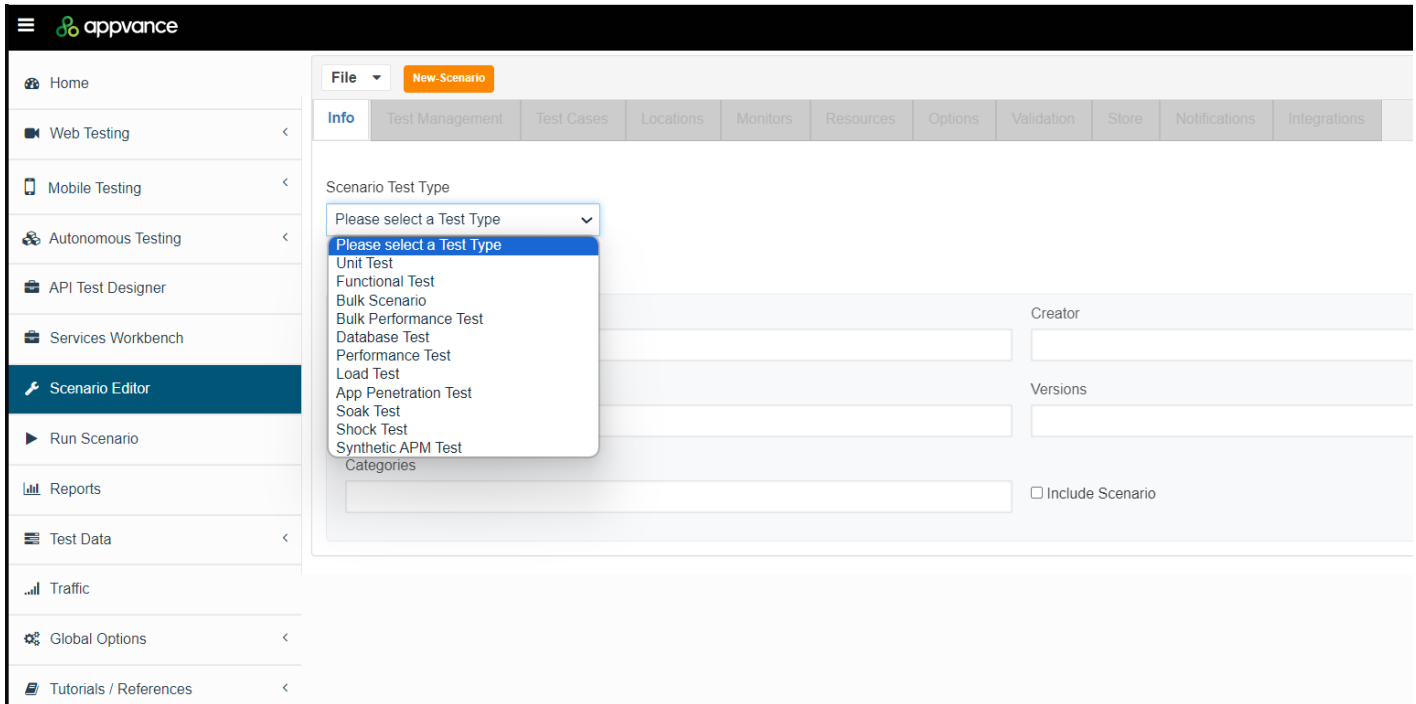


Types of Tests

- [Types of Tests](#)
- [Bulk Scenario](#)
- [Requests per Second and Transactions Per Second](#)

Types of Tests

Types of Tests



The following scenario test types are available:

- [Unit Test](#)
- [Functional Test](#)
- Bulk Scenario
- Bulk Performance
- [Database Test](#)
- [Performance Test](#)
- [Load Test](#)
- App Penetration Test
- Soak Test
- Shock Test
- Synthetic APM Test

Unit Test

Unit tests are used to test small, discreet parts of an application. Utilize your favorite domain-specific language (Selenium, C#, Java) at the API level or UX level.

Below is the list of Script types that a user can select in the test case panel when working with Unit Testing.

- Appium TestNG
- Command Line
- Command Line Resource
- Java
- JRuby
- Service Suite (Services Workbench)
- TestNG
- API Test

Functional Testing

Also known as a regression test or smoke test, is used to ensure that the existing functionality of an application works as new features are added. Run all your functional tests in series or parallel. Record tests in minutes and parameterizes them without any coding needed.

