



Appvance IQ

Release 5.2.0

Release Summary

Document Version 1.2

September 25, 2024

© 2024 Appvance Inc. All rights reserved

TOC

	1
About Appvance	5
Appvance IQ 5.2.0 Release	7
About this Document	7
Accessing AIQ 5.2.0	8
AI Script Generation in AIQ 5.2.0	9
Updating 5.0.x AISG Files to AIQ 5.2.0	11
New Coverage Map Types	12
Search and Filtering Improvements on the Coverage Map	15
Visual Accessors in AISG	19
AI Training Enhancements	25
SmartTag Visual Details	28
AWS Instance Test Node	29
Considerations for Custom Actions and SmartTags	30
AISG Repository Considerations	30
Mobile Application Testing	32
Mobile Designer - New User Interface Overview	33
Repository and File Manager in the Mobile Designer	36
Mobile Configuration	40
Mobile Designer Editor Options	46
Mobile Designer Test Options Details	49

Script Execution Options	50
Playback and Recording Options	50
Mobile Designer Test Options Comparison	52
Image Injection	54
Geo Location	62
Desktop Application Testing	63
Desktop Designer User Interface Overview	65
Desktop Designer Configuration	69
Creating a Desktop Test	72
Desktop Designer Test Options	75
Desktop Designer Export to Javascript	78
Desktop Designer Example - Kindle Desktop App	80
API Testing	94
New API Testing Building Blocks	96
Postman Conversion	100
Functional Changes	101
CICD Git Dashboard Housekeeping	101
Removed Support for SoapUI	101
AISG - Creating Custom Actions	101
User Management - Teams	101
Delete Execution Reports	102
Timestamp Displays	103

Iteration Summary Reports Optional	103
Scenario Editor	104
DataDog APM Integration	104
AIQ 5.0.x Enhancements	105
Scenario Execution Data	105
Added Support for Encryption of JavaScript DPL and Hash DPL	105
Added Add Encrypt/Decrypt DPL to Actions	106
Improved Resource Management in Scenario Editor	106
Enable Extraction from the Whiteboard within a SQL Script	107

About Appvance

Appvance is the technology leader in and inventor of AI-driven autonomous testing, which has revolutionized software testing.

Visit us at [Appvance.ai](https://appvance.ai).

Support

Contact our support team to request technical help, report an issue, or suggest a new feature. See the [Appvance Support](#) page for more information.

Visit the [Appvance Service Desk](#) to submit a request.

Online Documentation

Visit the [Appvance Documentation](#) site for AIQ product documentation.

Training and Certification

Your journey to Autonomous Testing Engineer Certification (ATEC) starts here. Take our classes and be among the very first to use and deploy the world's only Level-5 Autonomous software testing technology.

We offer self-paced classes, scheduled classes and tailored training. See the [Training and Certification](#) page for more information.

Content Library

[Explore the content library](#) to access a range of resources to help you succeed in all your quality initiatives. Understand best practices, see examples of success, and learn more about AIQ in action.

News

Visit the [Appvance Newsroom](#) to read about our most recent announcements and happenings.

Events

Connect with Appvance, learn more about AIQ, and engage in thought-provoking conversations on delivering world-class software quality, autonomous testing, and more.

[See where you can meet up with the Appvance team.](#)

Blog

[Read the blog](#) and learn about testing strategy & methods, AI, and get AIQ product updates.

Contact Appvance

Online: [Contact form](#)

Email: info@appvance.ai

Phone: (845) 254-1164

Mailing address: 3080 Olcott Street Suite B240 Santa Clara, CA 95054

Appvance IQ 5.2.0 Release

The Appvance IQ (AIQ) 5.2.0 release includes a significant leap forward in the world of AI-native automated testing. This release introduces Generative AI V3, a comprehensive update that enhances performance, reporting, visual object recognition, and bug-finding accuracy.

About this Document

This document contains information on the following functional enhancements and functional changes in the AIQ 5.2.0 release.

- "AI Script Generation in AIQ 5.2.0" on page 9
- "Mobile Application Testing" on page 32
- "Desktop Application Testing" on page 63
- "API Testing" on page 94
- "Functional Changes" on page 101
- "AIQ 5.0.x Enhancements" on page 105



Depending on what AIQ release you are upgrading from to AIQ 5.2.0, there may be additional functional enhancements included in this release. For your convenience, information about the enhancements in the AIQ 5.0.0 release has been added to this document. See "AIQ 5.0.x Enhancements" on page 105 for more information.

Accessing AIQ 5.2.0

Appvance customers can access the [5.2.0 release page](#). The release page includes a link to the AIQ 5.2.0 installer, the list of fixes included in the release, information on upgrading from a previous release of AIQ and other implementation considerations.

See the "Upgrading your Controller and Test Nodes" section of the [5.2.0 release page](#) for upgrade information.

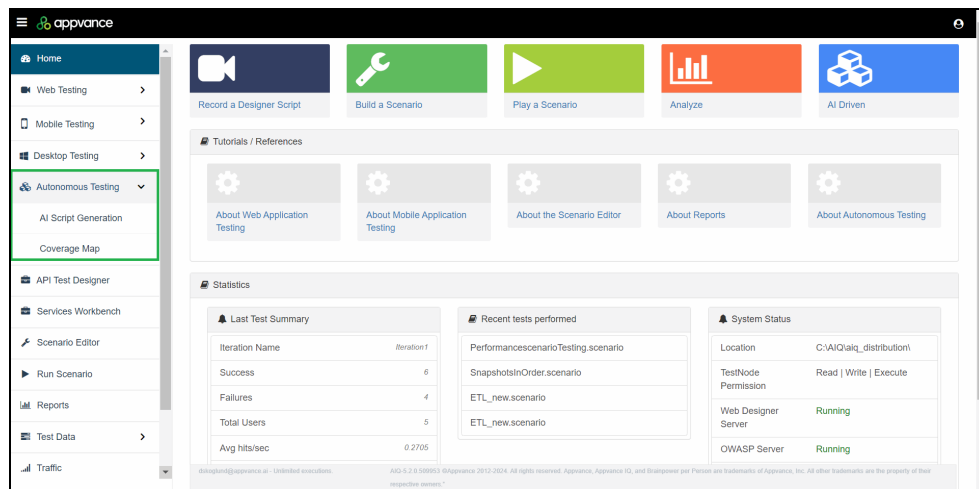


See "About Appvance" on page 5 for contact information.

AI Script Generation in AIQ 5.2.0

User Interface and Terminology Changes

The AIQ 5.0.x menu option **Blueprint Designer** has been renamed to **AI Script Generation** in AIQ release 5.2.x.



In addition to this change, the terms “AI Exploratory Testing” and “AI Blueprinting” are no longer used. Those functions are now referred to as “Autonomous Testing” and “AI Script Generation” (AISG). The user interface and documentation will be updated to reflect these changes.

The term "Blueprint" is still used to refer to the Blueprint (.abpt) file that is created during an AISG execution.

New Features and Enhancements

In addition to the following new features and enhanced functionality, there were also extensive improvements to the AI Engine that powers AISG.

- "New Coverage Map Types" on page 12
- "Search and Filtering Improvements on the Coverage Map" on page 15
- "Visual Accessors in AISG" on page 19
- "AI Training Enhancements" on page 25
- "SmartTag Visual Details" on page 28
- "AWS Instance Test Node" on page 29

Upgrade Considerations

- "Updating 5.0.x AISG Files to AIQ 5.2.0" on the facing page
- "Considerations for Custom Actions and SmartTags" on page 30

Updating 5.0.x AISG Files to AIQ 5.2.0

For the new features introduced in AIQ 5.2.0 such as the new Coverage Map types and Visual Accessors you must rerun any existing 5.0.x executions on 5.2.0 to take advantage of these new features.

For example, while you can open a 5.0.x Blueprint file (.abp) in AIQ 5.2.0, the new Coverage Map types will not be available automatically. The new Coverage Maps will only be available after the AISG execution is run using AIQ 5.2.0.

You can either add a new execution or export a Blueprint file to a template and create a new AISG from the template.



See [Add an Execution](#) or [AI Script Generation Templates](#) for more information.

New Coverage Map Types

AIQ 5.2.0 contains significant improvements in the AISG Coverage Maps. You can now select from three different dynamic visualizations of the Coverage Map.

- Original
- Extended Potential
- Actual Path

You can use these Coverage Maps to effortlessly track the AISG execution's journey through the application being tested, leverage AI hints more effectively, and generate precise test scripts to replicate issues.



To take advantage of the new Coverage Map types you must rerun any existing 5.0.x executions on 5.2.0. You can open a 5.0.x Blueprint file (.abpt) in AIQ 5.2.0 but the new Coverage Map types will not be available automatically.

How are the types of coverage maps different?

- The Original and Extended Potential coverage maps show you where the AI can go from one page to another, however it does not mean that the AI followed that exact path.
- The Extended Potential coverage map is an enhanced version of the Original coverage map. It shows more information such as more page states and page relationships. This can show you relationships that the AI did not actually navigate, but learned about during the execution.

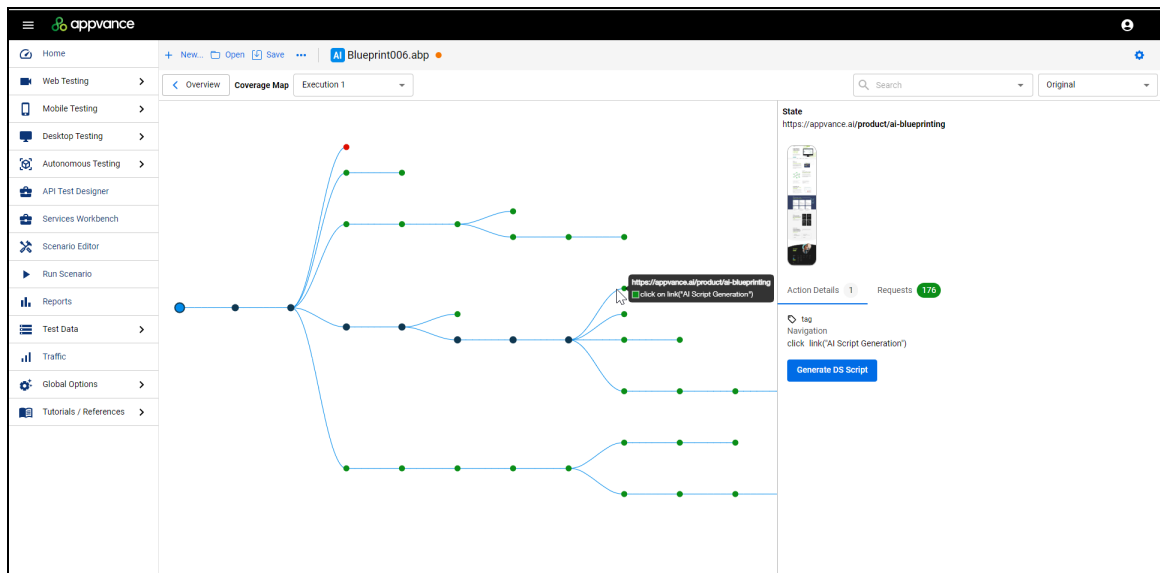
You will see more states and page relations on the Extended Potential coverage map than on the Actual Path coverage map. This is because

even though the AI didn't explore this, it is aware that those relationships exist.

- The Actual Path coverage map shows you the exact path that the AI followed through the application. This will let you track the AI training process and help you make further decisions on training the AI.

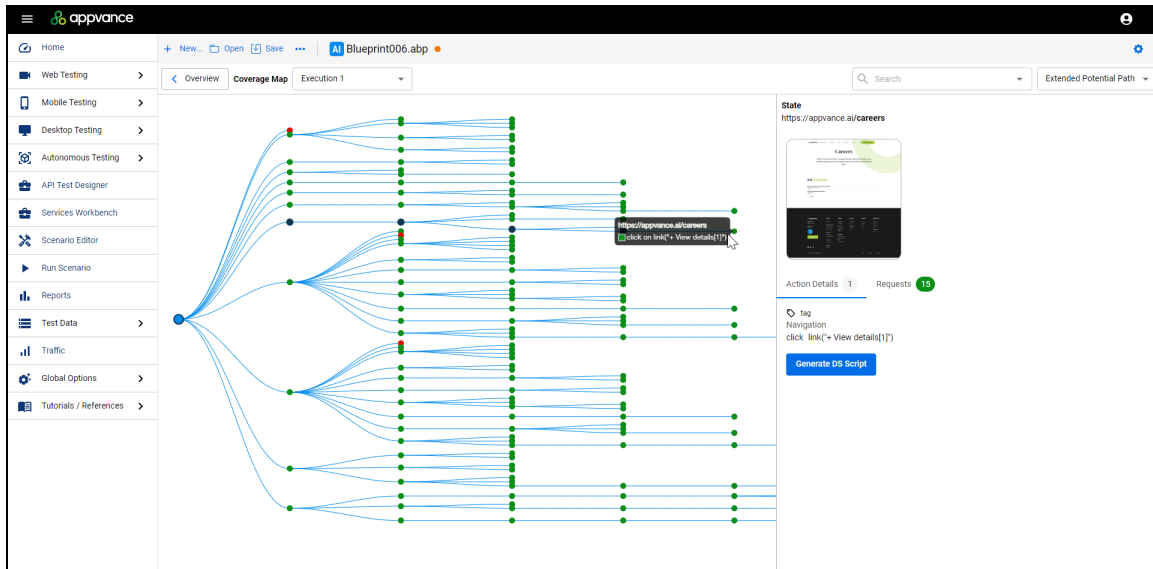
Original

This is the default visualization of the coverage map. It is the view that has been used in previous versions of AIQ.



Extended Potential

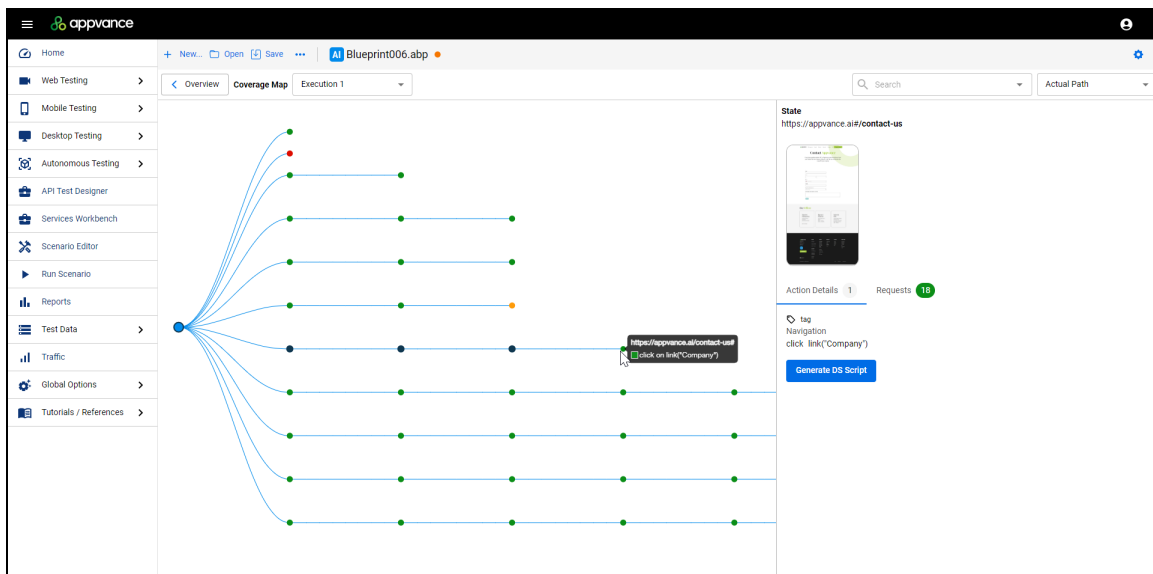
The Extended Potential view of the Coverage Map shows you the potential path the AI traveled to reach a particular page state. This view is similar to the Original view of the Coverage Map, but contains more states and page relationships.



Actual Path

The Actual Path view of the Coverage Map shows the actual path that the AI took when navigating the application under test. You can use this to track the AI training process and help you make further decisions on training the AI.

From the Actual Path Coverage Map, you can now accurately generate DS scripts that will contain the exact paths followed by the AI.

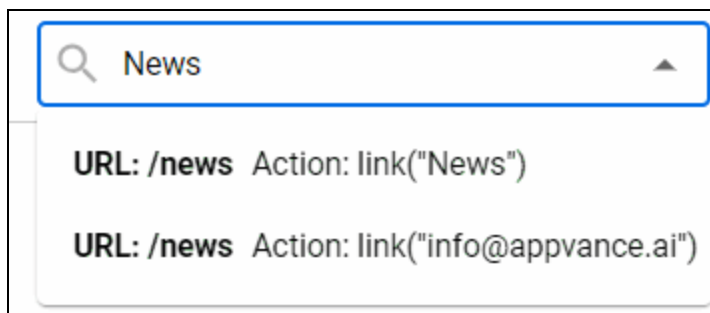


Search and Filtering Improvements on the Coverage Map

Improvements have been made in the Coverage Map to allow you to search and filter the results so you can explore the data more easily and efficiently.

Searching the Coverage Map

You can search the URLs in the Coverage Map. Click on a result to see the page state details.



Filtering Requests

You can now filter out specific requests or errors (404 errors, etc) from the Coverage Map. This action will be saved to the Blueprint (.abpt) file so that the requests will not display when the Blueprint is executed in the future.

When you hide a request on a particular node, that request will also be hidden for every other node that also has the same request.

Hiding a Request

- To hide a request on a node, right click on it from the list and select **Hide URL**. The request is filtered out of the list and added to the **Hidden Requests** list.

```
200 get https://demosit ... jpeg?1527769339
200 get https://demosit ... jpeg?1527769342
200 get https://demosit ... jpeg?1527769325
200 get https://demosit ... 230920145353.js
404 get https://demosit ... jpeg?1527769324
    //demosit ... jpeg?1527769324
    s://demosit ... =hcknujxx&end=1
200 get https://demosit ... 5318b67ceff.css
200 get https://demosit ... 9be4363ddc1.png
200 get https://demosit ... c040ef11c.woff2
200 get https://demosit ... jpeg?1527769343
200 get https://demosit ... pvance.net:443/
200 get https://demosit ... jpeg?1527769327
200 get https://demosit ... jpeg?1527769328
```

- To hide a request from the **Requests** list, select **Hide Request** from the **Action** menu (the three dots).

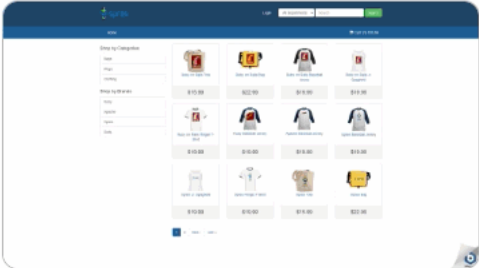
Method	Request	Status	Duration	Action
GET	https://demosite.appliance.com:443/spree/products/21/mini/tor_tote.jpeg?1527769324-8	404	36 ms	Hide URL
POST	https://demosite.appliance.com:443/orders/populate?	302	221 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	191 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	217 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	154 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	165 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	250 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	260 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	249 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	197 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	210 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	160 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	241 ms	...
POST	https://demosite.appliance.com:443/orders/populate?	302	237 ms	...

Viewing Hidden Requests

You can always choose to display all requests for a node have been hidden, or a list of all requests for the AISG execution that have been hidden.

- To view hidden requests on a selected node:
 1. On a node, click the **Requests** tab.

