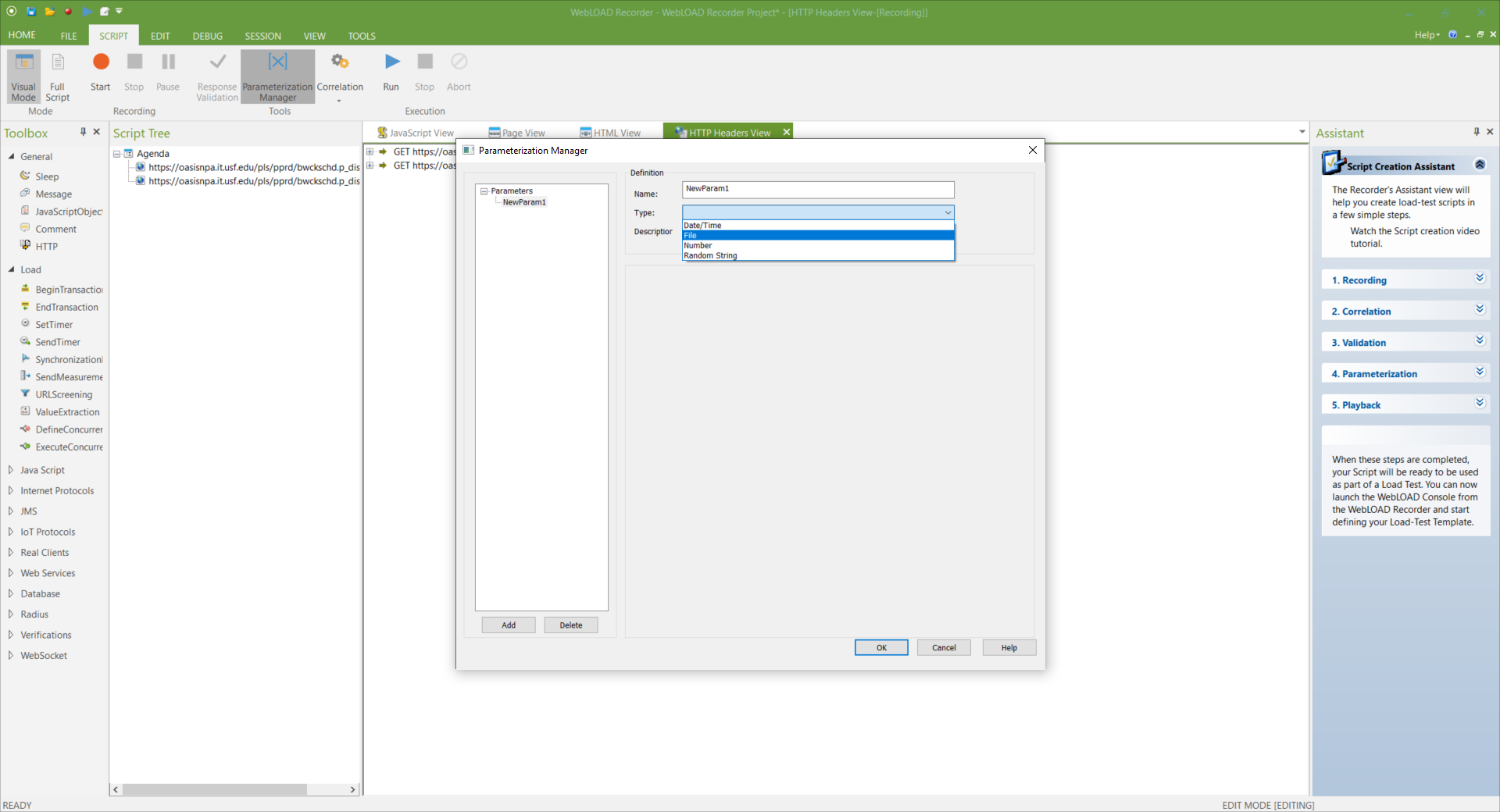
Use “Parameterization Manager” to pass parameter in the script that’s been recorded.