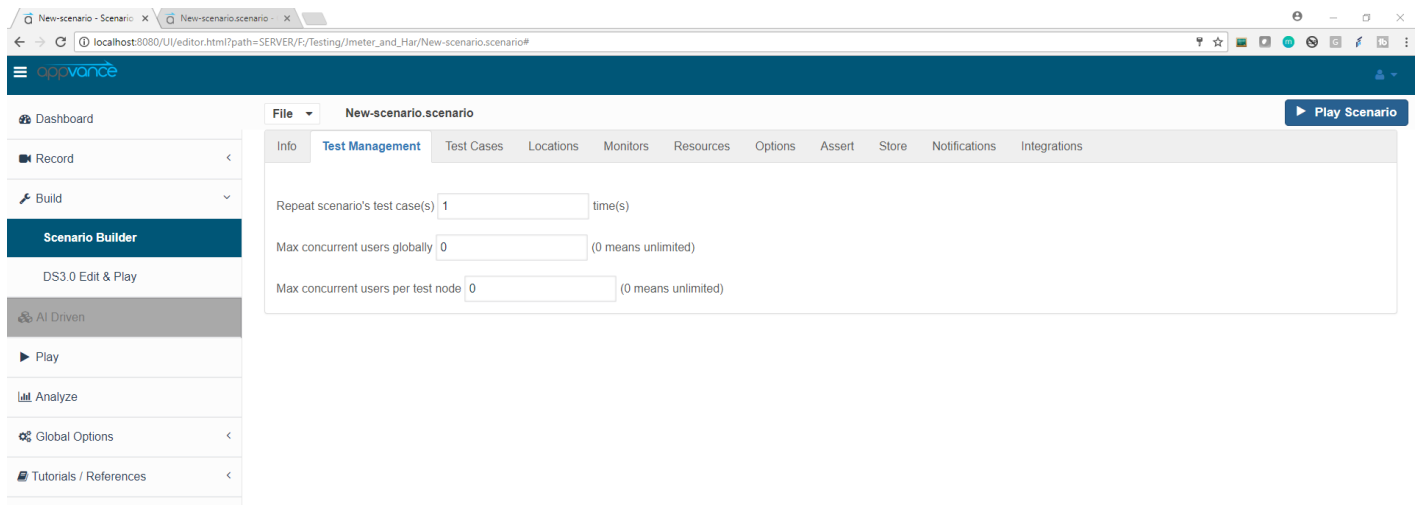
Below is the list of Script types that a user can select in the test case panel when working with Functional Testing.

- Appium-testNG
- Test Designer
- Mobile Designer
- Appvance Java Technology
- Command Line
- Command Line Resource
- Groovy
- HAR
- Java
- JMeter
- JRuby
- Selenium
- Service Suite (Service Workbench)
- SoapUI
- SQL
- TestNG
- API Test

SoapUI support was removed in release 5.2.0 of AIQ.

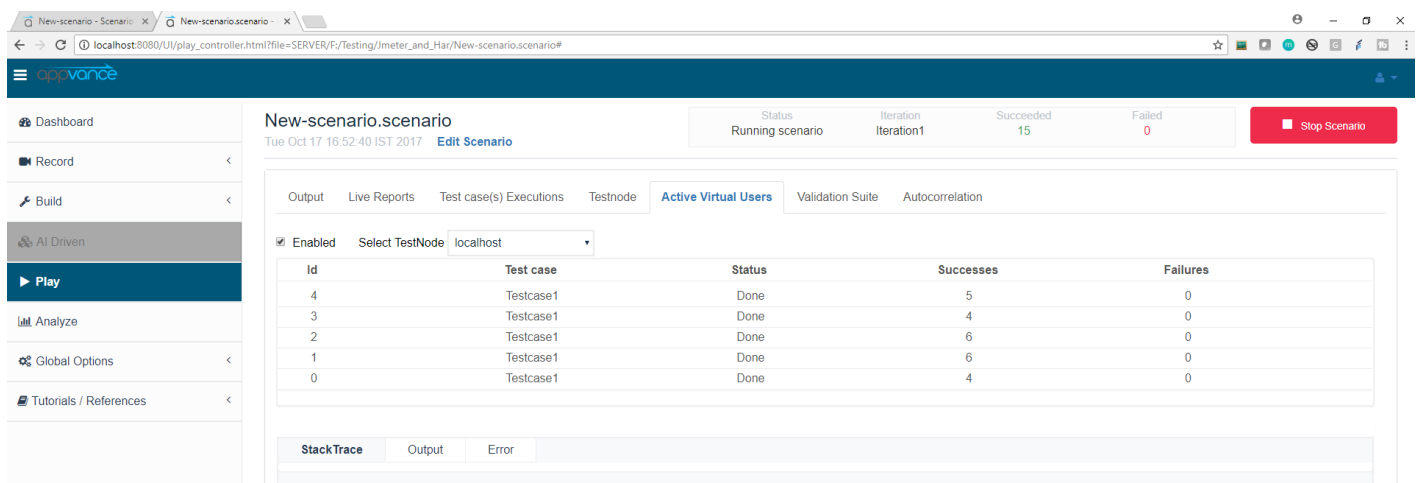
For a **Functional Test**, the user can choose how many times to **Repeat this test-case** option from the **Test Management** Panel