State
https://demosite.appvance.net//

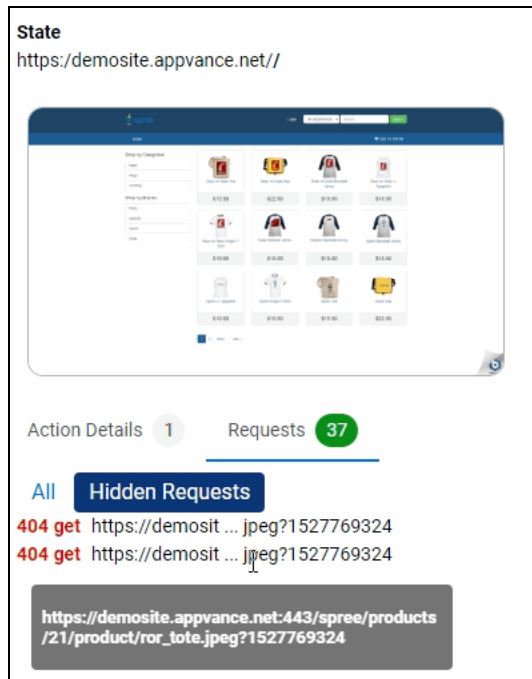


Action Details 1 Requests 37

All Hidden Requests

- 200 get https://demosit ... t?variant_id=11
- 200 get https://demosit ... jpeg?1527769340
- 200 post https://demosit ... =hcknujxx&end=1
- 302 post https://demosit ... orders/populate
- 200 get https://demosit ... y-on-rails-tote
- 200 get https://demosit ... jpeg?1527769338
- 200 post https://demosit ... =hcknujxx&end=1
- 200 get https://demositpng?1527769335
- 200 get https://demosit ... et:443/products
- 200 get https://demosit ... jpeg?1527769326
- 200 get https://demosit ... 48a19a26c362.js
- 200 get https://demosit ... jpeg?1527769337
- 200 get https://demosit ... ucts/spree-tote
- 200 get https://demosit ... corner-logo.png
- 200 post https://demosit ... =hcknujxx&end=1

2. Click **Hidden Requests**.



- To view all hidden requests for the AISG execution, navigate to the **Requests** page, and click **Hidden Requests**.

Request			
Method	Request	Status	Duration
GET	https://demosite.appvance.com:443/spree/products/21/mini/ror_tote.jpeg?1527769324=&	404	36 ms
GET	https://demosite.appvance.com:443/spree/products/21/product/ror_tote.jpeg?1527769324=&	404	62 ms

Visual Accessors in AISG

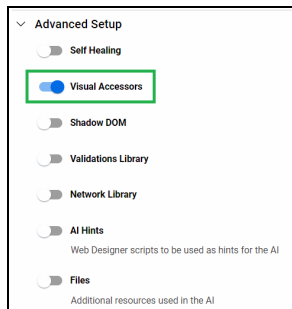
AIQ 5.2.0 has significant improvements in element-bounds for visual hints and the addition of Visual Accessors. Visual Accessors will be automatically captured when elements have fewer than four fallback accessors. In addition to the automatically captured Visual Accessors you have the option to manually create Visual Accessors for any element. You can also manually create them from scratch if needed.



For general guidelines on using Visual Accessors see [When to use Visual Accessors?](#)

Enabling Visual Accessors in AISG

Enable the **Visual Accessors** toggle in the **Advanced Setup** in the AI Script Generation configuration for the AI execution.

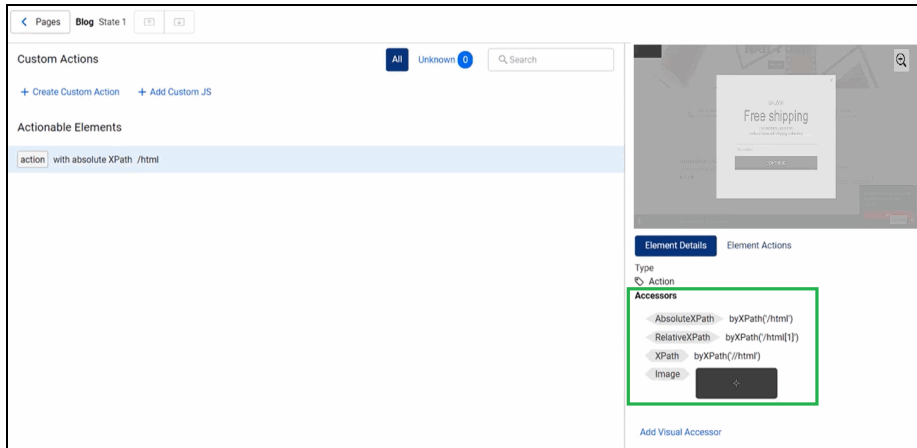


See the Advanced Setup Options section for more information on the available options.

Example

This example shows a situation where only three fallback accessors were identified, so a Visual Accessor was automatically added by the AI.

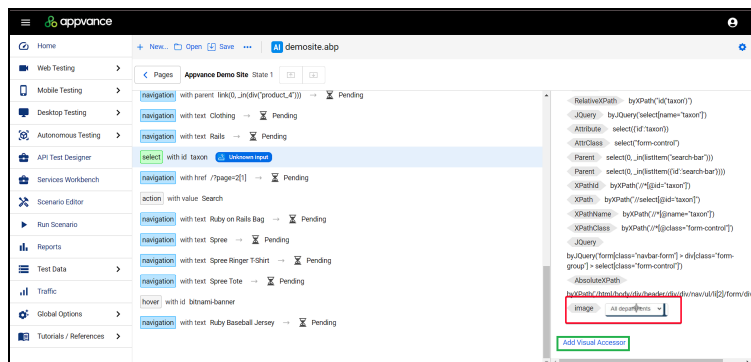
Visual Accessors are added at the end of the list of fallback accessors. This is for performance reasons. If AISG is able to match an element ID, it prefers to use that over a Visual Accessor. Using a Visual Accessor will slow down AISG, so it is used as a fallback accessor not as a primary accessor.



Manually Adding a Visual Accessor

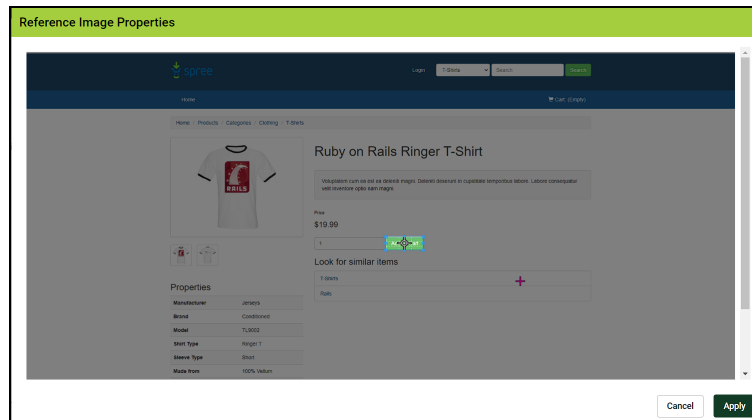
Visual Accessors can be manually added. Here is a basic example of the process.

1. Navigate to a page state and select the element for which you want to add a Visual Accessor.



2. Click **Add Visual Accessor**.

3. The **Reference Image Properties** window opens. From here, use the cursor to select your Visual Accessor.



4. Click **Apply** when you are done. The Visual Accessor is added.

Adding a Visual Accessor to a Custom Action

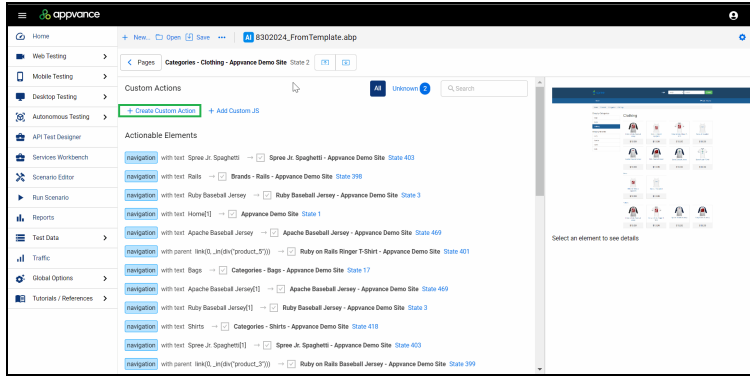
Visual Accessors can be added to Custom Actions. This is for situations where you want to create a Visual Accessor for an element that was not identified by AISG.

Here is a basic example of the process.

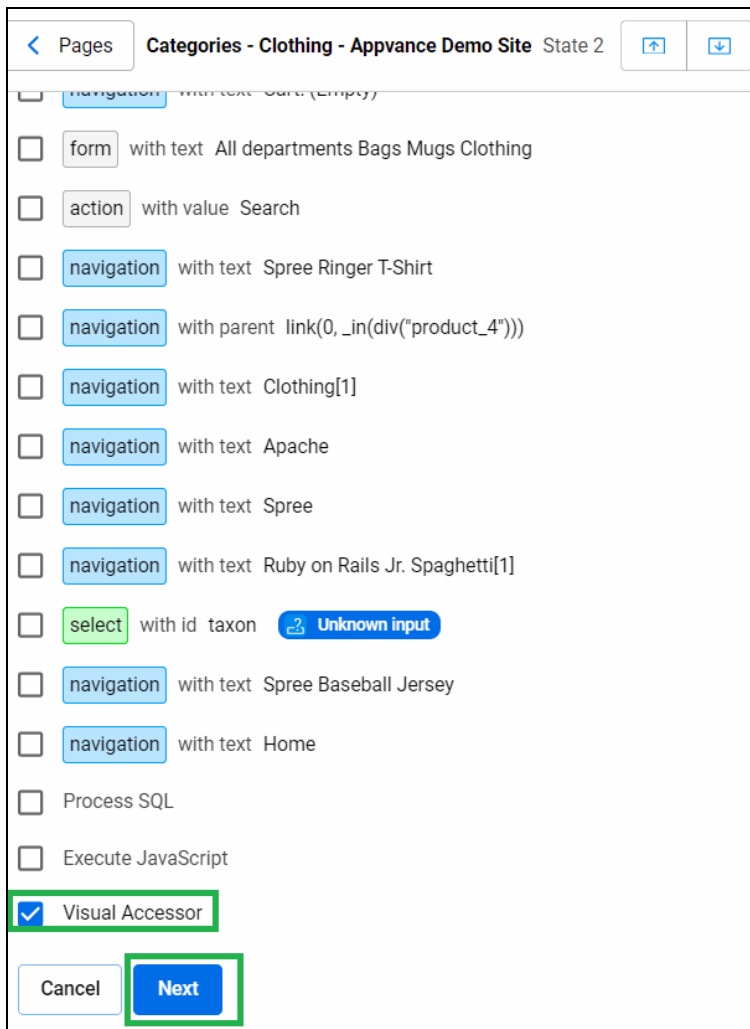


See [Creating Custom Actions](#) for general information on adding Custom Actions to resolved failed actions in AISG executions.

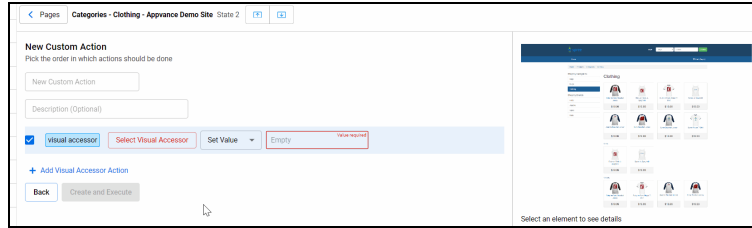
1. Navigate to the Page State for which you want to create a Custom Action.
2. Click Create **Custom Action**.



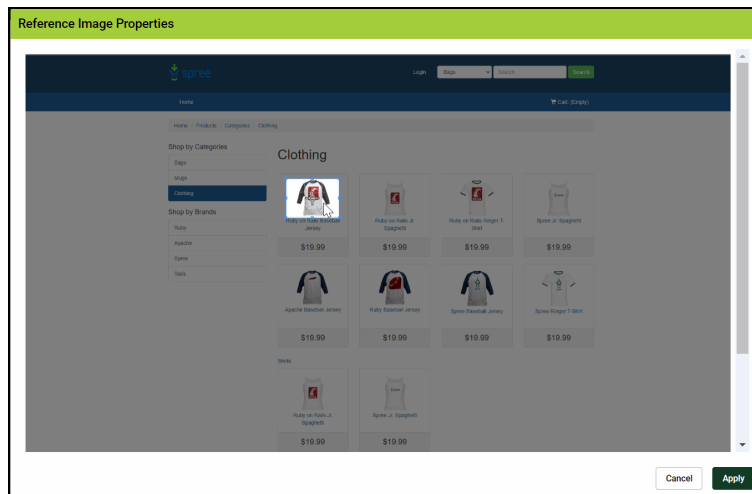
3. Select **Visual Accessor** and click **Next**.



4. Click **Select Visual Accessor**.



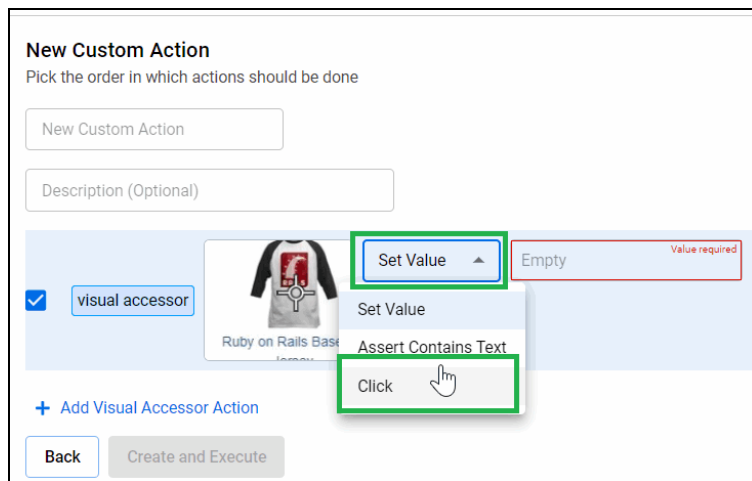
5. The **Reference Image Properties** window opens. From here, use the cursor to select your Visual Accessor.



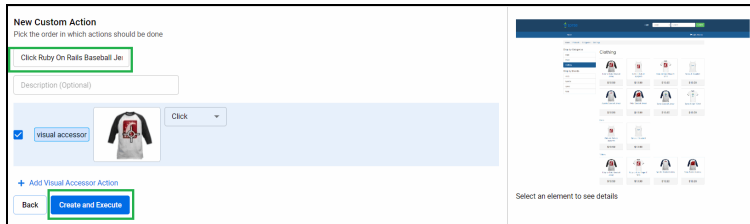
Click **Apply** when

you are done.

6. In the **Set Value** drop down, select **Click**.



7. Enter a name for the new Custom Action.



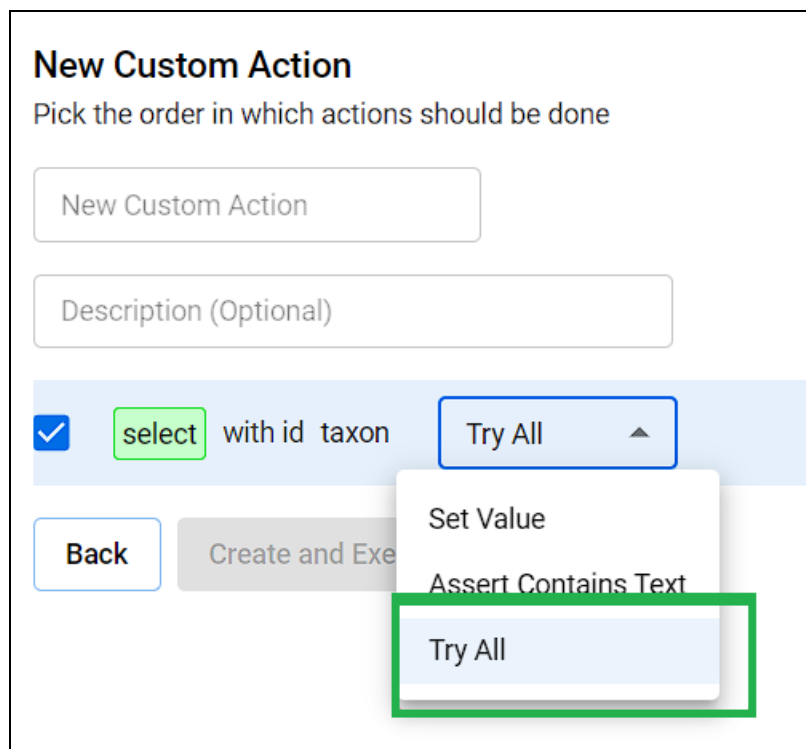
8. Click **Create and Execute**.

AI Training Enhancements

The following enhancements have been made to the AISG user interface to allow you to easily add additional resources and data to further train the AI on your applications.

Try All Option added to Custom Actions

The option to **Try All** has been added to Custom Actions. Custom Action can be created to resolve failed actions that occur during the AISG execution.



The screenshot shows the 'New Custom Action' form. It includes a title 'New Custom Action', a subtitle 'Pick the order in which actions should be done', and two text input fields: 'New Custom Action' and 'Description (Optional)'. Below these is a list of actions, with the first one 'select with id taxon' selected (indicated by a blue checkmark). A 'Try All' dropdown menu is open, showing options: 'Set Value', 'Assert Contains Text', and 'Try All'. The 'Try All' option is highlighted with a green border. At the bottom left, there are 'Back' and 'Create and Execute' buttons.



See [Creating Custom Actions](#) for more information.

Add Additional Resources

You can add additional resources to be used by the AI during the AISG execution. For example, you can add files that are required to be uploaded during

the test. You can also upload any files needed for any required SQL processing during the test.

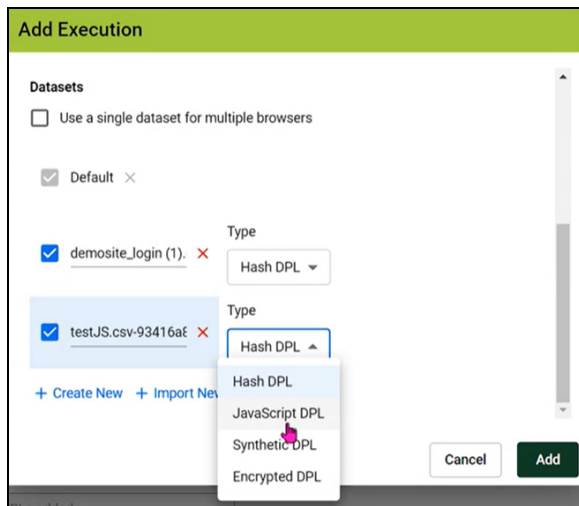
Resource files can be added in the Advanced Setup options in the AI Script Generation configuration for the AI to use them.



See the [Advanced Setup](#) options section in the documentation.

Support for Multiple Data Sets

You can add multiple data sets. There is a drop down where you can select the type of DPL that you are adding.



You can select from:

- Hash DPL
- JavaScript DPL
- Synthetic DPL
- Encrypted DPL
- Encrypted Hash DPL

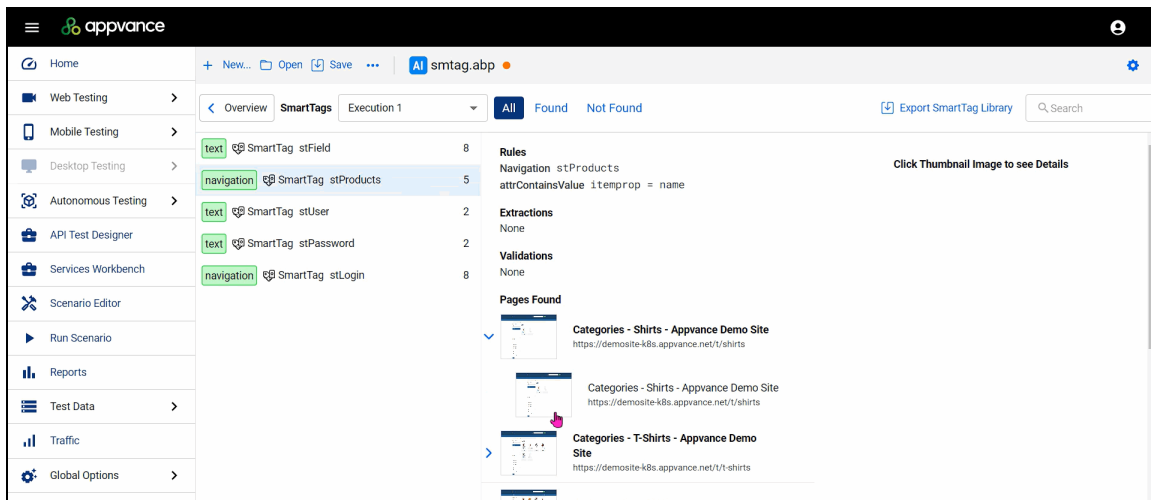


See [Add an Execution](#) for more information.