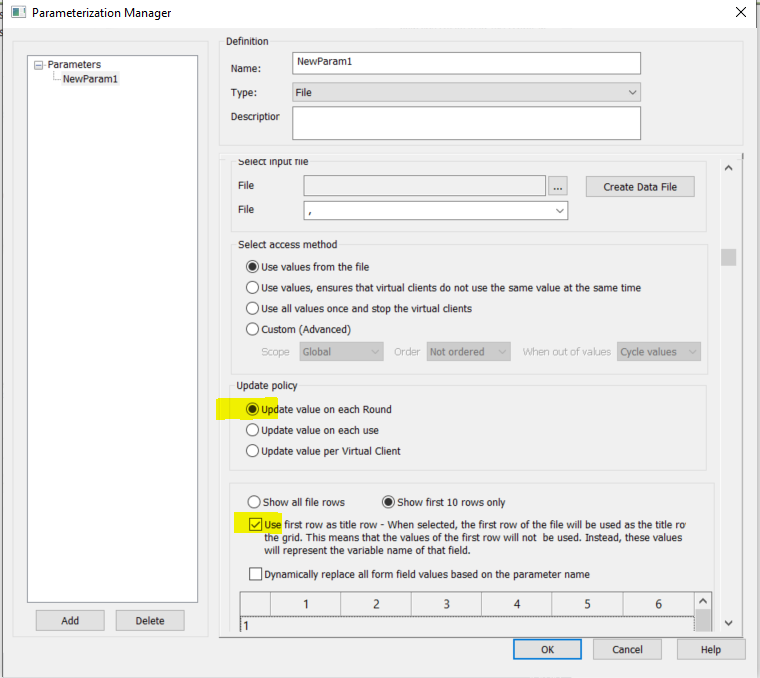
Click on “Add” to add parameters.



There is an option available to pass a parameter file



Make sure to check the box in case the first row is column name. Also, including column names helps in determining the parameter value later in the script. For more information on parameterization, check chapter 9 of [WebLOADIDEUserGuidePRO.pdf](https://usf.box.com/s/7efti0jdpknh6e7gjjondjpdke48ooaj)



### Test Data

Test data file has values of parameters that we pass during the test. For student registration we used a file with username, pin, crn1, crn2, crn3 parameters. so, the test data has IDs of students and CRNs

1. **Make sure the test data has met all the criteria for smooth load testing.**

The test data used for load testing should not stop or fail the transaction. To achieve that we need to find all the possible cases that fails the transactions and make sure to get the test data accordingly.

For this reason, to load test student registration process, we chose IDs of students who are eligible for registration without any holds and CRNs of classes that has no restrictions.

1. **Update contact details.**

If the functionality involves sending emails to users, make sure to update test data with dummy email IDs. If the functionality involves sending text messages, make sure to use a generic or dummy number in that case.

1. **Prepare test scripts.**

It’s advisable to prepare any scripts that makes preparing test data fast and effective.

## PMM

WebLOAD Console provides the PMM for monitoring the performance of various server-side applications, databases, stream technology, system, and Web server measurements in real-time while the test is running. We need to configure Hosts, Data Sources and Selected measurements from each data source on each host. We can maintain multiple configurations of servers, hosts and measurements to monitor. These configurations need to be done on webload server as well as host computers.

### Adding Tomcat jmx

Adding Tomcat *(Data Source)* from 3 application servers *(hosts)* Collecting all the resulted Measurements: -

1. **Tomcat 6.0 Using JMX**

To enable the PMM to monitor Tomcat 6.0 Using JMX, we need to configure the Tomcat 6.0 application server. In the server's execution script (for example, \bin\catalina.bat), add the following rows:

set CATALINA\_OPTS=%CATALINA\_OPTS% -Dcom.sun.management.jmxremote

-Dcom.sun.management.jmxremote.port=<port> -

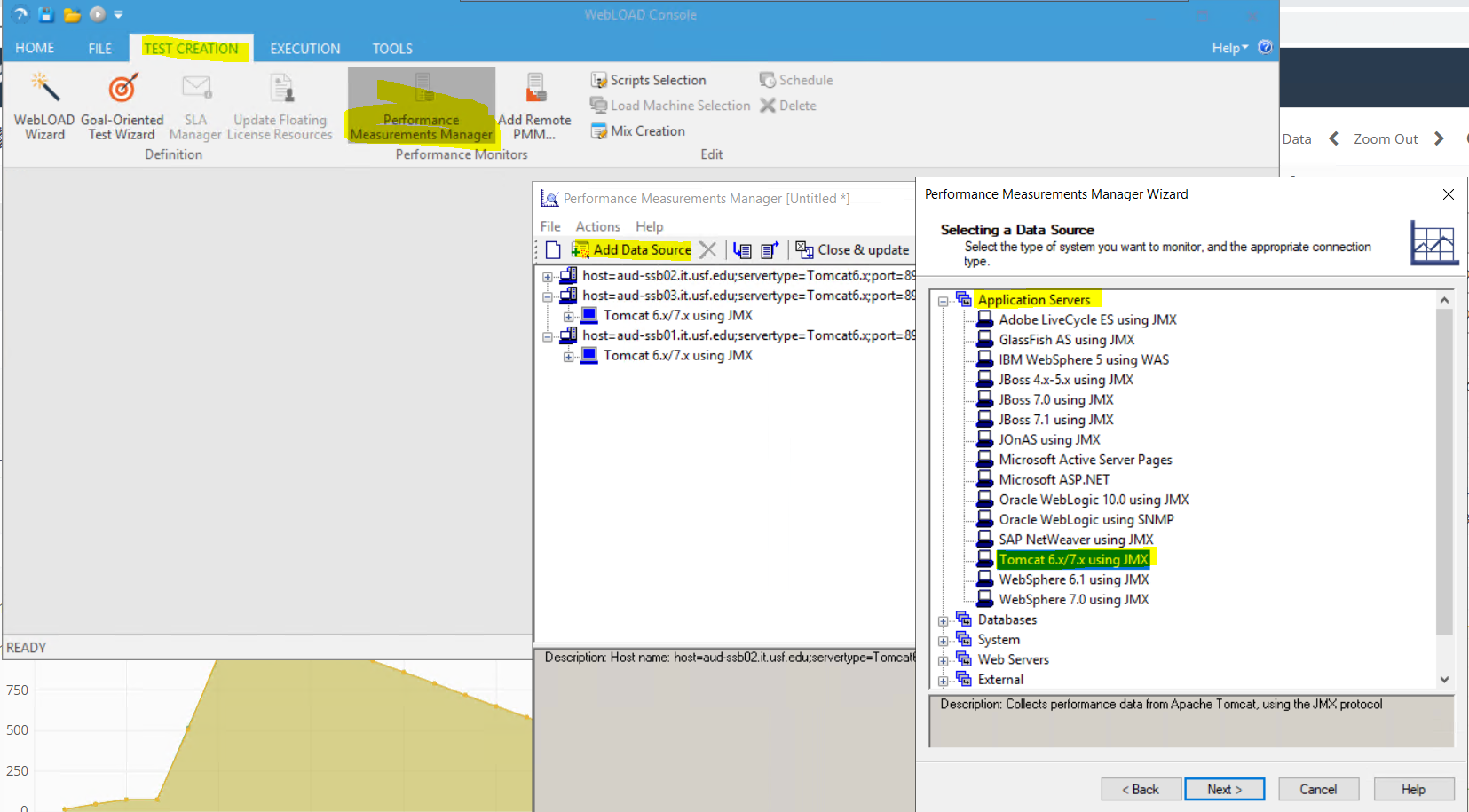