Providing a value in "Repeat this use case", the test case that is defined will run the specified number of times sequentially.



With functional testing, users also have options to control how many test cases to run concurrently with the options "Max Concurrent users globally" and "Max concurrent users per test node".

Assume a scenario is created with 3 test cases (Appvance Designer scripts) and "Max concurrent users per test node" is set to 2, After saving and clicking Play scenario, While running this scenario navigate to the Active Virtual Users tab on the controller window and select Enabled check-box, user can see two users executing and 1 user in waiting for status.



Also, users can set a maximum number of concurrent executions considering all test nodes using the "Max concurrent users globally" option.

NOTE: Normally all test cases in a functional test will run concurrently.

Database Test

Database tests enable you to run SQL scripts to validate your DB with every build.

Please refer to this chapter for more information: Database Testing in Appvance IQ.

Load and Performance Test

Performance Tests: Simulate millions of users with real use cases from beginning to end (from the UX to the back end). Identify performance issues including client-side code. Runs using use cases already created for functional tests.

Load Tests: Simulate users, usually at the API level, over varying and increasing loads. Ability to use existing frameworks like SoapUI, JMeter, etc.

Load and Performance tests operate the test cases in one or more virtual users as many times as possible in a given amount of test time or with Different Completion Criteria available. By running a load test at several levels of virtual users, AIQ identifies the Scalability Index of your application's scalability.

Choose one or more virtual user levels to operate the test. Click Add virtual user level to add a level and Define an Iteration.

Below are some of the bookkeeping activities for your test scenario, none of these fields are required fields.

Author Name - The name of the author of the test scenario details can be saved, this detail is also shown in the CSV iteration summary report.

Links - Any link details can be mentioned here.

Versions - Can be used for version control.

Categories - This is a project category, once entered shall be shown on Analyze and Play pages

Include Scenario - Include Scenario is a useful feature. You can have one in include per each possible database you want to use, or per each set of predefined locations or cloud configurations.

You can have a predefined set of scenario files that typically have only details of Locations and Logs.

When you want to run tests in so and so locations and so and so Database details, all those can be created as individual files and can be included here and those scenarios shall be run in those mentioned locations and logs will be stored in the mentioned databases.

Example Include Scenario file below:

```
<testscenario xmlns="www.pushtotest.com/tm5"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="www.pushtotest.com/XSD/testscenario.xsd" version="3">
```

```
<testnodes>
<node name="cloudhost" location="http://localhost:8080/TMServer/ws/TestNode"/>
</testnodes>
<logs>
<log type="database">
<opt name="hibernate.connection.username" value="appvance"/>
<opt name="hibernate.connection.password" value="appvance"/>
<opt name="hibernate.connection.url"
value="jdbc:oracle:thin:@smoketestutp30oracle.cjzzvczxudhc.us-east-
1.rds.amazonaws.com:1521:ORCL"/>
<opt name="hibernate.default_schema" value="utp"/>
<opt name="hibernate.connection.driver_class" value="oracle.jdbc.OracleDriver"/>
</log>
</logs>
</testscenario>
```

Bulk Scenario

What if there are 100's or 1000's test designer or HAR scripts and we need to browse each of them in Scenario Editor to run them as Functional or performance Tests, which is of course a tedious task to do, to browse all those 100 or 1000's of scripts individually.

The answer for the above limits would be **Bulk Scenario** where you could browse the entire directory at once and you could execute all those 100's or 1000s of scripts either in sequence or in parallel without having to browse each of those scripts individually.

Bulk Scenario is a scenario type in Scenario Editor that lets you browse the root directory and a directory inside the root directory where the scripts are located.