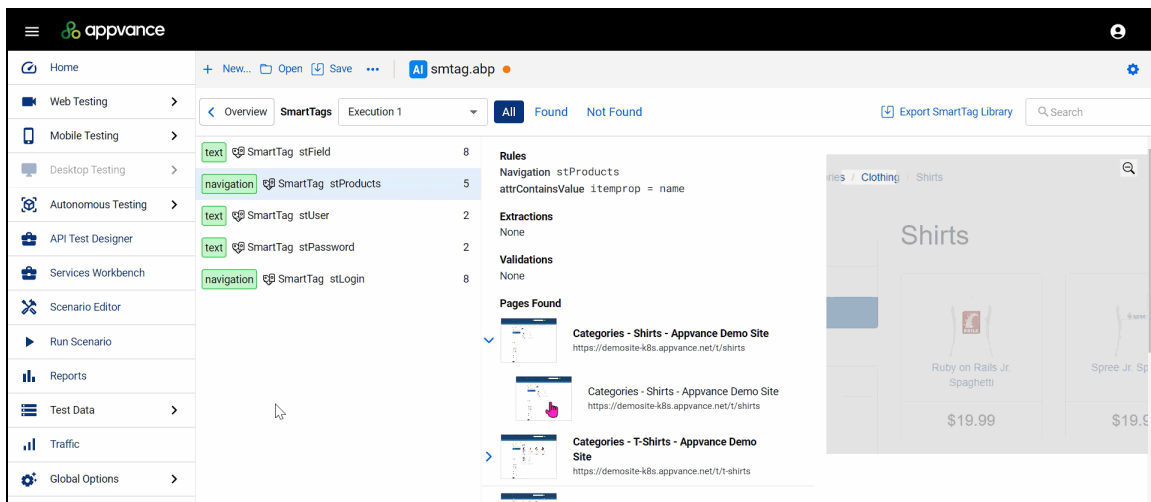
SmartTag Visual Details

You can now view details about the SmartTags that were found during the AISG execution.

If you click the **SmartTags** button from the Blueprint Overview, you can see details of the SmartTags within your Blueprint.



The **Click Thumbnail Image to see Details** link will show you a visual representation of where the SmartTag was found.



AWS Instance Test Node

The **Add Execution** windows has a new test node option in AIQ 5.2.0. You can select an **AWS instance** as your test node.

Add Execution

Execution Name: Execution 1 Start with: 1 Browser: Chrome Headless Browser

Test Node

AWS Instance

Cloud configuration file is not selected [Select File](#)
Required

Datasets

Use a single dataset for multiple browsers

Default ×

[+ Create New](#) [+ Import New](#)

Cancel **Add**

When you select **AWS Instance** you must select a cloud configuration file that contains a link to the Test Node on which to run the blueprint.

Considerations for Custom Actions and SmartTags

If you are using SmartTags within an AISG execution, there is a limitation for the `PROP` SmartTag when creating Custom Actions. The `PROP` SmartTag is currently designed to ignore the element on the page. This means that an element will not be picked up.



The `PROP` SmartTag is like "null", but it is used in parent SmartTags to create references.

Example

If you have a `PROP` SmartTag and you are trying to create a Custom Action with only that SmartTag, the Custom Action will fail because there are no associated elements. This is a known and expected limitation in AIQ 5.2.x. Any other SmartTag type (Navigation, Action, etc) will work as expected when creating a Custom Action in AIQ 5.2.0.

The workaround is to change the SmartTag to an action (once per app) or remove the SmartTag.



A future release of AIQ will restore the element association and remove this limitation.

AISG Repository Considerations

Git is not recommended for use with AISG. If your AISG executions are extremely large and contains a large amount of images, you may eventually see performance issues if you are using Git. The size limitation for Git is around 100MB.

Once you exceed 100MB, you may want to consider using an S3 repository. This is due to Git not being designed to efficiently handle large amounts of binary files. Images are resource-consuming elements at the disk, memory, and CPU level, so using an S3 repository will yield the best performance. This is because the images will be saved with the script in your S3 repository which simplifies image management.

If you are going to use Visual Accessors, you must use a SFTP, FTP, or S3 repository. Appvance does not recommend using Git to store your images if you are using the Visual Accessor functionality. Again, this is due to Git not being designed to handle large amounts of binary files efficiently.



These guidelines were introduced with the AIQ 5.0 release.
This is a reminder.

Mobile Application Testing

There are major changes and enhancements to the user interface for creating, maintaining and executing mobile tests in the 5.2.0 release of AIQ.

The new Mobil Designer interface aligns perfectly with modern editor capabilities offering a more flexible and intuitive user experience that makes testing tasks easier and more efficient. The new Mobile Designer user interface is similar to the new Web Designer user interface that was introduced in AIQ release 5.0.0.



This document is meant as a quick start guide to help acquaint you with the new Web Designer and explain the basic components in the interface.




Going forward the Mobile Designer (Classic) user interface is frozen. All new features and functionality will only be implemented in the new Mobile Designer interface. Mobile Designer (Classic) will be depreciated and removed from AIQ in a future release.

For more information on the new Mobile Designer see the following sections:

- "Mobile Designer - New User Interface Overview" on the facing page
- "Repository and File Manager in the Mobile Designer" on page 36
- "Mobile Configuration" on page 40
- "Mobile Designer Editor Options" on page 46

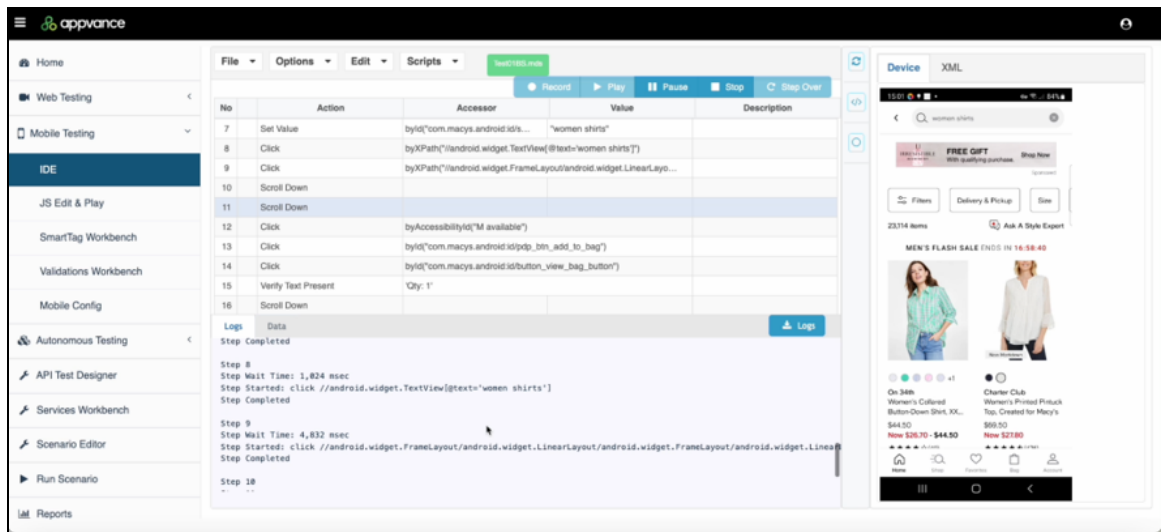
Mobile Designer - New User Interface Overview






See an introductory video showing the new Mobile Designer interface in use.




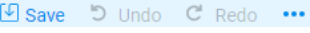
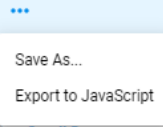


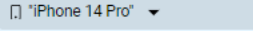
To access the Mobile Designer, from AIQ navigate to **Home > Mobile Testing > Mobile Designer**.







This is an example of the new Mobile Designer interface.



Overview of Mobile Designer Controls and Icons

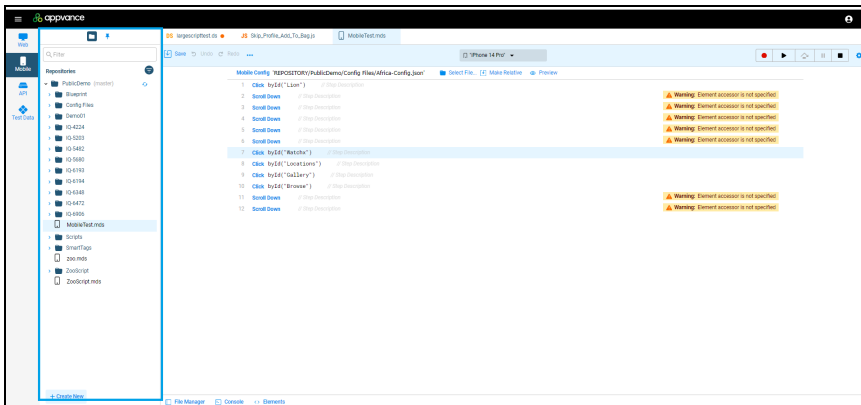
Icon / Control	Function
<p>Repositories </p> <ul style="list-style-type: none"> PublicDemo (master)  Blueprint Config Files Demo01 	<p>Repository pane. Displays the contents of your repository. You can sort and filter using various controls.</p>
	<p>Repository / Quick Access toggle. Switches the File Manager pane between the repository and quick access views.</p>

Icon / Control	Function
	<p>Filter. Allows you to filter the items displayed in the File Manager.</p>
	<p>Sort by Name / Sort by Type. Sorts the contents of the File Manager pane by name or by type.</p>
	<p>Refresh Repository. Refreshes the display of the File Manager pane.</p>
	<p>Editing controls menu.</p> <ul style="list-style-type: none"> • Save • Undo • Redo
	<p>Additional save and export options menu. Lets you</p> <ul style="list-style-type: none"> • Save as • Export to JavaScript
	<p>File Manager toggle. Shows/hides the File Manager pane.</p>
	<p>Console toggle. Shows/hides the Console</p>
	<p>Mobile Device selection control. Controlled by the mobile configuration that you created for your test.</p>

Icon / Control	Function
	See "Mobile Configuration" on page 40 for more information.
	<p>Test controls.</p> <ul style="list-style-type: none"> • Record • Play • Skip Step • Pause • Stop
	Open Test Options menu. See "Mobile Designer Test Options Details" on page 49 for more information.
<p>Mobile Config</p> 	<p>Menu bar for interacting with a Mobile Configuration file.</p> <p> Select File...  Make Relative  Preview</p> <p>You can:</p> <ul style="list-style-type: none"> • Select a configuration file • Make the file path relative • Preview the json configuration file <p>See "Mobile Configuration" on page 40 for more information.</p>

Repository and File Manager in the Mobile Designer

This section details the controls and options available in the Repository and File Manager pane (highlighted) of the Mobile Designer interface.



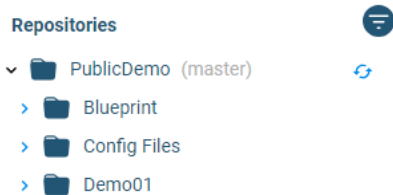
Repository / Quick Access Toggle

Switches the File Manager pane between the repository and quick access views.




Repository Pane

Displays the contents of your repository. You can sort and filter using various controls noted below.



File Type Icons

The following icons display before the file names to indicate the test type.

-  : Mobile test file (.mds)
- **JS** : JavaScript test file (.js)

The following test type icons and files are hidden in the Web Designer because the context is invalid.

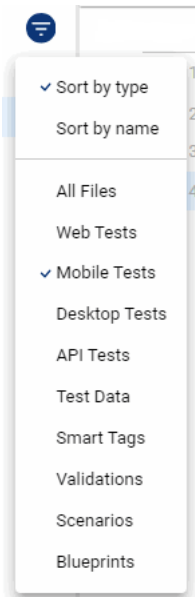
- **DS** : Web test file (.ds)
- **API** : API test file (.api)

Filter

Allows you to filter the items displayed based on your search criteria.

Sort Toggle

Sorts the contents of the Repository File Manager pane by name or by type.



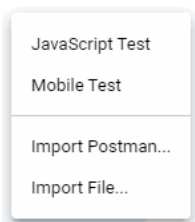
Refresh Repository

Refreshes the display of the File Manager pane.



Create New

Shortcut to create new test files.




Available options:

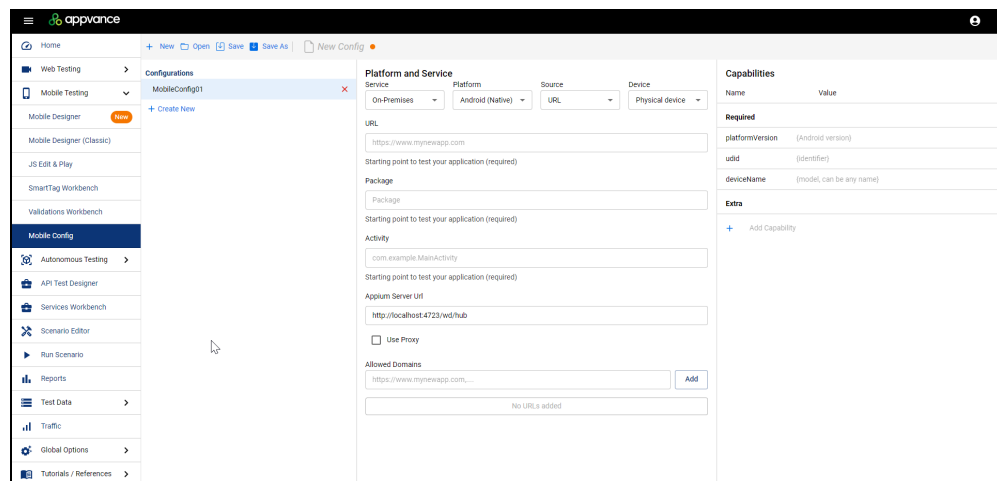
- Mobile Test
- Javascript Test
- Import Postman
- Import File

Mobile Configuration

A mobile configuration file contains all the necessary configuration information that is needed when testing a mobile application.

 This topic describes the Mobile Configuration page in AIQ release 4.9.4 and later.

1. Navigate to AIQ > **Home** > **Mobile Testing** > **Mobile Config**.
 - Click New to begin creating a new mobile configuration file.
 - Click Open to open an existing mobile configuration file.



Platform and Service

Use the **Platform and Service** area of the page to define the basic parameters of your mobile configuration. Some options are only available in certain configurations. The page will update to show you the available options based on what you have selected.

Platform and Service

Service: On-Premises Platform: Android (Native) Source: Package

Package: com.amazon.mShop.andr

Starting point to test your application (required)

Activity: com.amazon.mShop.deta

Starting point to test your application (required)

Allowed Domains: https://www.mynewapp.com,.... Add

No URLs added

Service

Valid values are:

- On-Premises
- BrowserStack
- Sauce Labs
- LambdaTest

Platform

Valid values are:

- Android (Native)
- iOS (Native)

Source

Valid values are:

- Package
- URL

Device

- Physical device
- Emulated device

Start URL

- Starting point to test your application (required)

Bundle ID

- Starting point to test your application. Required.

Package

- Starting point to test your application. Required.

Activity

- Starting point to test your application. Required.

Appium Server URL

- The URL of the Appium server that you are using.

Allowed Domains

- Determines access to URLs that might normally be blocked.

Capabilities

Use the **Capabilities** area of the page to define the specific parameters of your mobile configuration. Some options are only available in certain configurations. The page will update to show you the available options based on what you have selected.

Capabilities	
Name	Value
Required	
platformVersion	33
udid	emulator-5554
deviceName	device
Extra	
noReset	true ▼ ✕
+ Add Capability	

On-Premises Capabilities

- platformVersion - Version of Android or iOS installed on the device.
- udid
- deviceName

- `bundleId` - iOS only. Application bundle ID. Application must be installed if the value for `app capability` is blank.
- `xcodeOrgId` - iOS only. Team ID.
- `xcodeSigningId` - iOS only. iPhone developer.
- `updatedWDABundleId` - iOS only.

BrowserStack Capabilities

- `browserstack.appium_version`
- `os_version` - Version of Android or iOS installed on the device
- `device` - Model of the mobile device
- `realMobile` - True/False
- `project`
- `build`
- `name`
- `app`

SauceLabs Capabilities

- `platformVersion` - Version of Android or iOS installed on the device.
- `deviceName` - Model of the mobile device.
- `deviceOrientation` - Orientation of the device. Portrait or Landscape.
- `app` - URL of the Android Package Kit (apk).

LambdaTest Capabilities

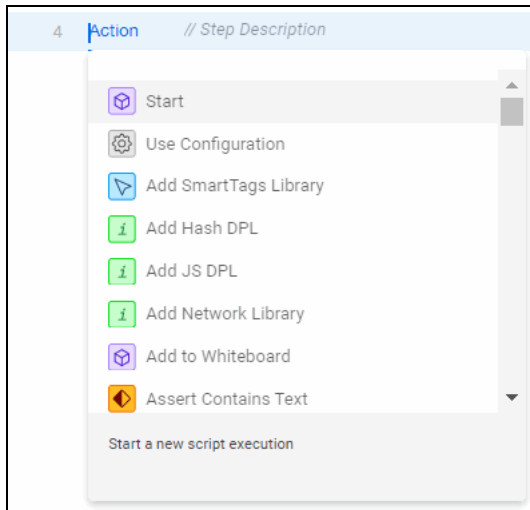
- platformVersion - Version of Android or iOS installed on the device.
- deviceName - Model of the mobile device.
- deviceOrientation - Orientation of the device. Portrait or Landscape.
- app - URL of the Android Package Kit (apk).
- isRealMobile - Real device (True) or emulated device (False)
- noReset - Determines if the app state will be reset the session.

Mobile Designer Editor Options

The following options are available in the Mobile Designer.

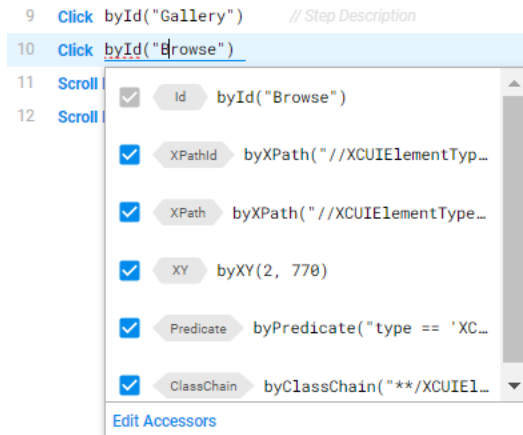
Available Actions

The auto assist feature will show you the list of available actions.



Edit Accessors

- Right-click with the accessor column to see the list of accessors for that step. You can use the individual blue check boxes to toggle individual accessors on and off.
- Click Edit Accessors to edit multiple accessors at once.



Right Click Menu

The following options are available from the right click menu in the test script editor. Your cursor needs to be in the right most column of the editor to access the right click menu.

- Show Quick Help
- Copy
- Cut
- Paste
- Extract to Included file
- Add Breakpoint
- Record From Here
- Add Step Below
- Add Step Above
- Add Description
- Disable Step
- Delete

Show Quick Help




Extract to Included Files

You can extract the step to a separate test script.

Add Description

You can add descriptions for any and all steps in your test script. This lets you document the important details such as a specific search operation or a login details. Script descriptions are extremely useful for Appvance Support if they are called upon to help with script debugging.



 In all AIQ reports, descriptions will be replaced as the step names.

Show / Hide Descriptions

When a line of the script is highlighted, you can click the Description icon to toggle the display of comments for that line.

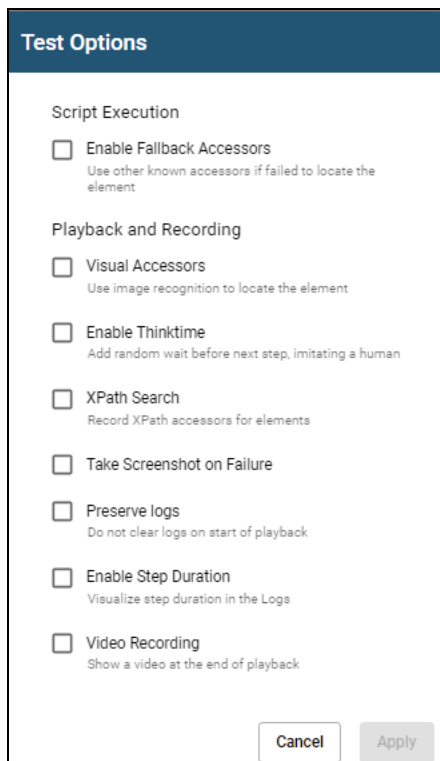


Removing Descriptions

When a line of the script that has a description is highlighted, there is a **Remove Description** option available from the right-click menu. You can also click the three dots to the right of the Description icon and select **Remove Description** from the menu.

Mobile Designer Test Options Details

Use the **Test Options** screen to configure various options for your test script.