Dcom.sun.management.jmxremote.ssl=false

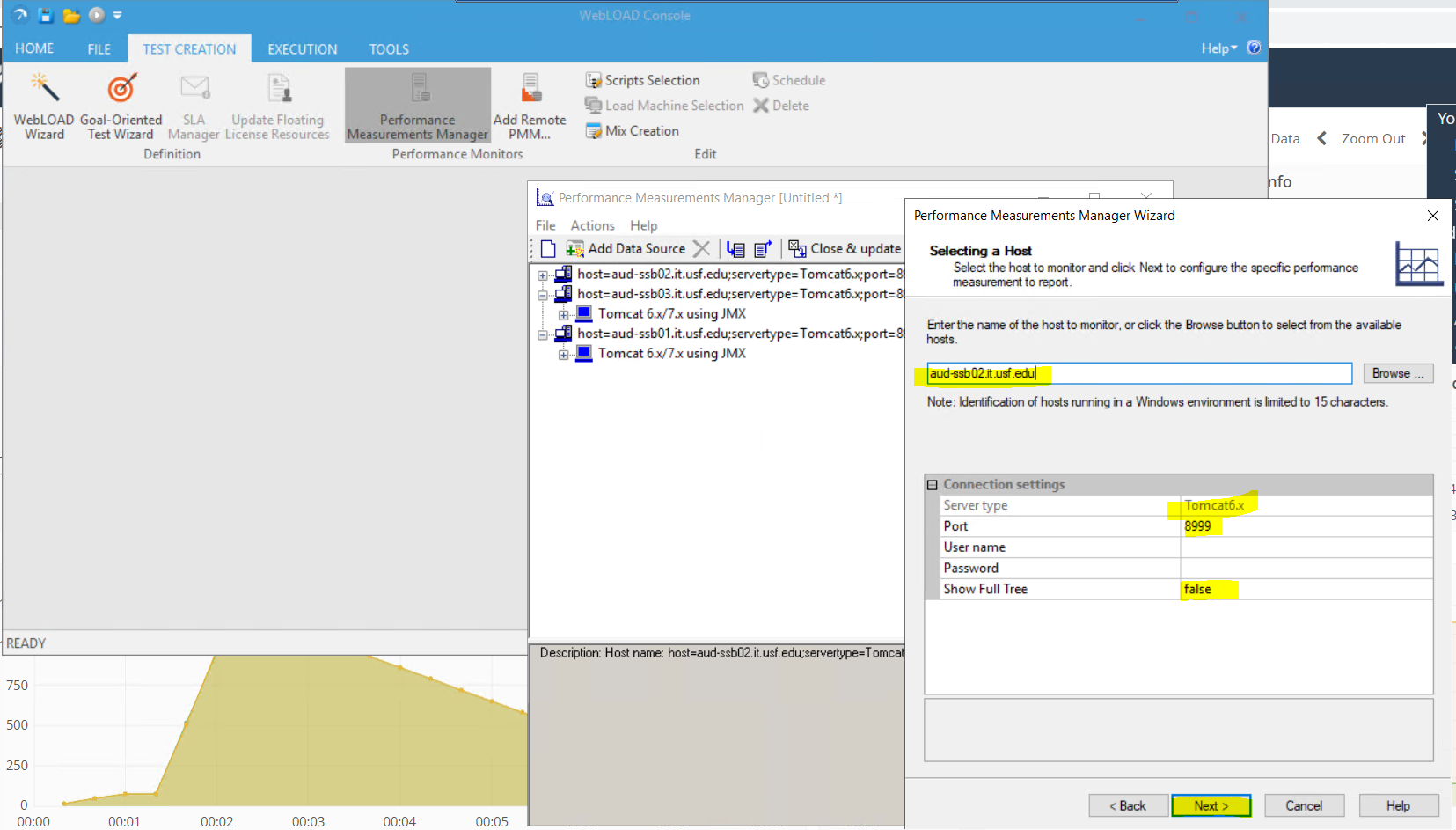
-Dcom.sun.management.jmxremote.authenticate=false -