Bulk Scenario supports executing the following script types:

- Designer Scripts
- Java Scripts
- HAR
- Jmeter

A separate script type called **Bulk Test Designer** and **Bulk HAR** is added to support browsing the directory.

To begin with, Navigate to Scenario Editor and chose **Bulk Scenario** as the Scenario Type.

The screenshot displays the Appvance IQ Scenario Editor interface. On the left is a sidebar with navigation options: Dashboard, Test Designer, AI Scripting, Javascript DPL, Scenario Editor (highlighted), Traffic, Play, Analyze, Global Options, and Tutorials / References. The main area has a top bar with 'File' and 'New-scenario2.scenario' and a 'Play Scenario' button. Below this is a tabbed interface with 'Info' selected. The 'Info' tab shows 'Scenario Test Type' set to 'Bulk Scenario'. A message says 'Please select a Scenario Test Type'. Below are input fields for 'Author Name', 'Creator', 'Links', and 'Versions'. There is also a 'Categories' field and an 'Include Scenario' checkbox.

It is important to choose the **Root folder** under the **Test Management** tab

appvanceIQ

Dashboard

Test Designer

AI Scripting

Javascript DPL

Scenario Editor

Traffic

Play

Analyze

Global Options

Tutorials / References

File

New-scenario2.scenario

Play Scenario

Info

Test Management

Test Cases

Locations

Monitors

Resources

Options

Validation

Store

Notifications

Integrations

Root Folder

SERVER/F:/Testing/BulkTesting/

Browse

Relative

Max concurrent users globally

0

(0 means unlimited)

Max concurrent users per test node

0

(0 means unlimited)

So why do you have to choose the Root folder for the Bulk Scenario?

Bulk Scenario would be executing the scripts which are based on the root folder, All the .ds, .csv or any supporting files are relative to this root folder, not to the scenario file.

Another reason is to have control, all the scripts and supporting files should be under the root folder so the bulk scenario has much more control concerning the directories that you choose.

The next step is choosing **Bulk Test Designer** or the **Bulk HAR** as the script type under Script Type and choosing the directory with all the scripts.

Testcase1

Importance

Normal

Save

Cancel

Setup

Add Setup

Script

Script Name

Untitled1

Script Type

Bulk Test Designer

Resource

BulkTesting/Scripts/

Browse

Language

DS

Browser

Chrome

Regex

*

Instance

Add DPL

The screenshot shows a web-based interface for configuring a test case. At the top, there's a header bar with a text input 'Testcase1', a dropdown 'Importance' set to 'Normal', and 'Save' and 'Cancel' buttons. Below this, the interface is divided into sections. The 'Setup' section is currently empty. The 'Script' section is active and contains a form with several fields: 'Script Name' with the value 'Untitled1', 'Script Type' with a dropdown menu showing 'Bulk HAR', 'Resource' with the text 'BulkTesting/Scripts/' and a 'Browse' button, 'RegEx' with the text '*', and an empty 'Instance' field. There is also an 'Add DPL' button with a folder icon. On the right side of the 'Script' section, there are links for 'Add Setup' and 'Add Script'.

Save the Test case and save the scenario and play. This will execute all the files in the selected folder. It's that simple!

Some common questions you might get:

What if there are 100 scripts in the folder and there is a combination of .ds and .js, how do I execute only .ds or .js scripts?

If you had noticed in the test case, After selecting Bulk Test Designer, you could choose the language, And also there is a field called **RegEx** where you could provide a regular expression matching either ds or js.

Example:

*.ds will match all the designer scripts.