Test Options

Script Execution

- Enable Fallback Accessors
Use other known accessors if failed to locate the element

Playback and Recording

- Visual Accessors
Use image recognition to locate the element
- Enable Thinktime
Add random wait before next step, imitating a human
- XPath Search
Record XPath accessors for elements
- Take Screenshot on Failure
- Preserve logs
Do not clear logs on start of playback
- Enable Step Duration
Visualize step duration in the Logs
- Video Recording
Show a video at the end of playback

Cancel Apply



See "Mobile Designer Test Options Comparison" on page 52 for the differences in test options between Mobile Designer and Mobile Designer (Classic).

Script Execution Options

Enable Fallback Accessors

You can enable the Fallback Accessors option so that AIQ keeps test running if an accessor fails or has changed, by automatically choosing other available accessors.

- If Fallback Accessors are enabled, AIQ will try one time for each accessor. If an accessor fails then it will scroll down and again try again each accessor one time and scroll until it reaches the end of the page or reaches maximum number scrolls.
- The scrolling logic will scroll up to ten times, depending on what it locates. If it is not locating elements, it will stop before it reaches ten. If it is locating elements, it will continue until ten.
- If Fallback Accessors disabled, playback will function as it currently does. Tests may fail if an accessor fails or has changed.

Playback and Recording Options

Visual Accessors

Visual Accessors are image based accessors which can be used to enhance your test scripts in situations where traditional code-based accessors will fail.

AIQ will capture an image of the website and then crop the surrounding boundaries of the element that you are working on. During playback it takes the image and applies a feature based matching algorithm which creates a pattern from the image and uses that to locate the image on the webpage.

Using a Visual Accessors means that AIQ is not only looking at just the code elements in a page, the AI is also looking at the visual elements on the page.



See Visual Accessors for Mobile Testing for more detailed usage information.

Enable Thinktime

- Think time options can be enabled or disabled depending on the requirements. Enabling think times will add wait for statements in between every action performed from the designer script.

XPath Search

- There are several ways to identify an element on the webpage, XPath search is one among them. When you record a use case with this option enabled, under the Accessor list you will notice the XPath accessors of that particular element as well. Disabling this option will not add any Xpath references to that element being identified.

Take Screenshot on Failure

Determines if a screenshot is taken even if the script fails before we reach that step where a snapshot was to be captured..

Preserve Logs

Determines if the logs should be cleared every time a script is played. This option is disabled by default.

Enable Step Duration

Determines if step durations are noted in the logs.

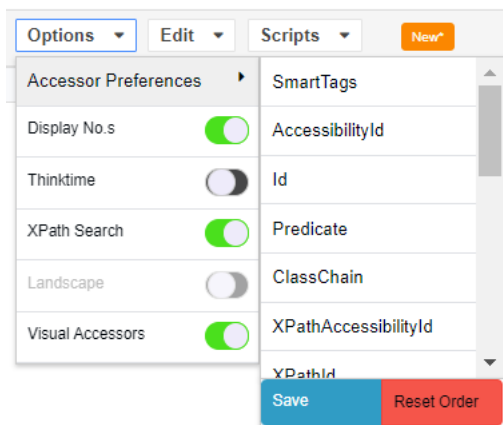
Video Recording

Determines if a video of the script is shown at the end of the script playback.

Mobile Designer Test Options Comparison

The implementation of Test Options is different between the Mobile Designer (Classic) and new Mobile Designer interfaces.

- Test Options available in Mobile Designer (Classic)



- Test Options available in the new Mobile Designer implementation

Test Options

Script Execution

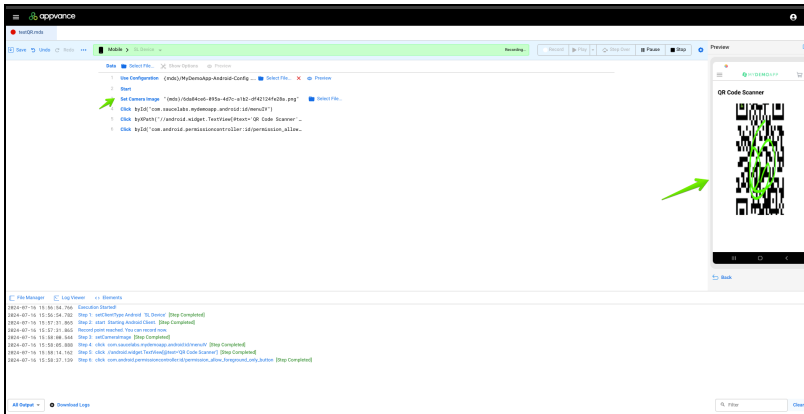
- Enable Fallback Accessors**
Use other known accessors if failed to locate the element

Playback and Recording

- Visual Accessors**
Use image recognition to locate the element
- Enable Thinktime**
Add random wait before next step, imitating a human
- XPath Search**
Record XPath accessors for elements
- Take Screenshot on Failure**
- Preserve logs**
Do not clear logs on start of playback
- Enable Step Duration**
Visualize step duration in the Logs
- Video Recording**
Show a video at the end of playback

Image Injection

The `Set Camera Image` function allows you to set an image to be used when triggering the camera during a test. For example, you could use this to inject a QR code into your test script.



Compatibility Summary

Service	iOS		Android		Requirements
	Device	Simulator	Device	Simulator	
On Premises	Not supported	Not supported	Not supported	Supported	Update emulator settings.
Sauce Labs	Supported	Not supported	Supported	Not supported	Add capability: <code>sauceLabsImageInjectionEnabled: true</code>
Browser-Stack	Supported	Not supported	Supported	Not supported	Add capability: <code>enableCameraImageInjection: true</code>

Lambda-Test	Supported	Not supported	Supported	Not supported	Add capability: enableImageInjection: true
-------------	-----------	---------------	-----------	---------------	--

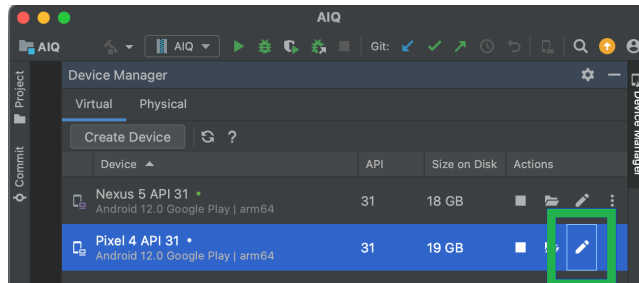
On-Premises

Available for Android emulators only when running AIQ locally.

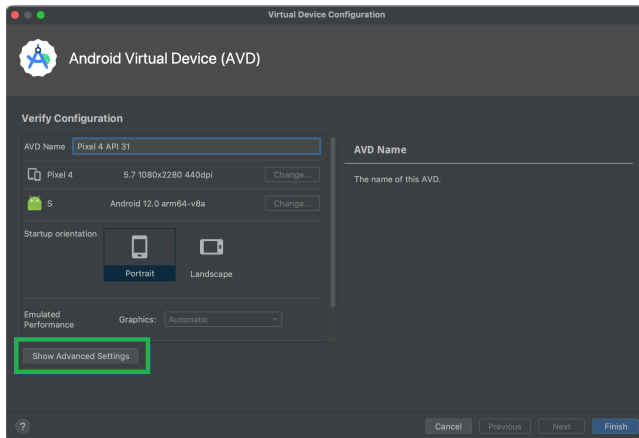
When running AIQ on an EC2 we don't have access to `JAVA_HOME` environmental variable nor the local file system, hence we can't override the emulator settings.

Android Emulator Setup

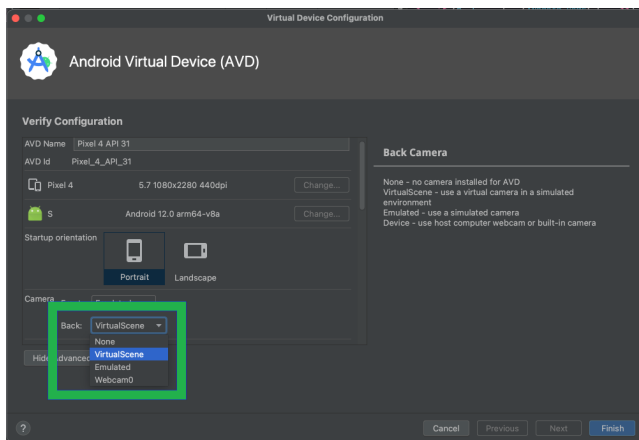
1. From Android Studio open the Device Manager.
2. Click the **Edit** button for the emulator that you want to use.



3. In the **Virtual Device Configuration** window, click **Show Advanced Settings**.



4. In the **Camera** section select **VirtualScene** from the drop down.



5. Click **Finish**.

Sauce Labs

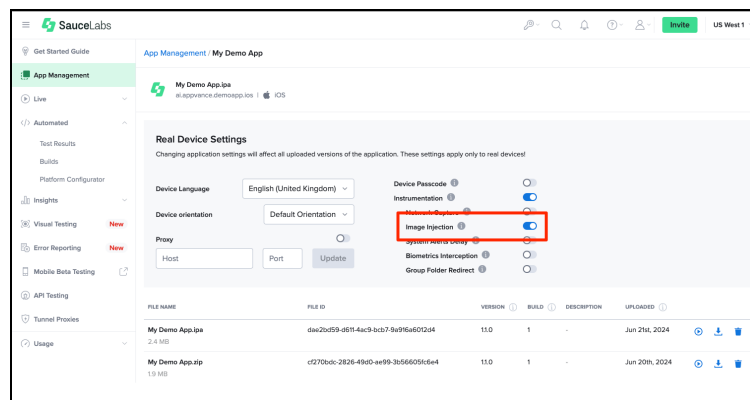
Available for Android and iOS physical devices only. There's support for native apps only.

The config file's capabilities need to include the following field:

```
"sauceLabsImageInjectionEnabled": "true"
```

Sauce Labs' App Config

1. Login into your Sauce Labs account.
2. Go to App Management.
3. Click the Settings button for the app you want to use.
4. Turn the **Image Injection** switch on.



Supported APIs

Android System Camera

For Android devices, there are multiple ways to capture an image, as described in the [Android Camera API](#) developer documentation. AIQ supports the following:

- `ACTION_IMAGE_CAPTURE` Opens the system camera and notifies the calling app when the image is taken.
- camera2 API: Everything is configured and handled from within the app.

- cameraX: Leverages the capabilities of `camera2`, but uses a simpler, use case-based approach that is lifecycle-aware.
- Camera API (deprecated): partially supported. As with `camera2`, everything is handled in the app itself. QR Code readers often use `Camera#setPreviewCallback`. We pass the injected image to this method, but the rest of this deprecated API is not supported. UI Elements will not likely display the injected image.

iOS System Camera

For iOS devices, the camera can be configured with different outputs.

AIQ supports the following:

- `AVCapturePhotoOutput`: for capturing still images. The results are received via the `AVCapturePhotoCaptureDelegate` and the method `captureOutput:didFinishProcessingPhoto:error:`. The other methods in this delegate are either deprecated or handle live photos, which we don't support.
- `AVCaptureVideoDataOutput`: for capturing video frames and processing them. The frames are received via `AVCaptureVideoDataOutputSampleBufferDelegate` and the method `captureOutput:didOutputSampleBuffer:fromConnection:`.
- `AVCaptureMetadataOutput` for reading QR-Codes. The QR Codes are passed to the app via `captureOutput:didOutputMetadataObjects:fromConnection:`. We are detecting the `AVMetadataMachineReadableCodeObject` and QR Codes are part of that.



Sauce Labs documentation is available [here](#).

BrowserStack

The config file's capabilities need to include the following field:

```
"browserstack.enableCameraImageInjection": "true"
```

Supported APIs

Camera image injection is supported for the following Camera APIs. Check with your app development team to get more details about the Camera APIs your app uses.

OS	OS version	Supported APIs
iOS	iOS 13 and above	<p>didFinishPickingMediaWithInfo API of UIImagePickerController iOS SDK class for capturing an image. For more details, please check Apple documentation.</p> <p>AVCapturePhoto iOS SDK class for receiving captured photos from AVCapturePhotoOutput API. For more details, please check Apple Documentation.</p> <p>jpegStillImageNSDataRepresentation of AVCaptureStillImageOutput iOS SDK class. For more details, please check Apple documentation.</p>
Android	Android 6 and above	<p>android.hardware.camera.PreviewCallback API for API level 1-20</p> <p>android.hardware.camera2.CameraManager, android.hardware.camera2.CameraDevice, android.hardware.camera2.CameraCharacteristics, android.hardware.camera2.CameraCaptureSession, android.hardware.camera2.CaptureRequest, android.media.ImageReader APIs for API level >=21</p>

Camera Intent API



Browser Stack full documentation [here](#).

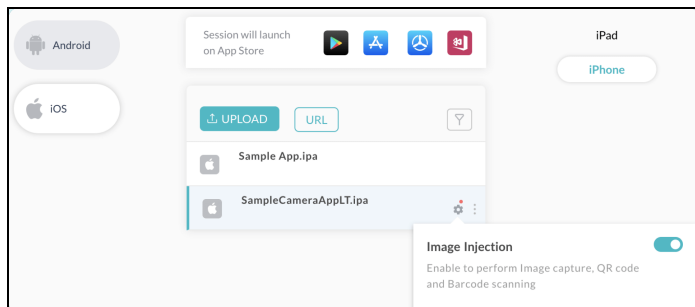
LambdaTest

The config file's capabilities need to include the following field:

```
"enableImageInjection": "true"
```

LambdaTest App Setup

1. Login to your LambdaTest account.
2. Go to App Testing under Real Devices.
3. Hover the Settings icon and toggle the Image Injection switch on.



Supported APIs

Below given is the list of Camera APIs which are supported by LambdaTest Image Injection tool. Please ensure that your app uses these APIs only to enable Image Injection.

OS	Devices	Supported APIs
iOS	iOS 13 and higher	didFinishPickingMediaWithInfo API of UIImagePickerController iOS SDK class for capturing an image. For details, please check Apple documentation .
		AVCapturePhoto iOS SDK class for receiving cap-

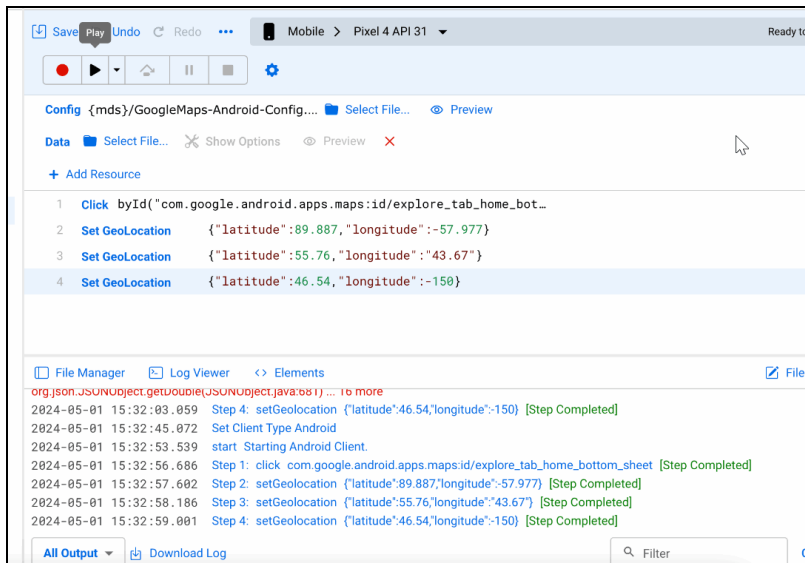
		tured photos from AVCapturePhotoOutput API. For details, please check Apple Documentation .
Android	Android 9 and higher	CameraX API is used. For details, please check CameraX Documentation
		Camera API is used. For details, please check this link .
		Camera2 API is used. For details, please check this link .
		ACTION_IMAGE_CAPTURE is standard Intent action that can be sent to have the camera application capture an image and return it. For details please check this link .



LambdaTest full documentation [here](#).

Geo Location

You can use Geo Location for iOS and Android phones with Browser Stack, SauceLabs, and On Premise testing. The Geo Location format is latitude and longitude in decimal format.

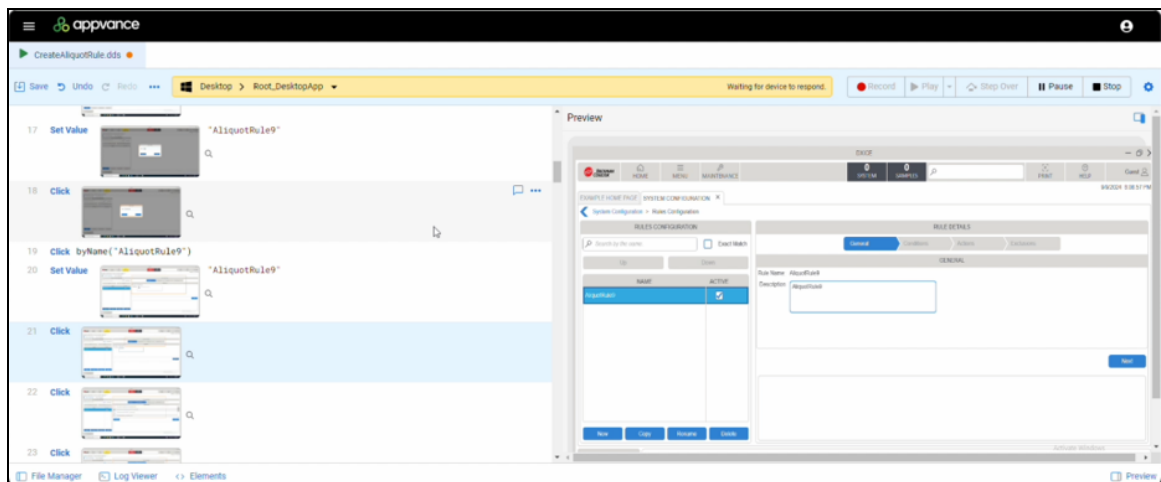


Desktop Application Testing

Desktop Designer is a new testing interface that takes advantage of AIQ's Mobile Designer technology so that you can test desktop applications across diverse environments, including Citrix. You can use Visual Accessors to create new test scripts using simple as point-and-click. Fallback accessors and self-healing script capabilities are built-in to help make creating robust and resilient test scripts effortless.



Here is a brief introductory video on the features and benefit of Desktop Designer.



Benefits and Highlights

- You can your test scripts as a Scenarios and view the results on the Dashboard.
- Record a video of the test script execution.
- Easy access to all Windows keys and actions.

Desktop Application Testing

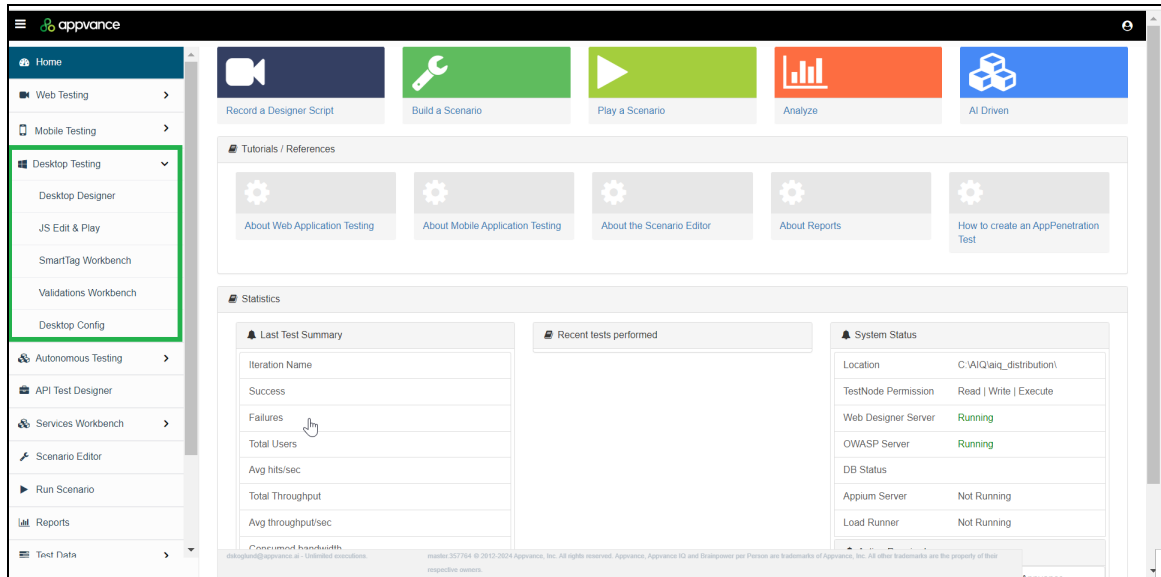
- Point and shoot to record your script or add the logic you need with the use of Javascript commands
- Detachable preview mode allows you to work on two monitors. You can view your test script on one monitor and the application on another monitor.