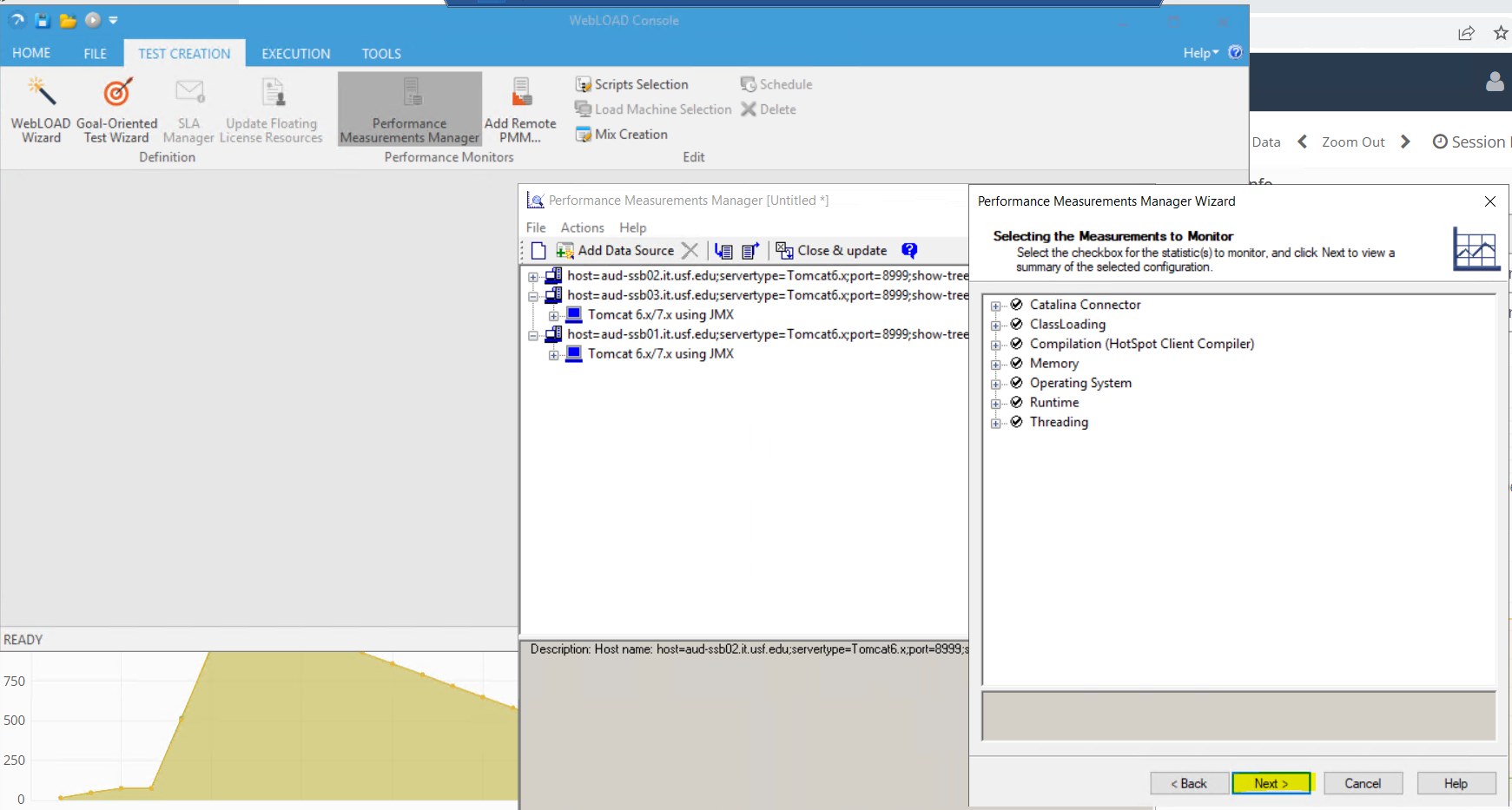
Djava.rmi.server.hostname=<hostname>

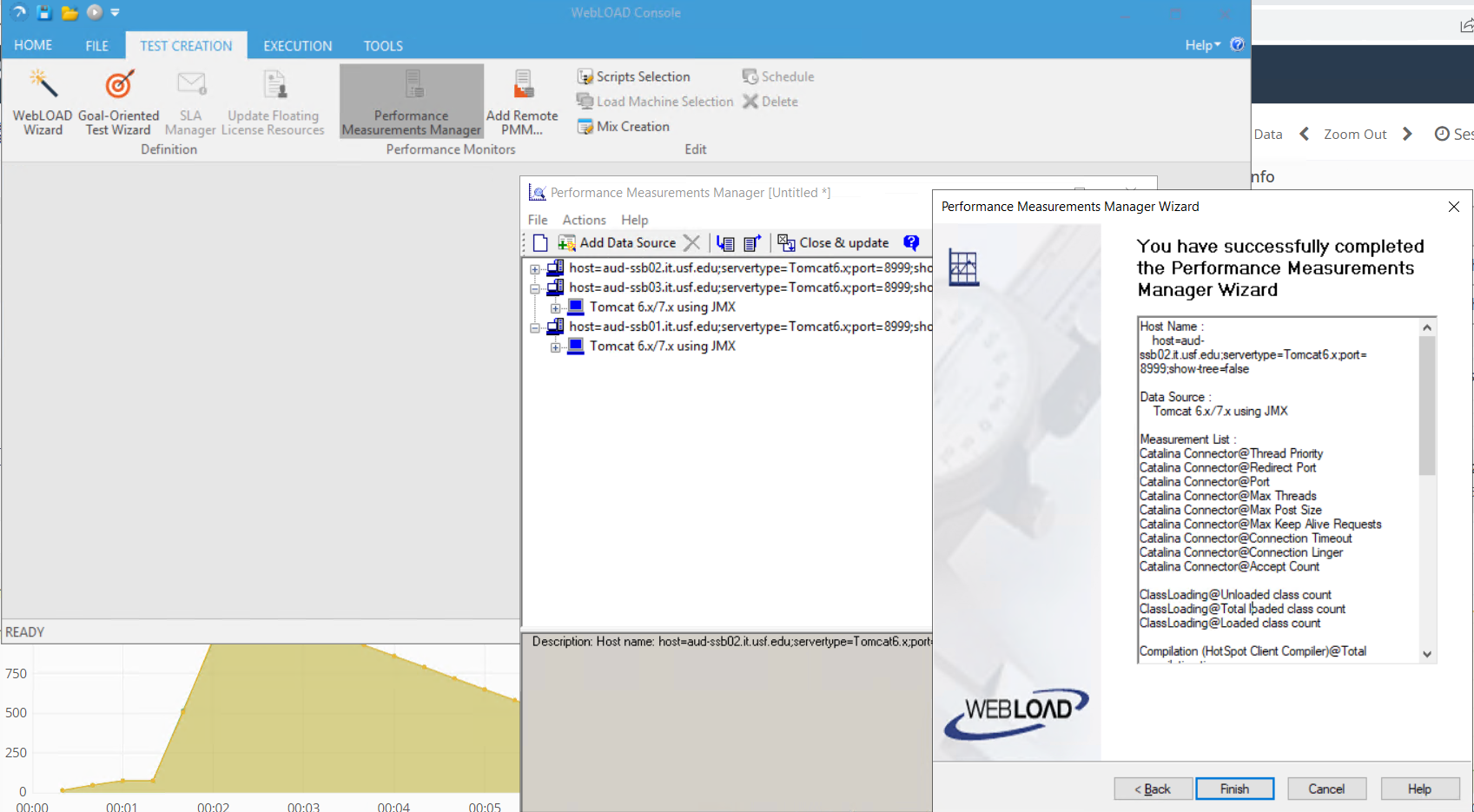
Please refer to page 384 of [Webload Console Users guide](https://usf.box.com/s/gxj5pqt685p7jxk8bbsv98vvjmuvpiq7) for more information.