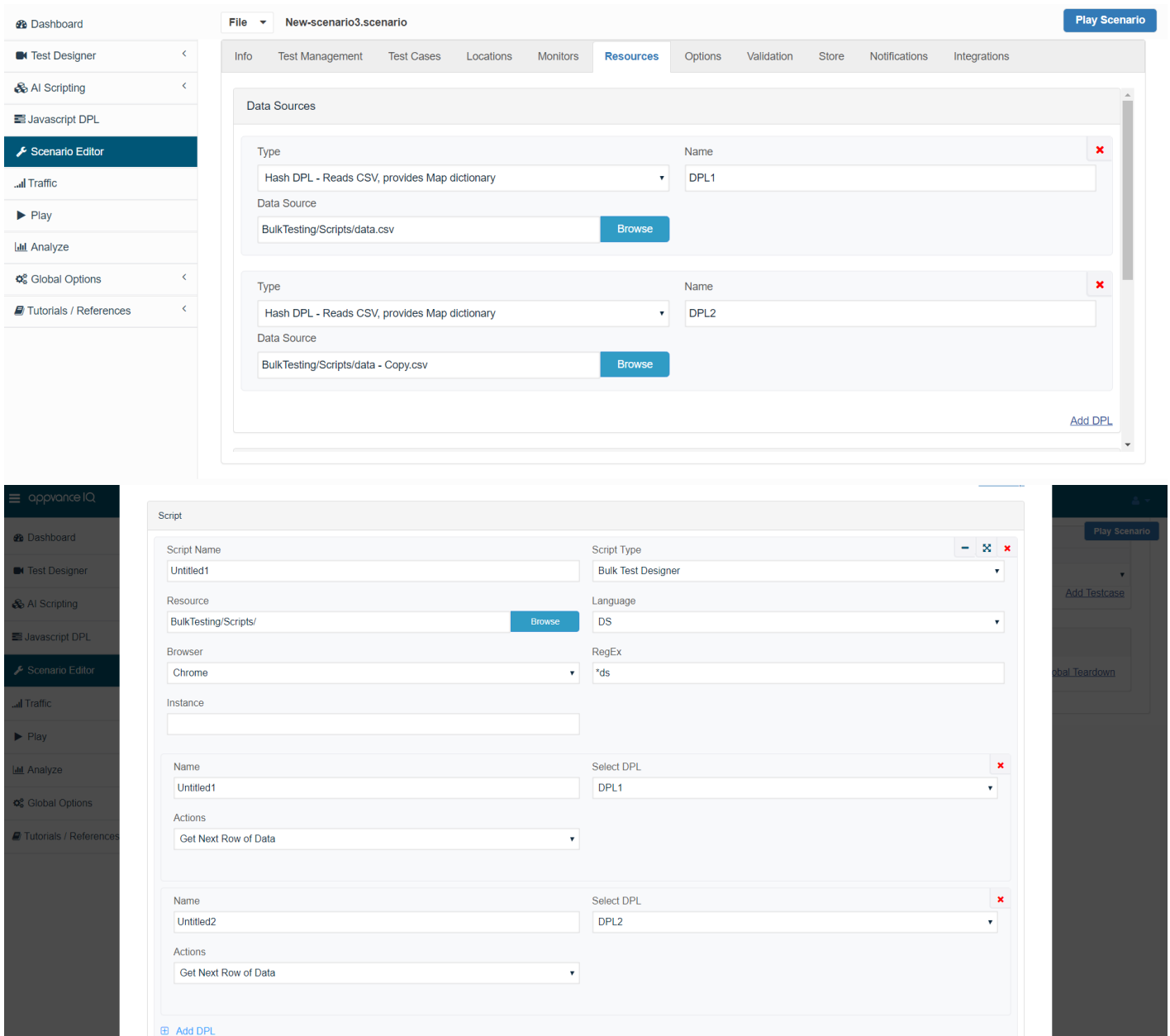
*.js will match all the JS scripts.

What if there are different nested folders inside the chosen folder?

The bulk scenario will execute all the scripts inside those nested folders as well.

What if the script is data-driven and all those supporting files (.CSV) are in the same folder as the designer scripts?

Data-driven scripts work the same way as working with the normal Test designer script types. Just to go Resources tab and create the required DPL (Even if there are several DPLs to be created), after defining the DPLs, you could come back to the test cases panel and mention those DPLs below the test cases.



My scripts use include files and those included files are in the same folder as the main script? How do I make sure the Bulk scenario executes only the main scripts and does not repeat to execute the same included scripts?

Make sure to make use of Regular expression matching to execute only the main scripts like `main*.ds` - This way it will exclude executing all the scripts matching the regular expressions and ignore the ones that do not match.

Requests per Second and Transactions Per Second

When running load or performance tests, you see the requests per second (RPS), transactions per second (TPS), or a successful number of transactions per second (Success TPS) as part of the execution reports.

These options were added in the 5.0.0 release of AIQ.



The graphic above shows the RPS, TPS, and Success TPS details for a scenario execution. While the graphs for RPS and TPS are similar, they are not identical. Requests per second (RPS) is a different measurement than transactions per second (TPS). This is because a single transaction may consist of several requests. For example, a single login operation may contain five HTTP requests that can be grouped together as a single transaction.