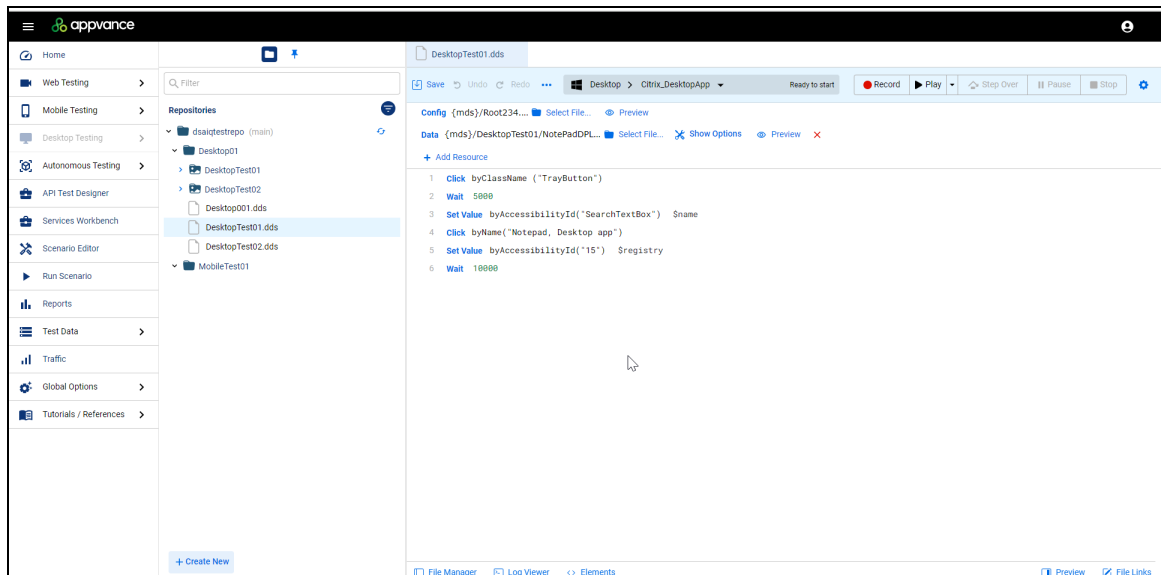
Desktop Designer currently supports Windows applications and mainframe systems. Support for Desktop Testing on other operating systems will be added in future releases of AIQ.

Desktop Designer User Interface Overview

To access the new Desktop Designer, from AIQ navigate to **Home > Desktop Testing > Desktop Designer**.

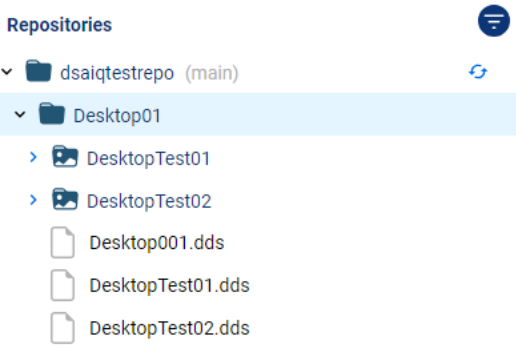




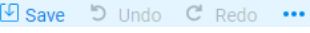


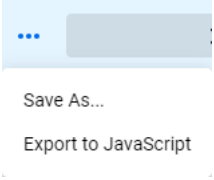


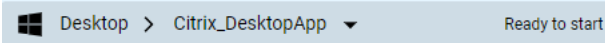



This is an example of the new Desktop Designer interface.




Overview of Desktop Designer Controls and Icons

The following table provides an introduction of the various controls and icons in the new Desktop Designer interface. The various elements are controls are detailed in the following pages of this document.

Icon / Control	Function
	<p>Repository pane. Displays the contents of your repository. You can sort and filter using various controls.</p>
	<p>Repository / Quick Access toggle. Switches the File Manager pane between the repository and quick access views.</p>
	<p>Filter. Allows you to filter the items displayed in the File Manager.</p>
	<p>Sort by Name / Sort by Type. Sorts the contents of the File Manager pane by name or by type.</p>
	<p>Refresh Repository. Refreshes the display of the File Manager pane.</p>
	<p>Editing controls menu.</p> <ul style="list-style-type: none"> • Save • Undo

Icon / Control	Function
	<ul style="list-style-type: none"> • Redo
	<p>Additional save and export options menu.</p> <ul style="list-style-type: none"> • Save as • Export to JavaScript. See "Desktop Designer Export to Javascript" on page 78 for an example.
	<p>File Manager toggle. Shows/hides the File Manager pane.</p>
	<p>Log Viewer toggle. Shows/hides the logs.</p>
	<p>Browser selection control. See Selecting the Browser for more information.</p>
	<p>Test controls:</p> <ul style="list-style-type: none"> • Record • Play • Skip Step • Pause • Stop
	<p>Opens the Test Options menu. See Desktop Designer Test Options for more information.</p>
	<p>Select Configuration controls. You must have a valid desktop configuration file selected</p>

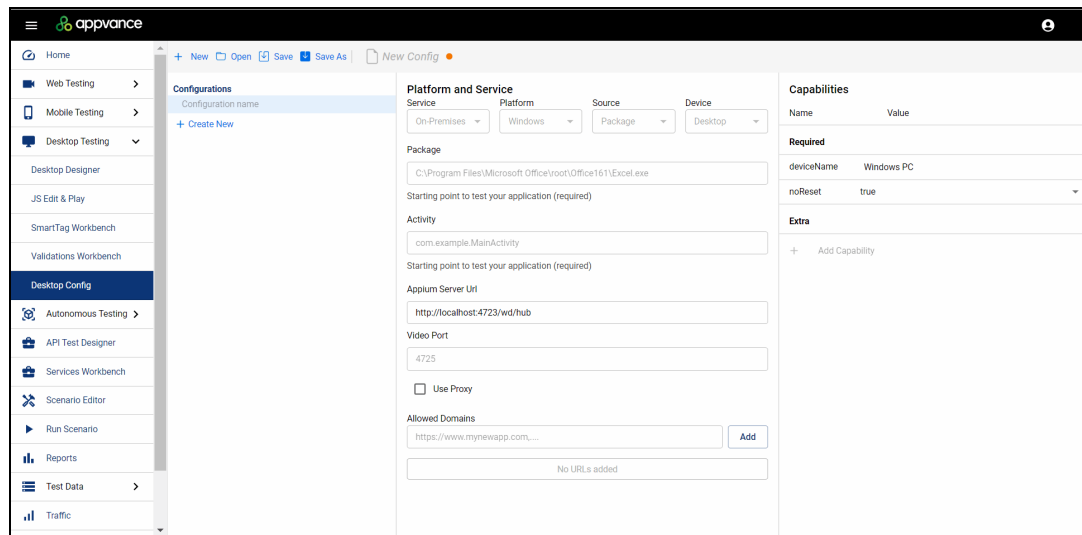
Icon / Control	Function
	before you can begin a desktop test. See "Desktop Designer Configuration" on the facing page for more information.
	Select data controls. You can attach test data to your desktop test.

Desktop Designer Configuration

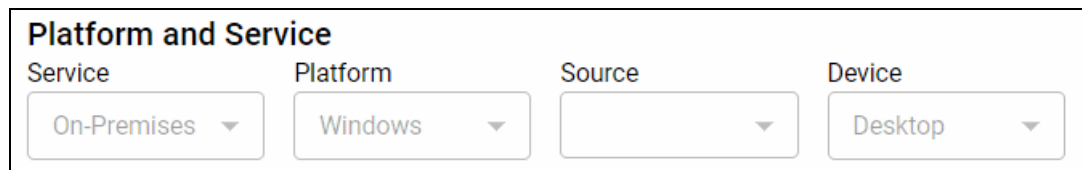
All desktop tests must have a desktop configuration file associated with them. This file contains all the necessary settings in order for the test to run. Configuration files are assigned to desktop tests in the Desktop Designer. A desktop configuration file can be associated with multiple desktop test files.

Create or edit a desktop testing configuration file

1. From AIQ navigate to **Desktop Testing > Desktop Config.**

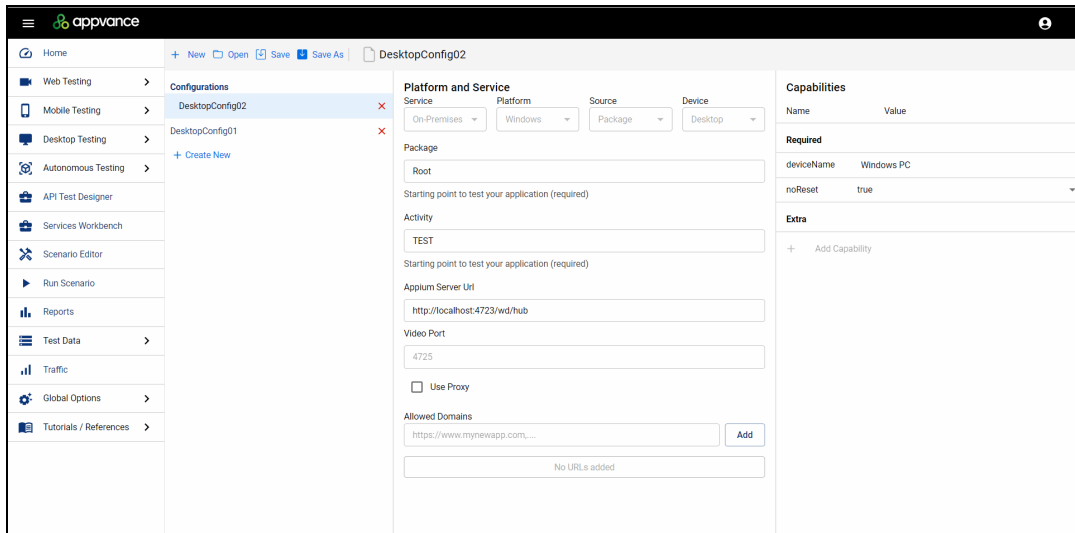


2. Click **Create New** and name the file.
3. The parameters for **Service, Platform, Source** and **Device** are set by default.

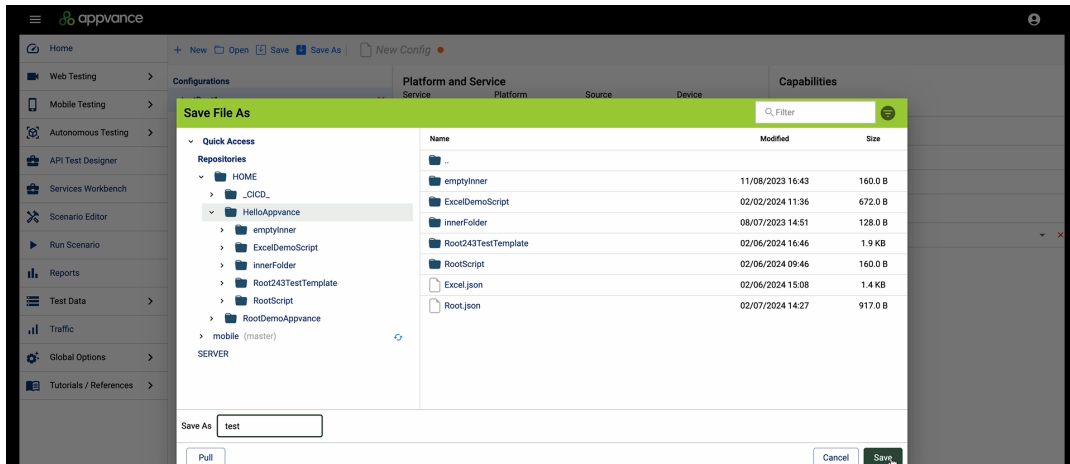


Desktop Application Testing

- Specify the required parameters.
 - Package** specify **Root**
 - Activity** specify **TEST**
 - Appium Server URL** specify the IP address of the desktop machine you will be testing



- Click **Save** to save your desktop configuration file to your repository.



Example Desktop Configuration File

```
[
  {
    "configName": "Root_DesktopApp",
    "app": "",
    "appPackage": "Root",
    "bundleId": "",
    "source": "package",
    "deviceMode": true,
    "service": "On-Premises",
    "serverURL": "http://100.20.219.14:4725/wd/hub/",
    "platformName": "Windows",
    "appActivity": "TEST",
    "validDomains": "",
    "capabilities": {
      "systemPort": "8201",
      "deviceName": "WindowsPC",
      "noReset": "true"
    }
  }
]
```



This is the preview of the configuration file that you can view in the Desktop Designer user interface.



See [Mobile Configuration](#) for more information about Mobile Configuration files.

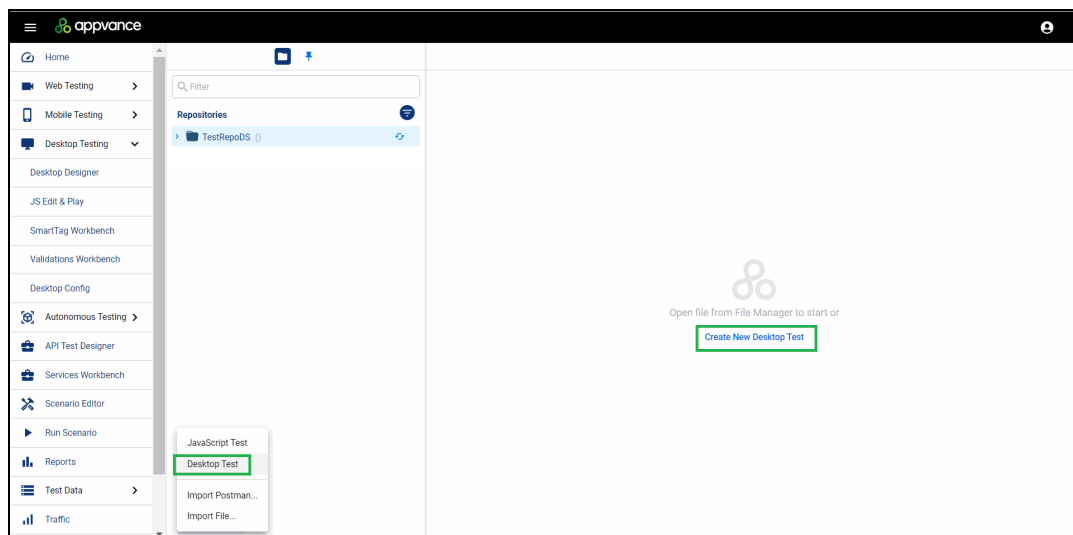
Creating a Desktop Test

Here are the basic steps for creating a new desktop test file in the Desktop Designer.

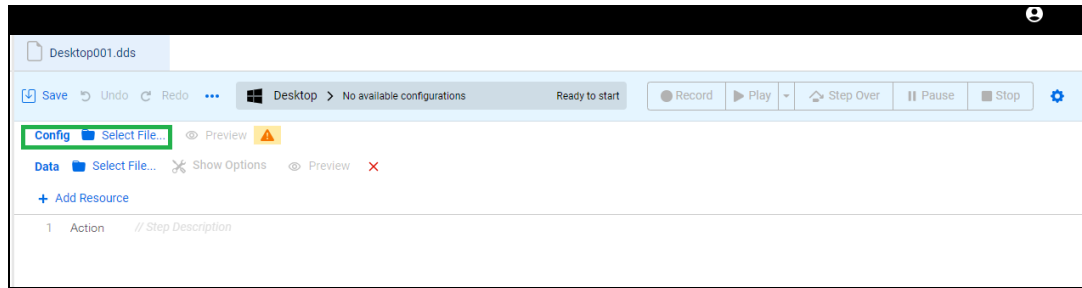


Prerequisite: A desktop configuration file must be created before you create the test file. See "Desktop Designer Configuration" on page 69 for more information.

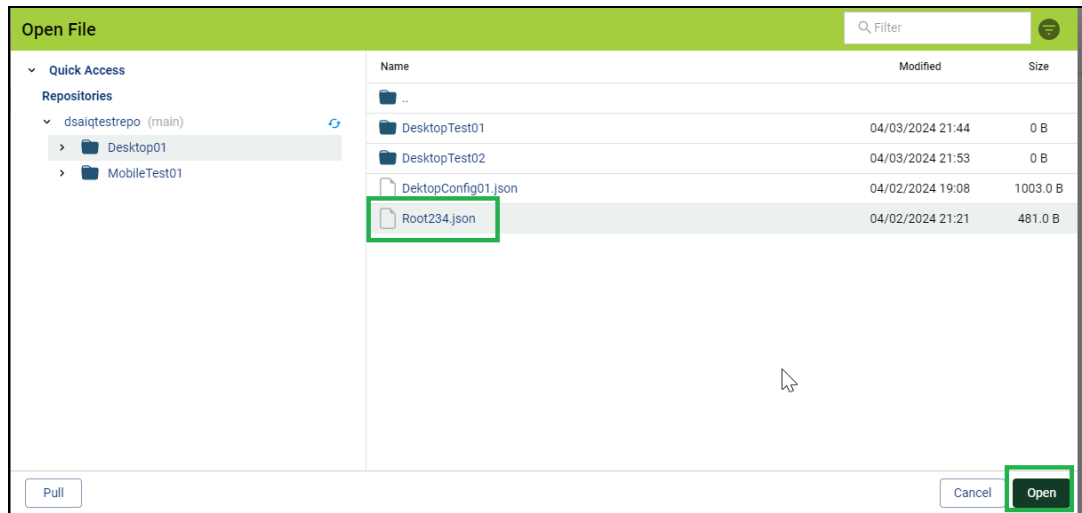
1. In AIQ navigate to **Desktop Testing > Desktop Designer**.
2. Click **Create New** in the and select **Desktop Test**. You can also click **Create New Desktop Test** in the main portion of the screen.



3. Name the test file.
4. On the **Config** line, click **Select File** to add a desktop testing configuration to your test. This needs to be a preexisting file. See "Desktop Designer Configuration" on page 69 for more information.



5. Browse to the location of your configuration file in your repository and click **Open**. Configuration files are `.json` files.



6. You can click **Preview** to verify the details of the configuration file.



7. You can now begin recording your test. Click **Record** to begin.



See "Desktop Designer Example - Kindle Desktop App" on page 80 for an example desktop test.

Desktop Designer Test Options

The following options are available for desktop testing in the Desktop Designer.

Test Options

Script Execution

- Enable Fallback Accessors
Use other known accessors if failed to locate the element

Playback and Recording

- Visual Accessors
Use image recognition to locate the element
- Enable Thinktime
Add random wait before next step, imitating a human
- XPath Search
Record XPath accessors for elements
- Take Screenshot on Failure
- Preserve logs
Do not clear logs on start of playback
- Enable Step Duration
Visualize step duration in the Logs
- Video Recording
Show a video at the end of playback

Cancel Apply

Script Execution Options

The following options pertain to execution of scripts.

Enable Fallback Accessors

You can enable the Fallback Accessors option so that AIQ keeps test running if an accessor fails or has changed, by automatically choosing other available accessors.

- If Fallback Accessors are enabled, AIQ will try one time for each accessor. If an accessor fails then it will scroll down and again try again each accessor one time and scroll until it reaches the end of the page or reaches maximum number scrolls.
- The scrolling logic will scroll up to ten times, depending on what it locates. If it is not locating elements, it will stop before it reaches ten. If it is locating elements, it will continue until ten.
- If Fallback Accessors are disabled, playback will function as it currently does. Tests may fail if an accessor fails or has changed.

Playback and Recording Options

The following options pertain to playback and recording of scripts.

Visual Accessors

Visual Accessors are image based accessors which can be used to enhance your test scripts in situations where traditional code-based accessors will fail. AIQ will capture an image of the website and then crop the surrounding boundaries of the element that you are working on. During playback it takes the image and applies a feature based matching algorithm which creates a pattern from the image and uses that to locate the image on the webpage.

Using a Visual Accessors means that AIQ is not only looking at just the code elements in a page, the AI is also looking at the visual elements on the page.



See [Visual Accessors for Web Testing](#) for more detailed usage information.

Enable Thinktime

Think time options can be enabled or disabled depending on the requirements. Enabling think times will add wait for statements in between every action performed from the designer script.