1. **Configure PMM in Console server and save the .pmm file for future use. Follow the remaining process from screen shots below to finish the setup.**







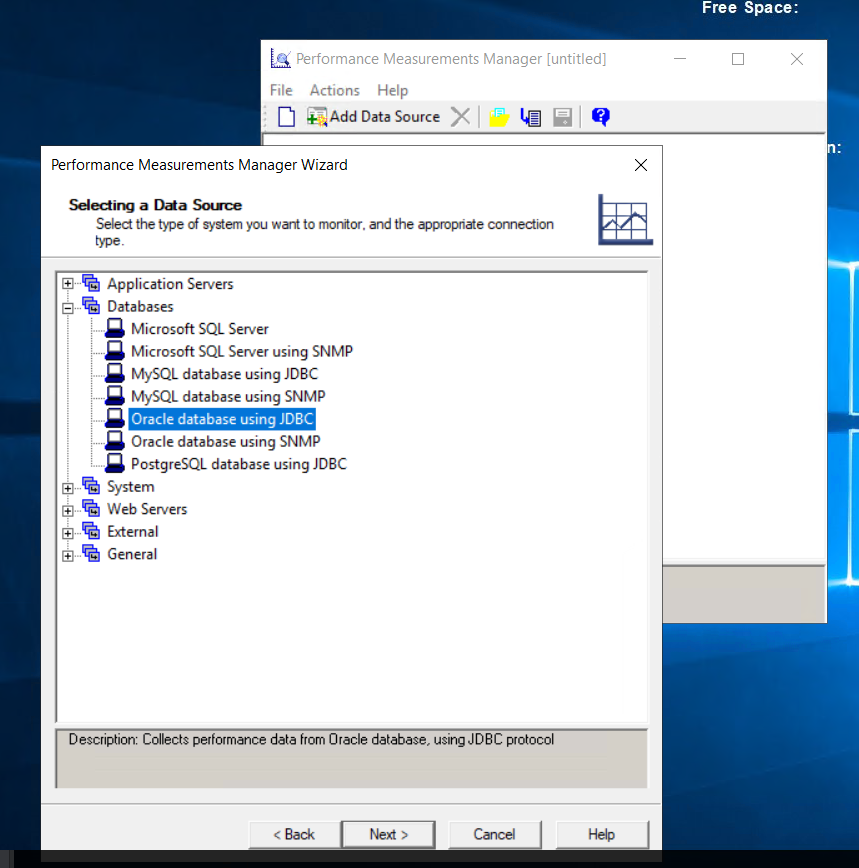


### Adding JDBC

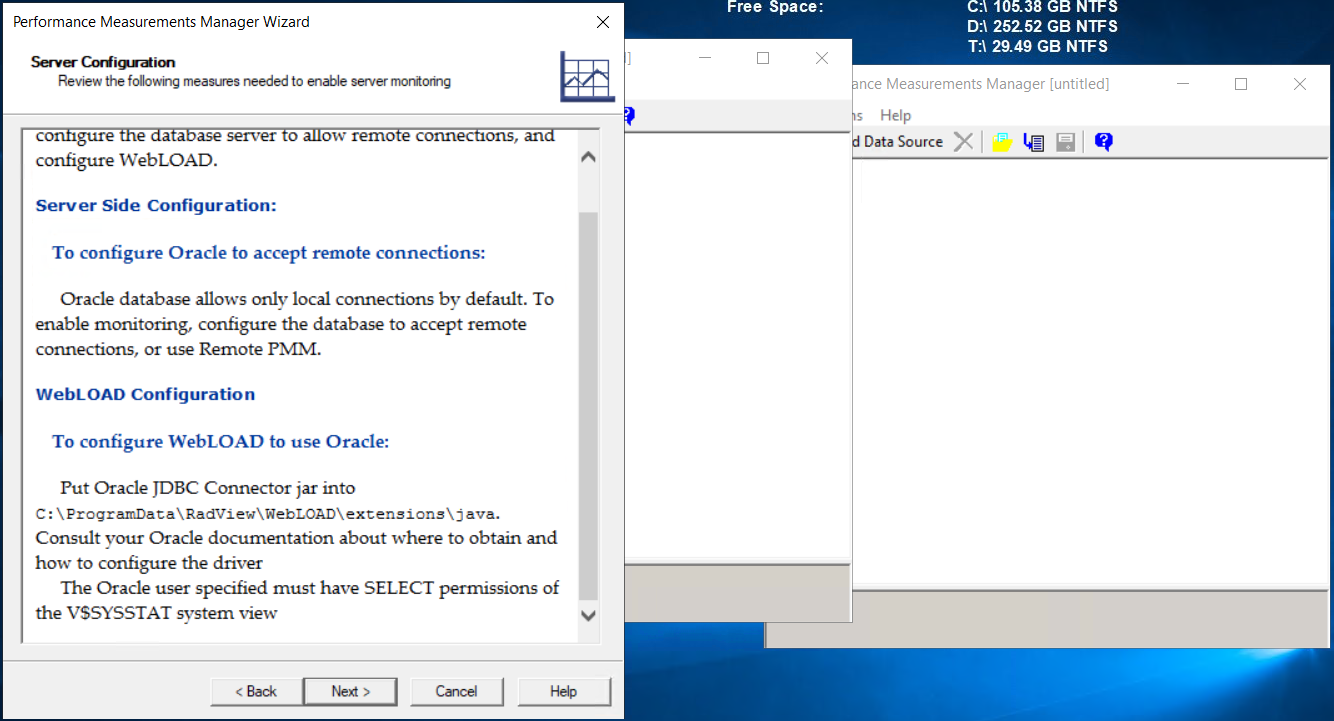
1. Make the server-side setup. Please refer to the jira ticket below to get the server side setup done by DBA team. <https://usfjira.atlassian.net/browse/HELP-122619>
2. Take care of ACLs. Here is the example jira ticket that helps to raise an ACL request.

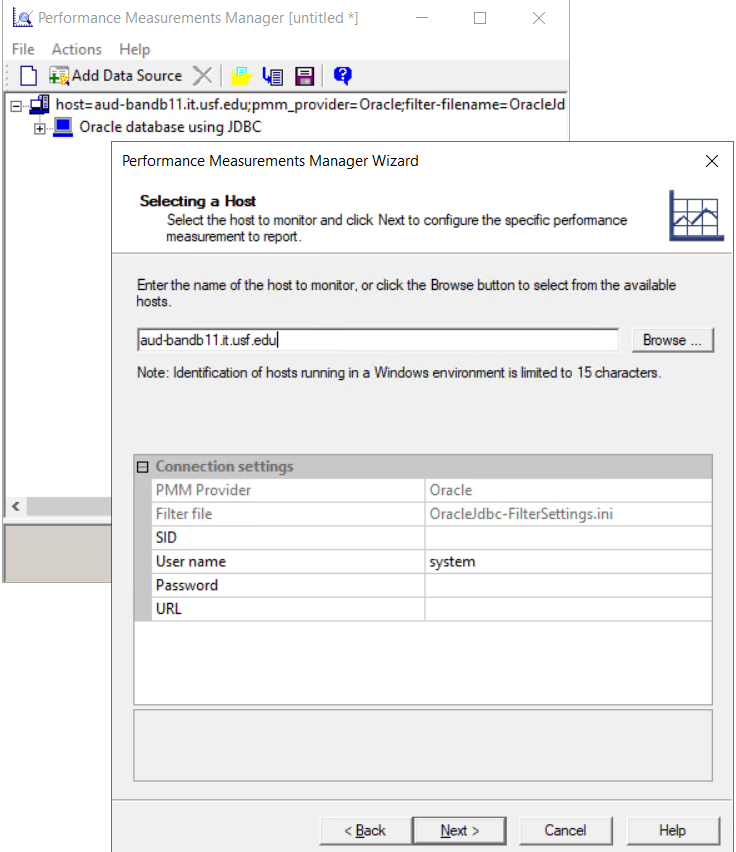
<https://usfjira.atlassian.net/browse/HELP-122349>

1. Add PMM to Console server and create a .pmm file for future use, by following the screen shots shown below.



Make sure to place the ojdbc8.jar file in WebLOAD\extensions\java folder as mentioned in the screen shot below.