XPath Search

There are several ways to identify an element on the webpage, XPath search is one among them. When you record a use case with this option enabled, under the Accessor list you will notice the XPath accessors of that particular element as well. Disabling this option will not add any Xpath references to that element being identified.

Take Screenshot on Failure

Determines if a screenshot is taken even if the script fails before we reach that step where a snapshot was to be captured..

Enable Step Time Duration

Visualize step durations in the log files.

Preserve Logs

Determines if the logs should be cleared every time a script is played. This option is disabled by default.

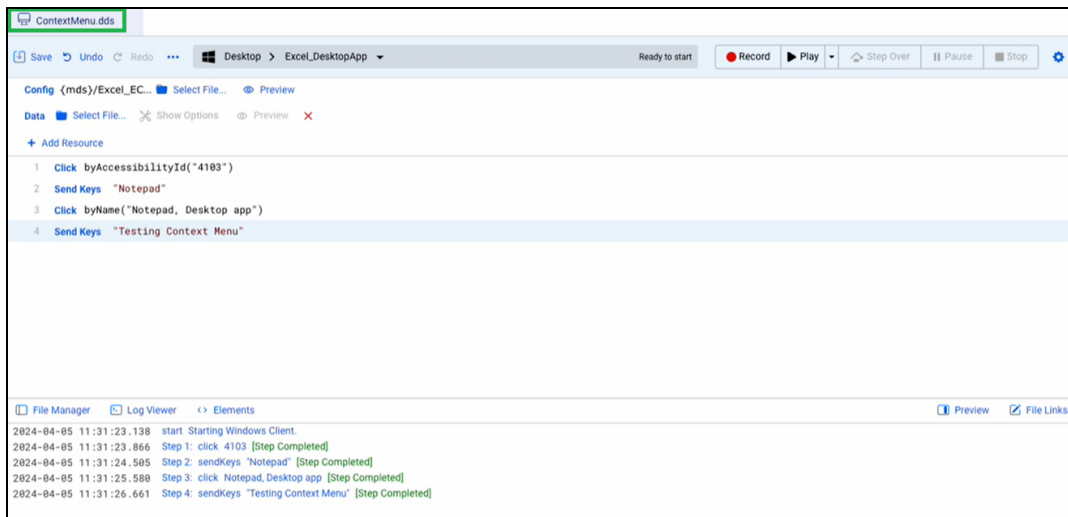
Video Recording

If enabled, it will show a video at the end of the playback.

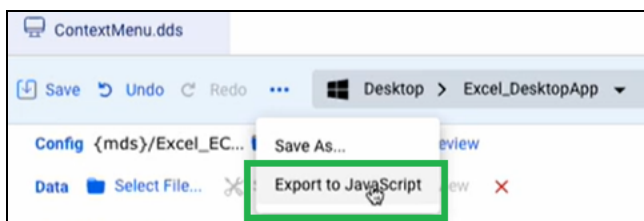
Desktop Designer Export to Javascript

You can export a desktop designer test file to javascript. This will make your test script available for web testing. The javascript file is created with the same name as your desktop designer test file.

1. For example, you have a desktop testing file named **ContextMenu.dds**.



2. Click the additional options menu (...) and select **Export to Javascript**.



3. The file **ContextMenu.js** is created.

A screenshot of a code editor window. The title bar shows 'ContextMenu.ddd' and 'JS ContextMenu.js'. The editor contains the following JavaScript code:

```
1 setVariablesIfNeeeded('', 'HashDPL', 0);
2 setConfiguration!({nds}/ExceL_EC2.json', 'ExceL_DesktopApp');
3 start();
4 click(byAccessibilityId("4103"));
5 sendKeys("Notepad");
6 click(byName("Notepad, Desktop app"));
7 sendKeys("Testing Context Menu");
8
```

Desktop Designer Example - Kindle Desktop App

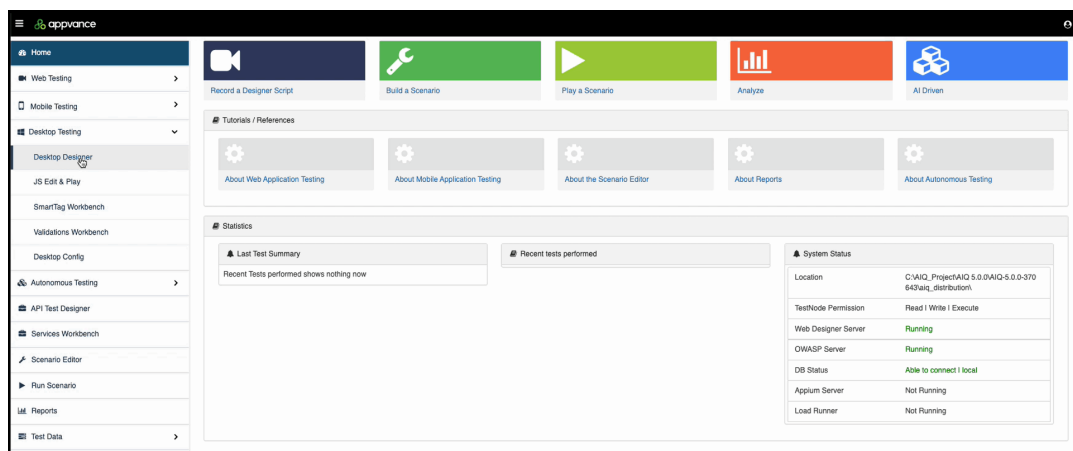
Here is a basic example of creating a desktop test that interacts with the Kindle desktop application.

This example will show:

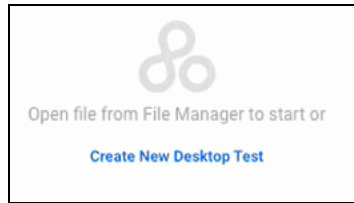
- Creating a new Desktop Designer test script
- Starting the recording
- Launching the desktop application being tested
- Interacting with the desktop application
- Creating Visual Accessors
- Creating an assert
- Exiting the application
- Editing the test script
- Playing back the recording of the test script

Example

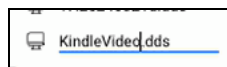
1. In AIQ navigate to **Desktop Testing > Desktop Designer**.




2. Click **Create New Desktop Test**.



3. Name and save your new test script. In this example the file is named `KindleVideo.dds`.



 `.dds` is the file extension for all Desktop Designer test scripts.

4. Select a Desktop Configuration file. If you have not created a Desktop Configuration file, see Desktop Designer Configuration Files for more information.
5. Click **Preview** to see your Desktop Configuration file.

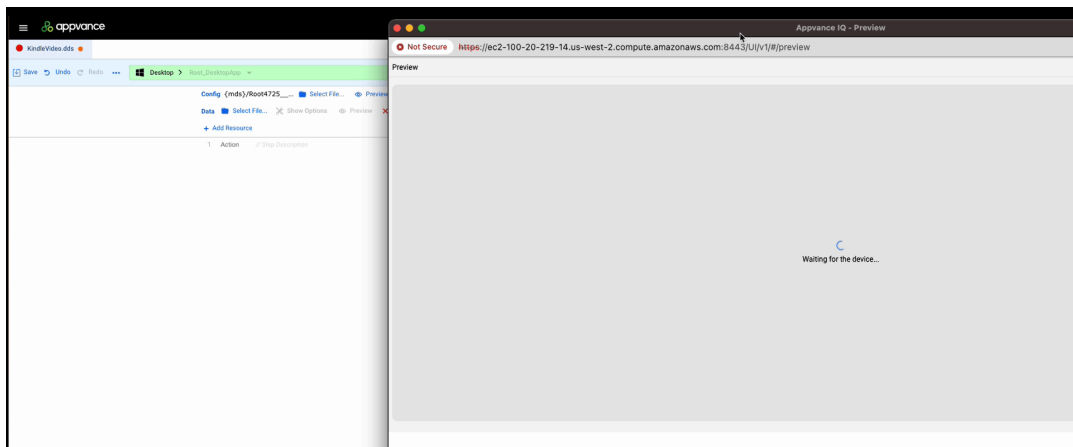


6. Click **Record** to begin recording your test script.

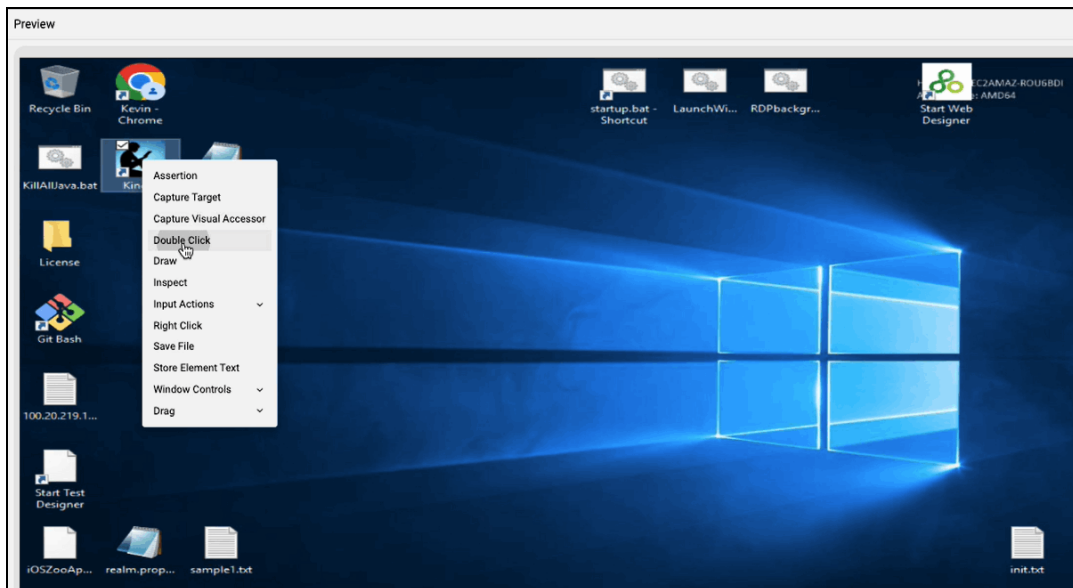
Desktop Application Testing



7. Wait for the device to load.



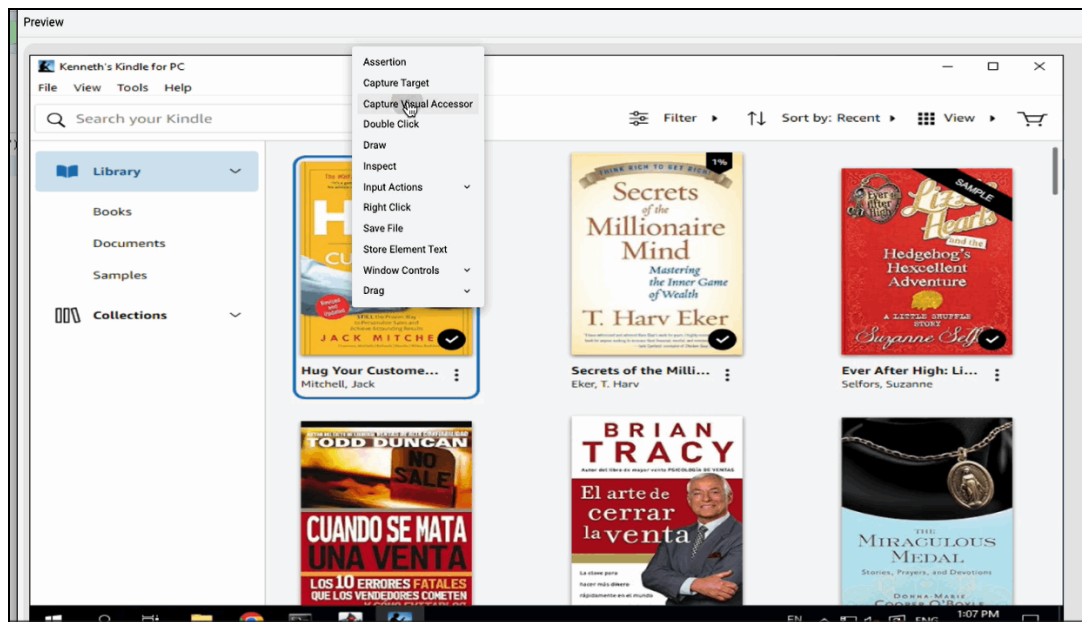
8. Right click the desktop icon for the Kindle app and select **Double Click**.



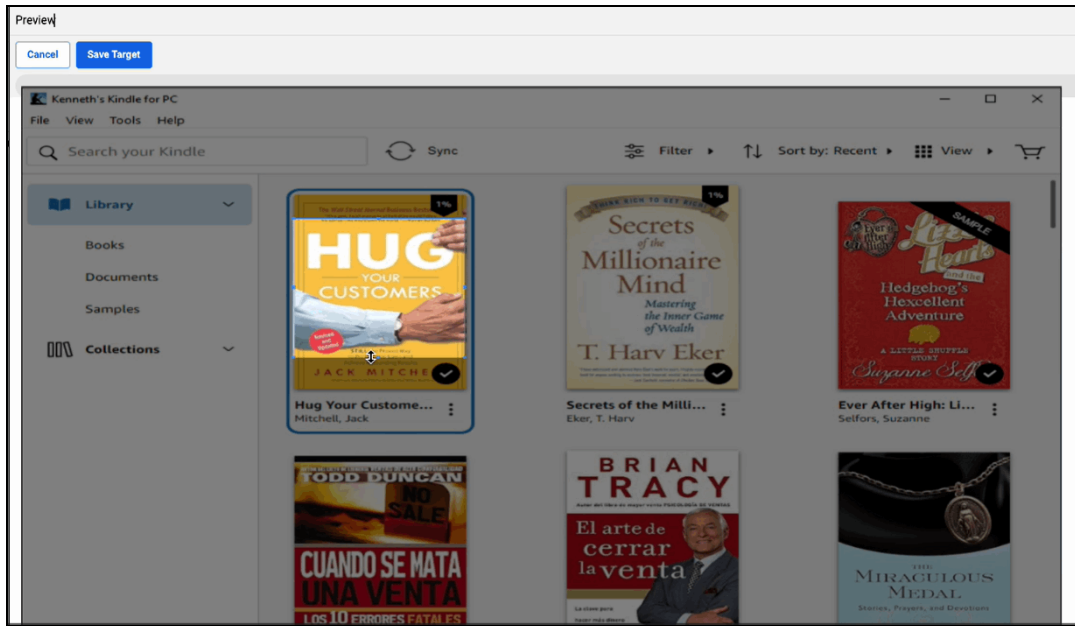
9. The Kindle desktop app starts.



10. Hover over one of the book covers. Right click and select **Capture Visual Accessor**.

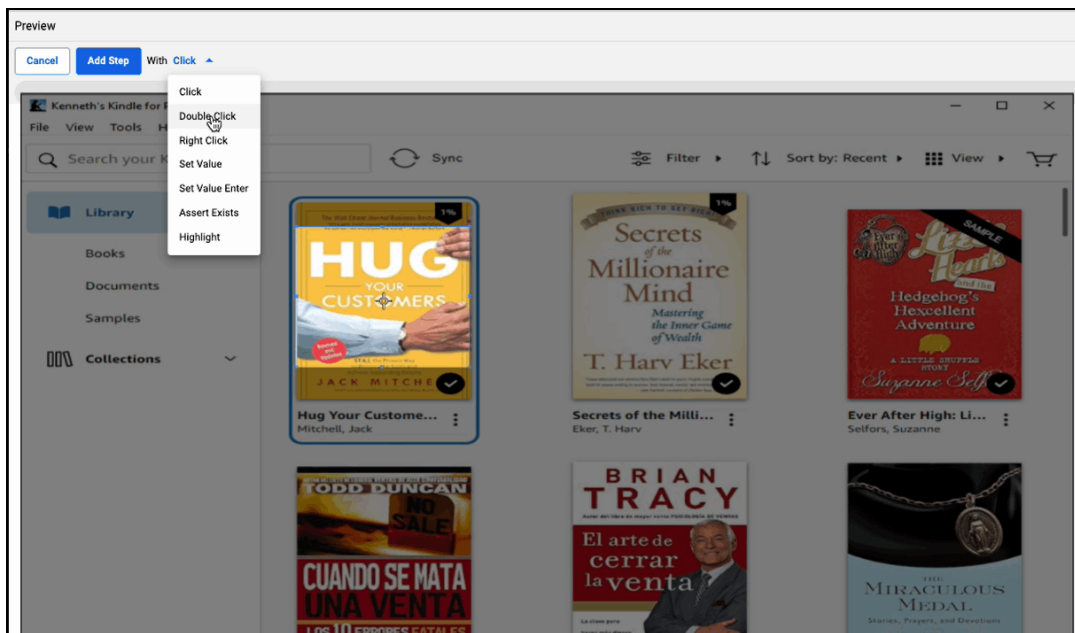


11. Select the boundaries of the Visual Accessor.
12. Click **Save Target**.

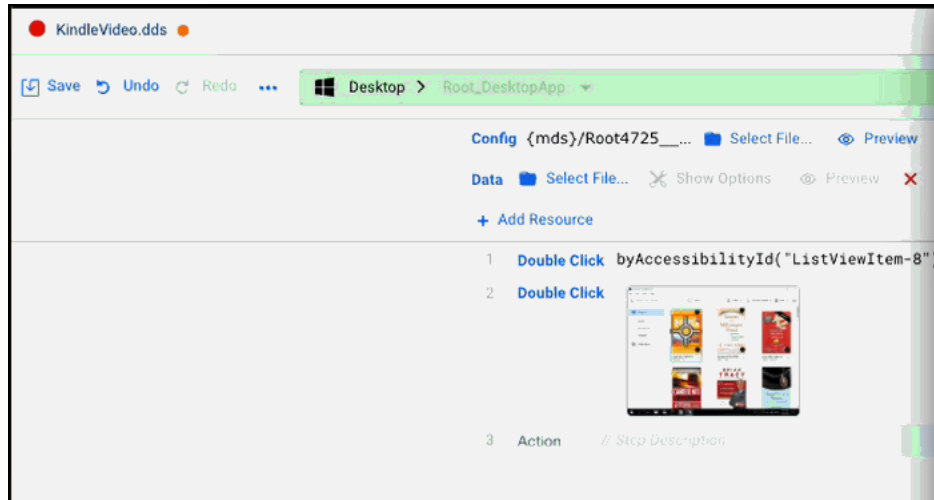


13. Click on the Visual Accessor.

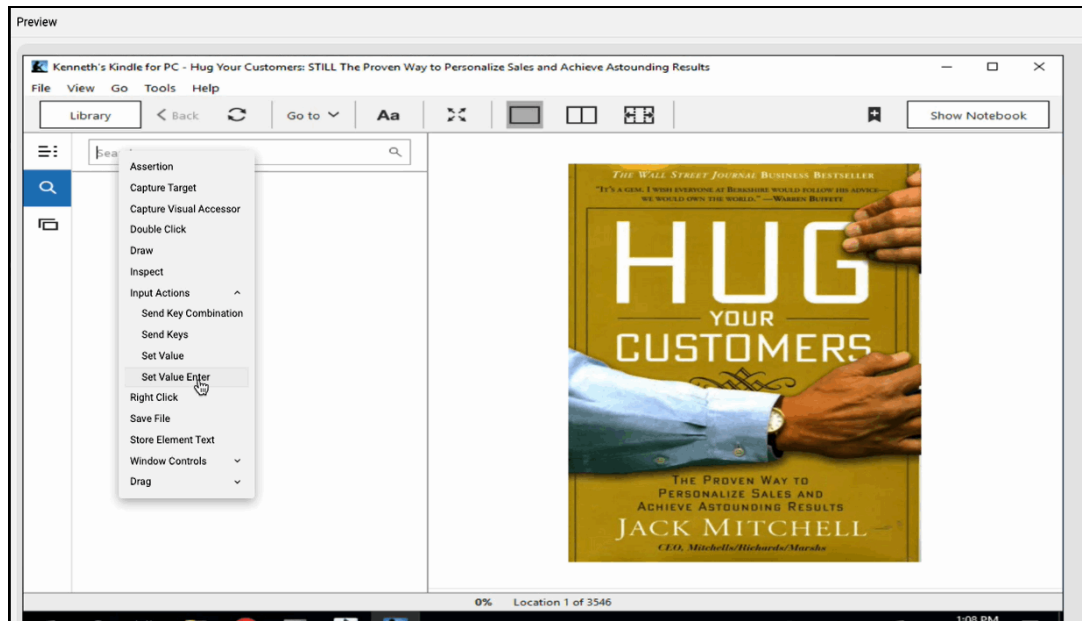
14. **Add Step with Double Click.**



- In the Desktop Designer window you can see that the step has been added to your test script.

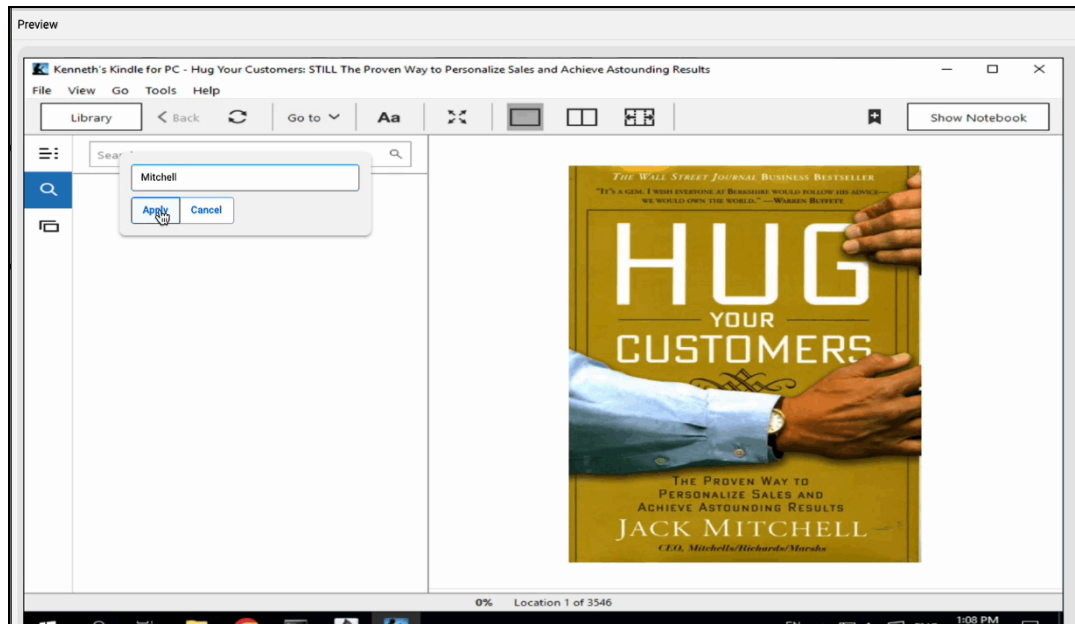


- Click in the Search bar and then right click.
- Select **Input Actions > Set Enter Value**.



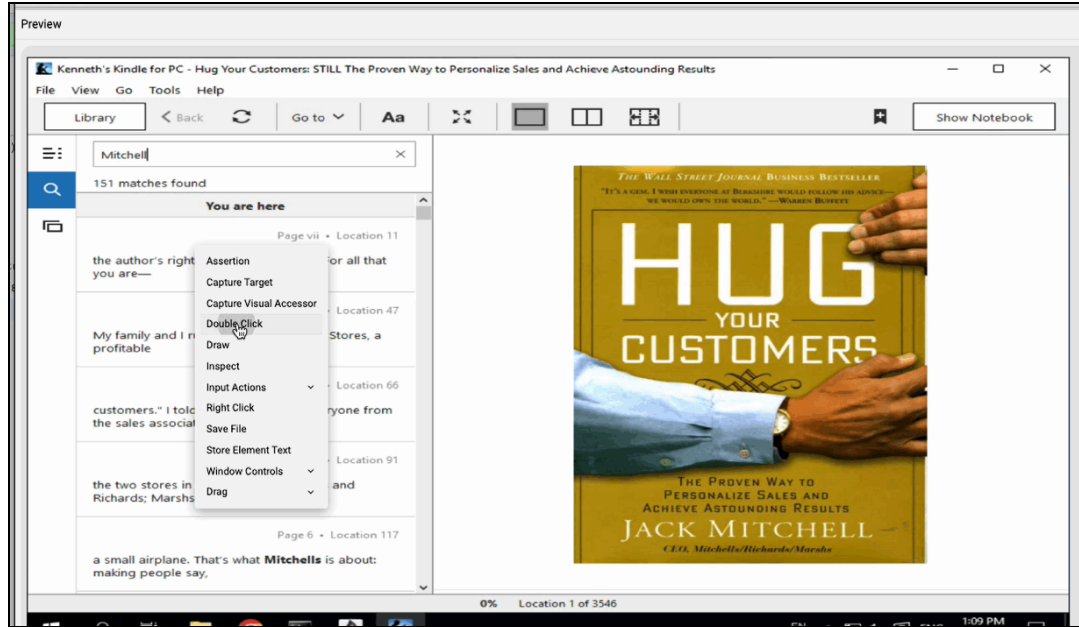
- Enter a search string. In this example the search string will be "Mitchell".

19. Click **Apply**.

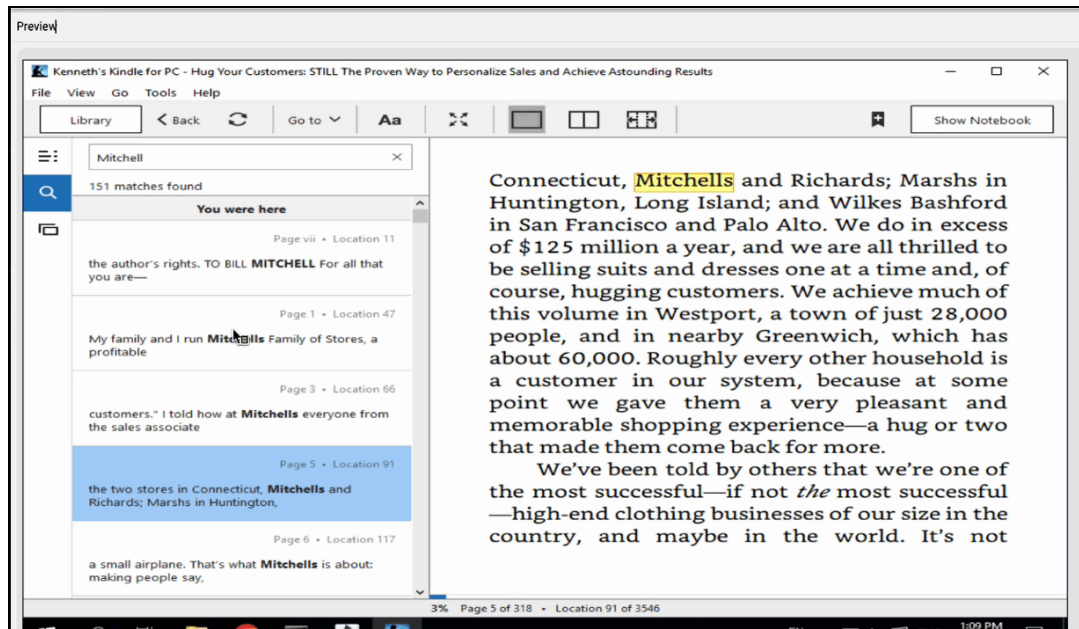


20. The search results load.

21. Select one of the search results and right click. Select the **Double Click** action.

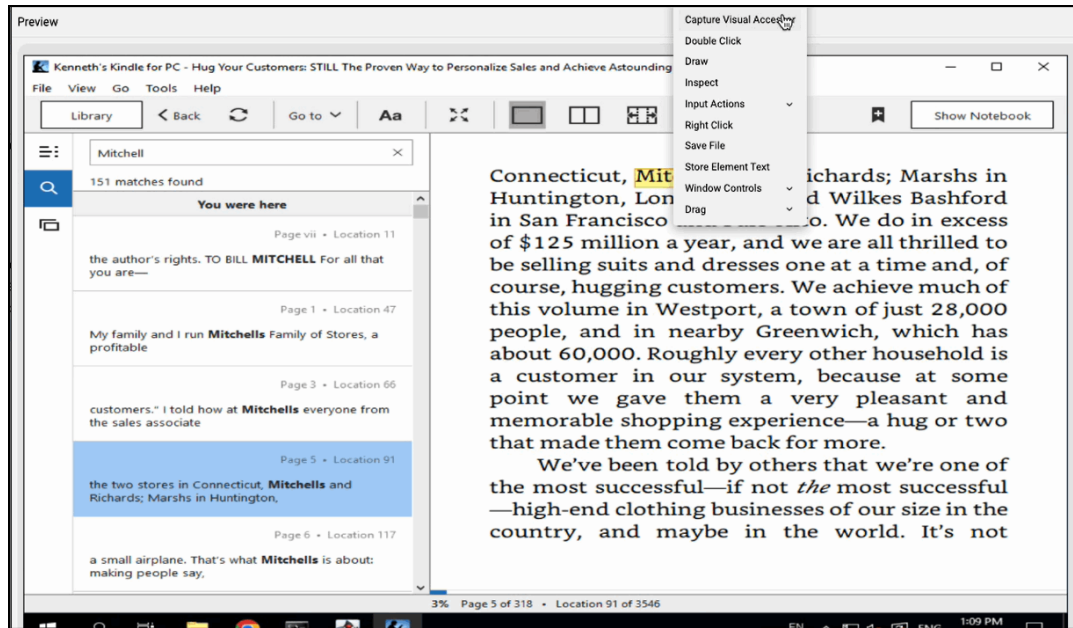


22. The page associated with the search result loads on the Kindle screen.

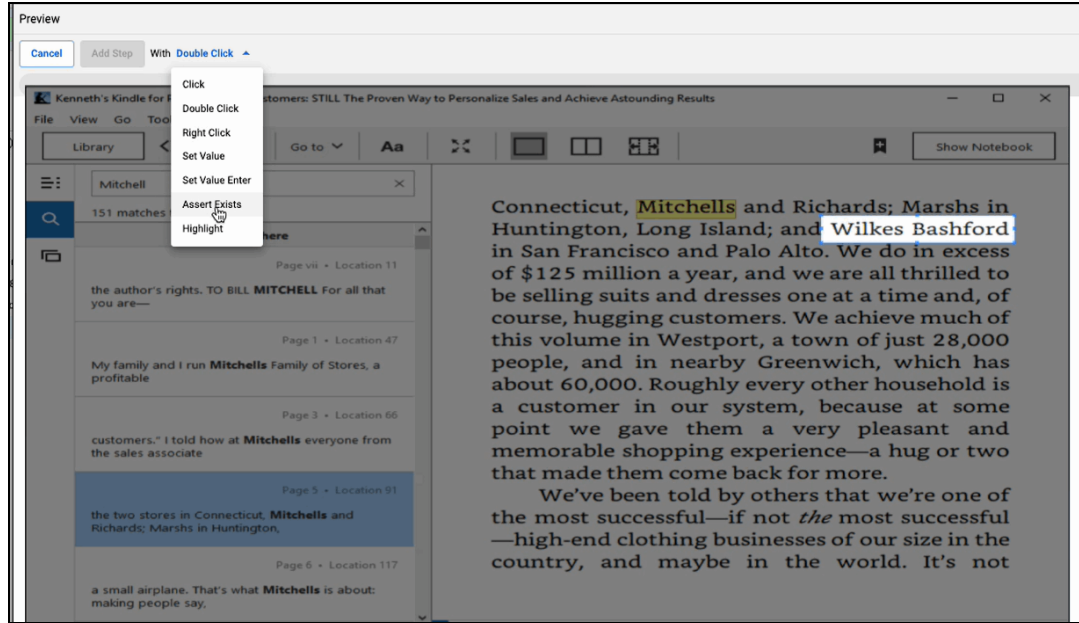


23. Hover over the page and right click. Select **Capture Visual Accessor**.

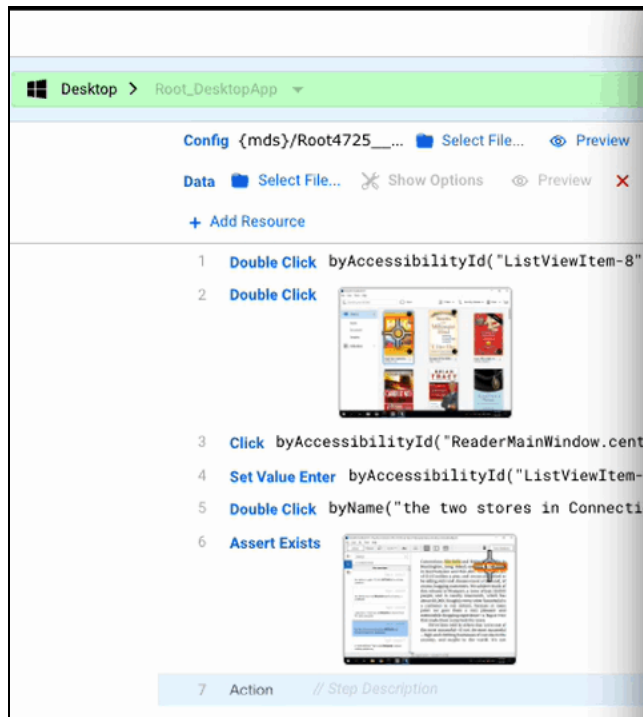
Desktop Application Testing



24. Adjust the Visual Accessor target. In this example it is the text "Wilkes Bashford" that is selected.
25. Click **Save Target**.
26. **Add Step** with **Assert Exists**.

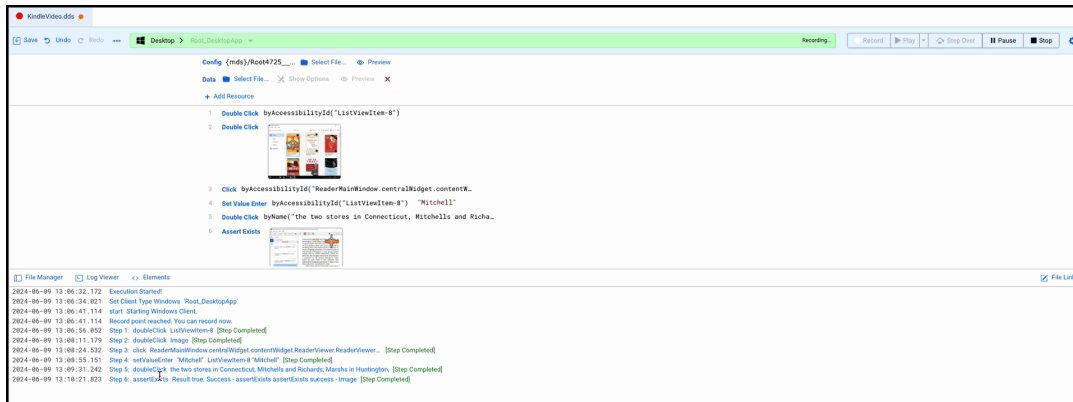


27. The test script updates in the Desktop Designer window.



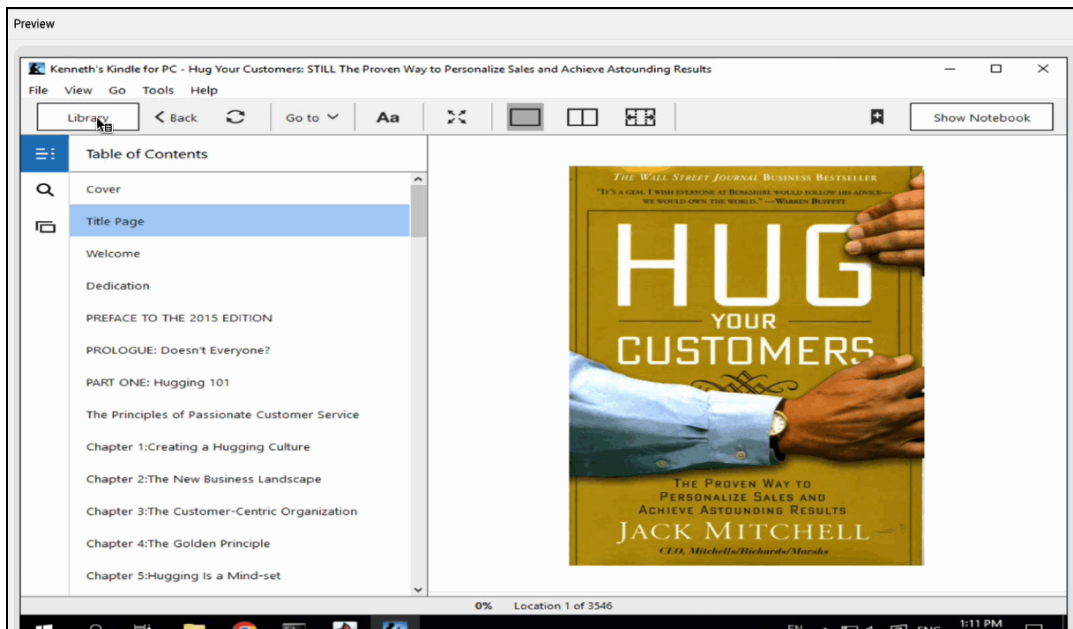
28. You can see more information in the **Log Viewer** tab.

Desktop Application Testing

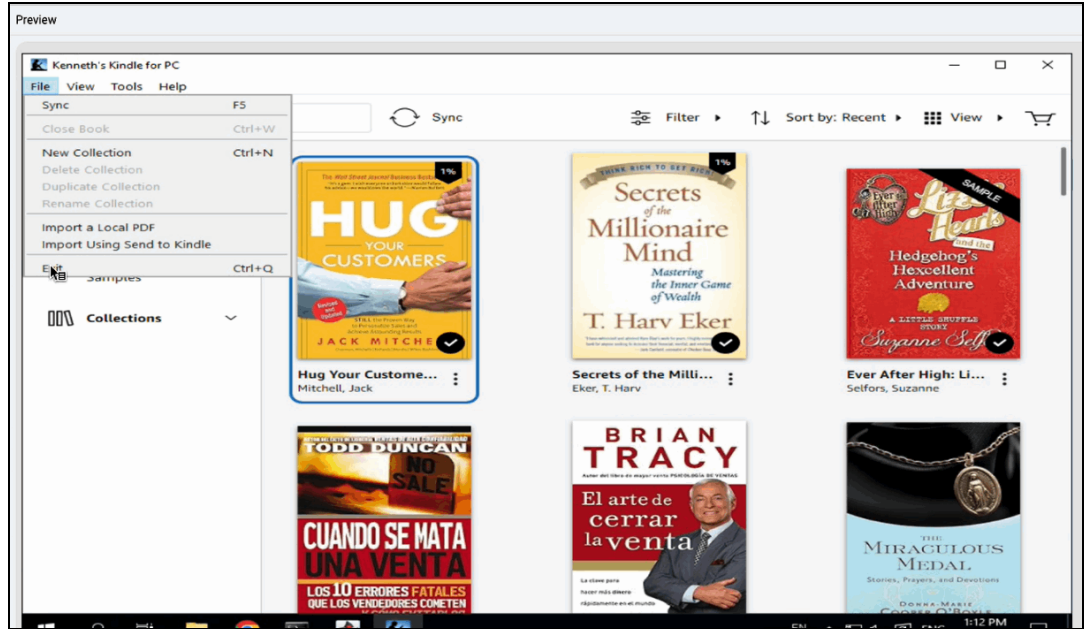


29. Click in the TOC and select **Title Page**.

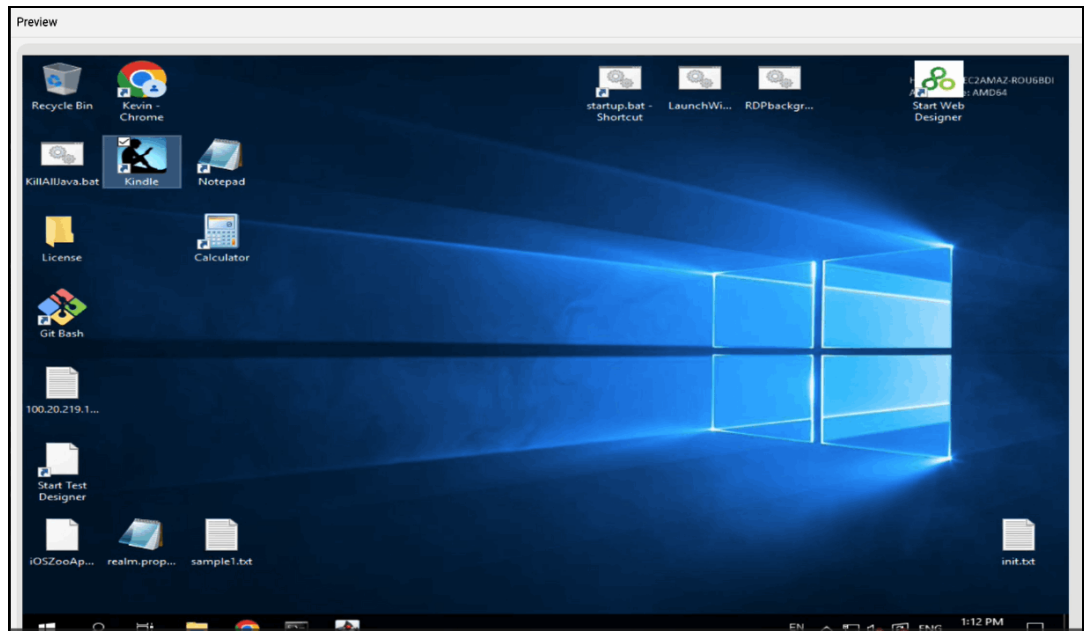
30. The title page loads.