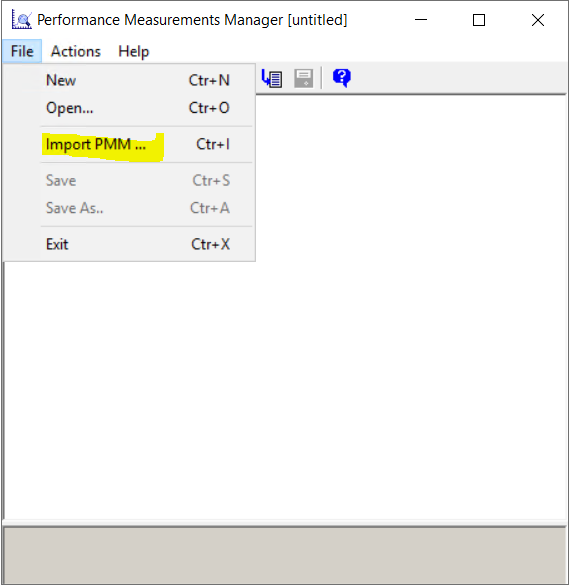




Add the server details, click on next and finish the setup.

1. Import PMM File

We have an option available to import existing PMM file, so we can use that to create quick tests.



For more information on PMM and setup please check the page 343, chapter 23 of [Webload Console Users guide](https://usf.box.com/s/gxj5pqt685p7jxk8bbsv98vvjmuvpiq7).

# Game Days

Once we have test scripts recorded in WebLOAD recorder, Test data prepared, and templates from Console server, its time to test the capacity of Servers in terms of responding to the traffic.

Schedule a time to run your tests using templates. We called it a game day. Include DBA’s and developers to keep an eye on system performance during Load test sessions.

During the test, keep an eye on metrics from using options from console and dashboard based on your convenience. DBAs need to monitor all the performance metrics from their end.

Once the test is run, based on the performance, plan for another game day changing system configuration for better performance. Usually, the test templates stay same as we will be testing the same scenarios with better system configurations. Make sure to prepare the test data again for new game days.

Prepare reports with metrics with meaningful performance graphs and publish them. Circulate the published reports to concerned teams. We keep getting better at performance analysis every time with each game days. It’s completely based on practice.

# Appendix 1 – Glossary of Terms

|  |  |
| --- | --- |
| Term | Definition |
| Agenda | WebLOAD refers to the recorded load testing scripts as *agendas*. |
| Transaction | Transactions are logical groupings of actions inside a WebLOAD agenda. |
| Load Profile | The number of virtual clients to use and the distribution of those clients over time. |
| Template | The WebLOAD file used by the WebLOAD console to execute a load testing session. |
| Session | The results of a load test. The data file used to store the statistics of a test run. |
|  |  |
|  |  |

# Appendix 2 – Challenges and Risks

Some things will make it more difficult to test your application. Among these are:

* Two Factor Authentication
  + One reason for two factor authentication is to prevent bots from spamming a site. However, this is effectively what load testing is. So 2FA needs to be disabled for WebLOAD to work.

# Appendix 4 – Contacts

|  |  |  |
| --- | --- | --- |
| .Team | Contact | Description |
| Radview | [support@radview.com](mailto:support@radview.com) | Vendors contact for help with the WebLOAD tools. |
| DBAs |  |  |
| Cloud Services |  |  |
| Jeremy Radwan | [jradwan@usf.edu](mailto:jradwan@usf.edu) | Administrator for the console server and load generators. |
| Timothy Adams | [tadams@usf.edu](mailto:tadams@usf.edu) | Experience with WebLOAD tool |
| Anusha Ande | [anushaande@usf.edu](mailto:anushaande@usf.edu) | Experience with WebLOAD tool |
|  |  |  |

# Appendix 4 – Helpful Resources

Links that might be helpful:

* <https://www.radview.com/>
* <https://www.youtube.com/c/Radview/videos>
* [WebLOAD Recorder](https://usf.box.com/s/7efti0jdpknh6e7gjjondjpdke48ooaj)
* [WebLOAD Console](https://usf.box.com/s/gxj5pqt685p7jxk8bbsv98vvjmuvpiq7)
* [WebLOAD Analytics](https://usf.box.com/s/nh0h5b5rna6b6urtv2fnk0ti8b7ppma4)
* [WebLOAD Dashboard](https://usf.box.com/s/nh0h5b5rna6b6urtv2fnk0ti8b7ppma4)