31. From the File menu select **Exit**.

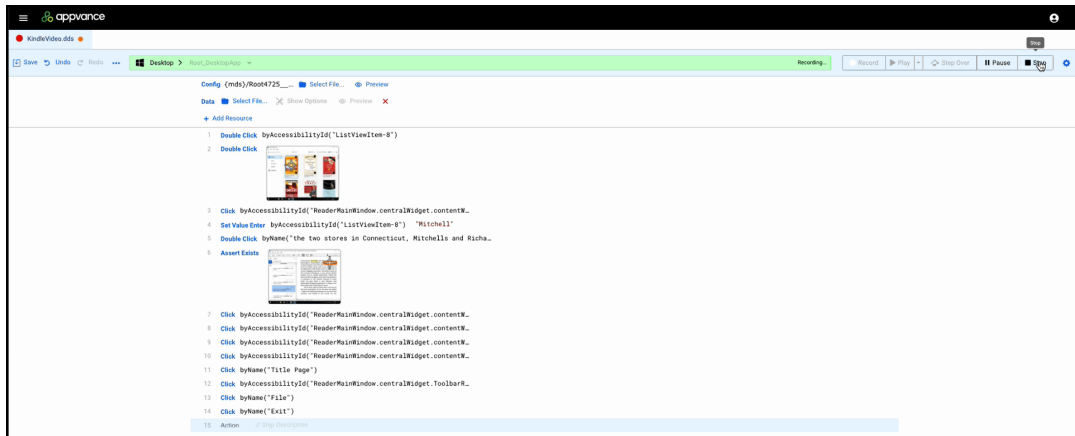


32. The Kindle desktop app closes.

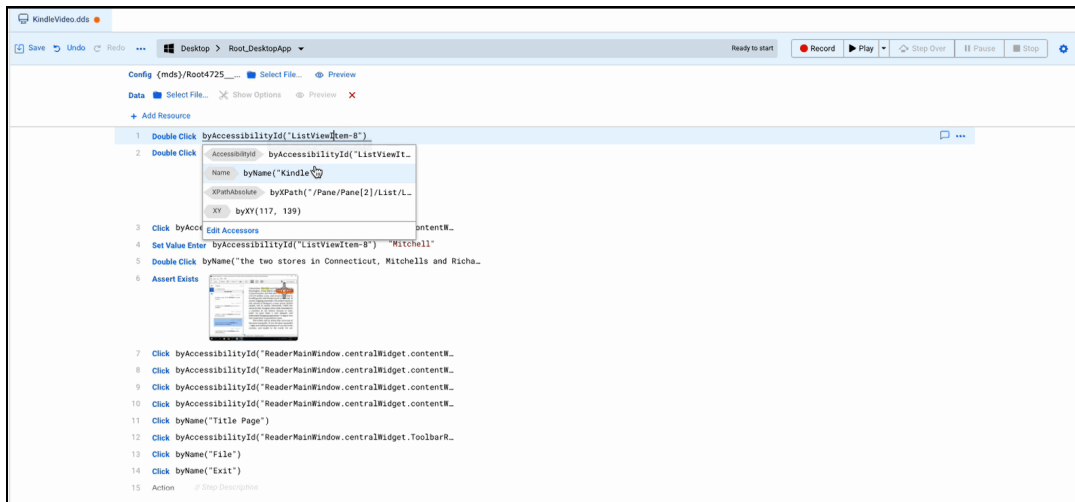


33. Click **Stop** to stop the recording.

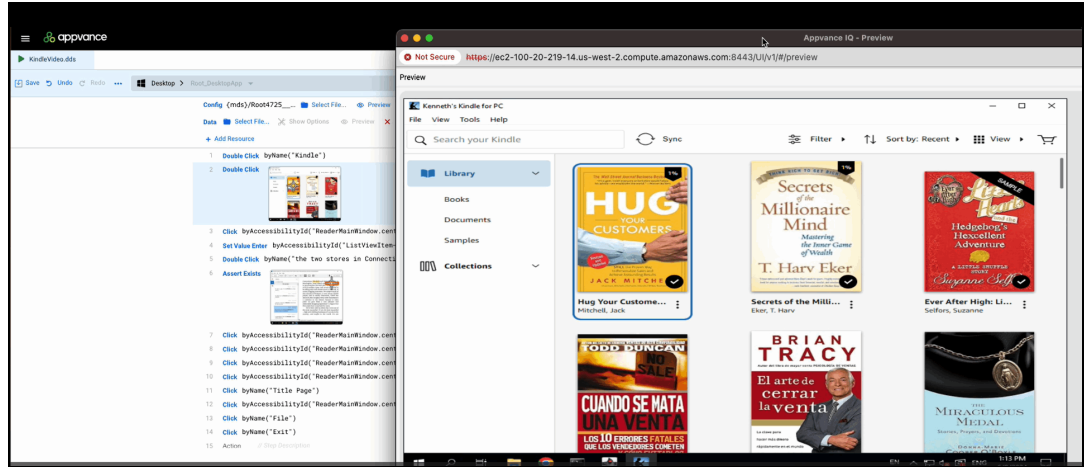
Desktop Application Testing



34. If needed you can edit your test script. For example, reordering an accessor.

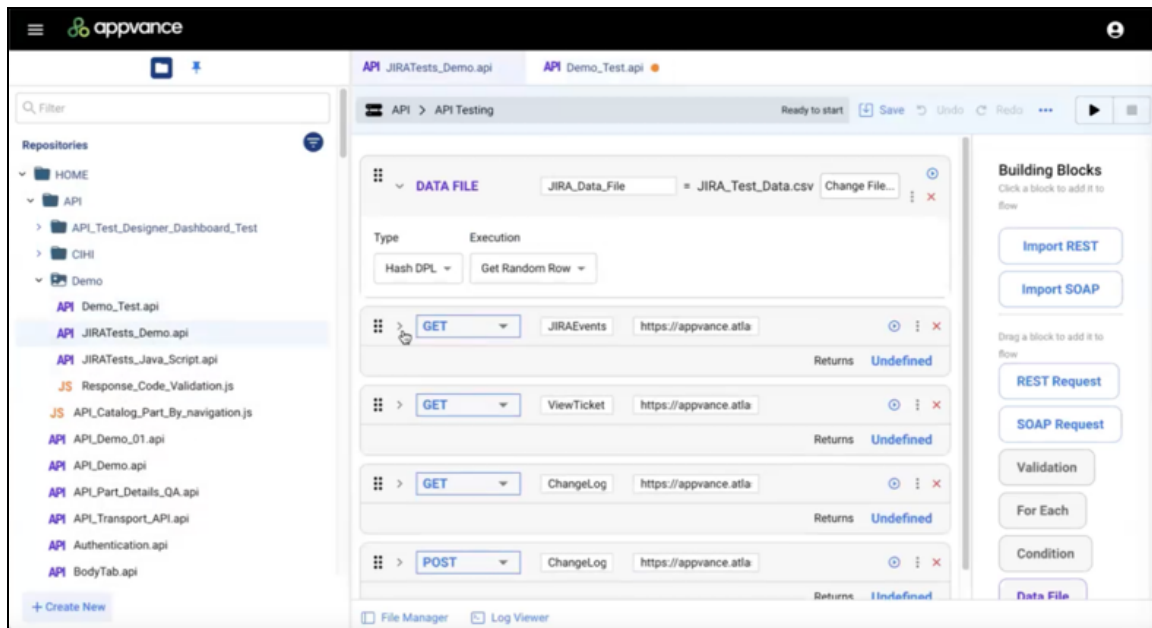


35. Click **Play** to play your test script.
36. The **Preview** window displays the playback while the corresponding steps are highlighted in the Desktop Designer window.



API Testing

API Designer has been revamped with powerful validations, DPL support, and more. This allows you create complex test scenarios in seconds, streamlining your testing workflow and boosting productivity.



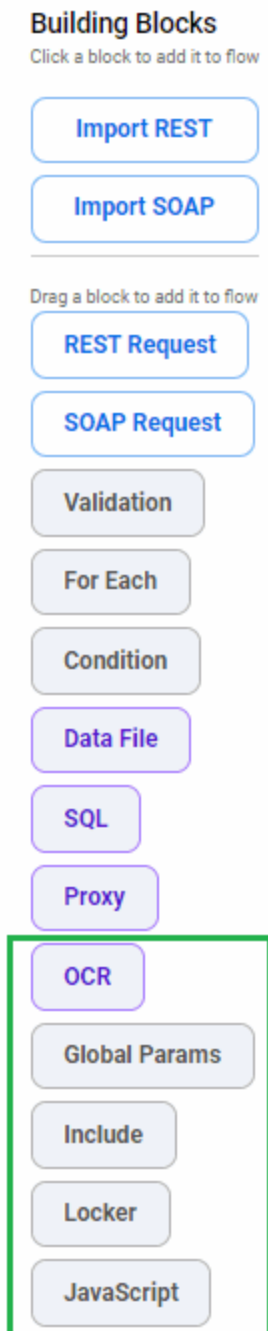
Enhancements include:

- Enhanced OCR Text Recognition: An advanced feature to get text from an image. This is not available in Postman.
- Support for Data Files: Added data file support (aka DPL) is available in API Designer like it is in the rest of the AIQ Designer suite.
- Global Header Section: This helps when the headers are repeated, but only need to be specified once.
- Multiple Edit Features added: You can now copy and paste API requests or import API Designer scripts.

- **Locker Functionality:** Allows you to use values in subsequent test cases or iterations.
- **Basic Certificate Support:** Now you can use Certificates for authorization purposes.

New API Testing Building Blocks

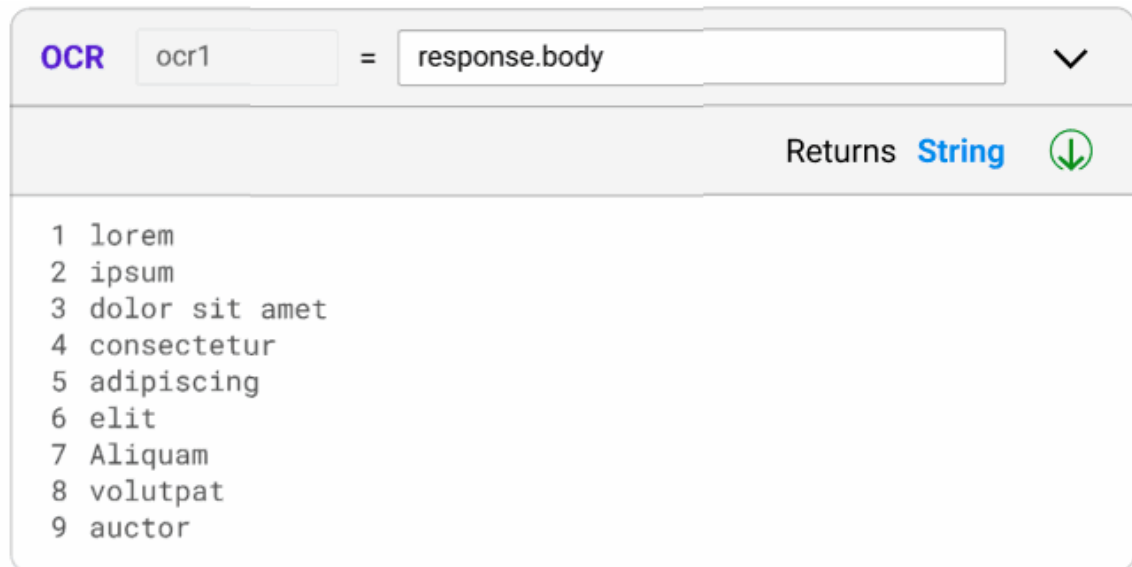
As part of the enhancements there are five new API testing building blocks.



The following building blocks have been added in the AIQ 5.2.0 release.

OCR Block

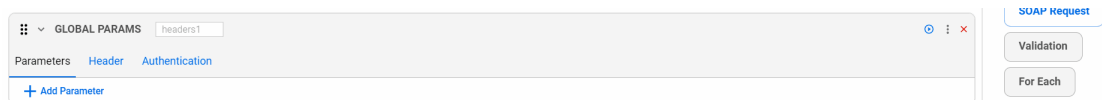
A new API building block is available in AIQ 5.2.0; an Optical Character Recognition (OCR) block.

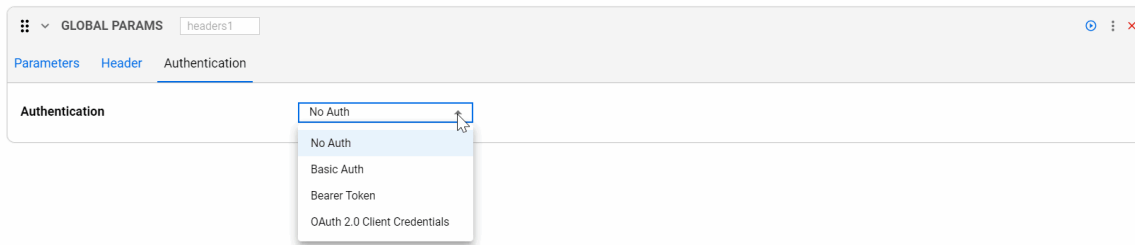


- Default name: `ocr1`
- Has one input, accepts body from previous responses
- When it is empty it shows the hint "Image"
- Returns string of characters, recognized text separated by a newline character

Global Parameters

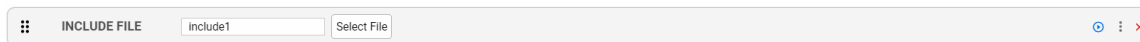
This is used as a Global Header. In each block you can add one or more headers that will be added to each subsequent request blocks.





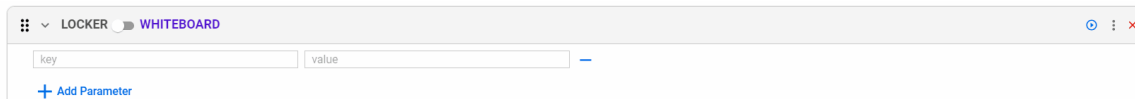
Include

You can include files with your API test.



Locker

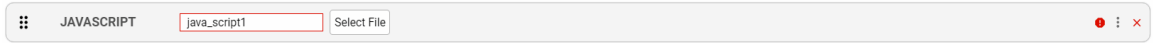
You now have the option to store the values of the results in a locker. This can be extremely useful if you are creating scripts in one of the AIQ Designers or Services Workbench, to allow you to store the data in the locker and reuse it in any script.



- Whiteboard
- Parameter
 - Key
 - Value

JavaScript

Allow you to include JavaScript in your API test.



Postman Conversion

AIQ introduces the ability to read Postman JSON files and convert them into Service Workbench API (`.swapi`) files. This allows you to integrate and utilize existing Postman tests within the Service Workbench environment for your automated testing.

Highlights

- New interface: New Interface to visualize imported API Scripts
- Import Feature to convert from Postman Scripts into SWAPI scripts
- Implemented the most common Postman commands. Other commands could be added quickly fast, in the rare case they are needed.
- Playback of SWAPI files to execute them. get them executed.
- Support for Scenarios, which means that you can add the SWAPI files into Scenarios and save results in the Dashboard

Converting Postman Files

Opening a Postman file from Services Workbench initiates the conversion process. The conversion process handles all key elements, including:

- HTTP methods
- Headers
- Body parameters
- URL parameters
- Authentication details
- Test scripts

One the Postman file is converted, you can edit the generated `.swapi` files within Services Workbench. From there you can execute the generated `.swapi` files individually or as part of larger test scenarios.

Functional Changes

These are the various small changes to functionality that were made in the 5.2.0 release of AIQ.

CICD Git Dashboard Housekeeping

Implemented a parameter to determine the maximum number of days that builds are retained. The parameter is `maxDays` and is shown in the below code snippet.

```
<taskdef name="historyManager" classname="com.appvance.ci.git.HistoryManager">
  <classpath refid="RestClientLibPath" />
</taskdef>
<historyManager
  folder="${localEmptyFolder}"
  product="${product}"
  environment="${environment}"
  executionType="scenarios"
  maxDays="10" <<<<< days to keep entries
  buildVersion="${buildVersion}"
/>
```

Removed Support for SoapUI

Support for SoapUI was removed in the 5.2.0 release of AIQ.

AISG - Creating Custom Actions

When you click on an unknown input on the actionable elements list from a state page, a custom action with the selected element will be created.

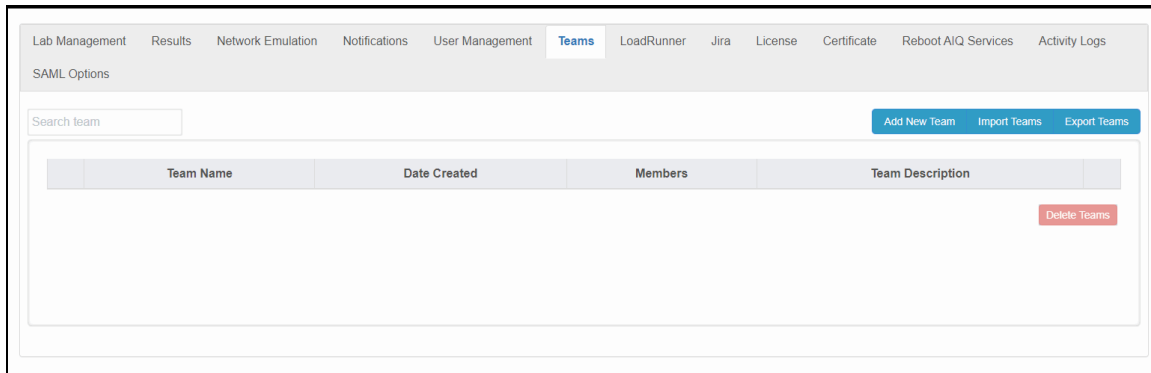
User Management - Teams

An enhancement has been made to user management so that you can now assign users to teams.



This function is only available to the Owner (appvance user).

Navigate to **Global Options > Admin Options > Teams** (tab).



Available functions:

- Add a team
- Delete a team
- Assign users to teams
- Remove users from teams
- Export teams
- Import teams



The import CSV file must contain the following five column: User Name, Role, Team, Password, Confirm Password

Delete Execution Reports

From the Reports screen you can now delete an execution report that you decide is not relevant. This improves the ability to identify relevant reports more easily by removing irrelevant reports. Deleting an execution report will remove all information and related reports for the execution.

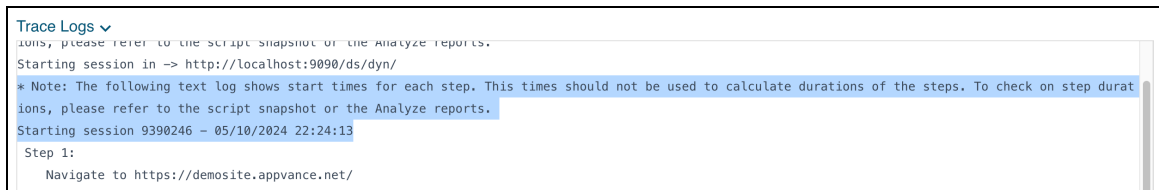
Some situations where this could be useful is to delete empty reports that clutter up the database. Empty reports are created when a scenario fails are one such cause of empty reports. For example, sometimes during the process of setting up a scenario, the scenario may play a number of times which generates a new report. However, when you open the report, an error occurs because the scenario failed to run to completion.

API Testing Reporting

The Dashboard now shows detailed reports for API scenarios similar to DS and JS scenarios.

Timestamp Displays

The Star Session timestamp is now in human readable format. Also, a note was added to clarify that the time stamps displayed in the text log should not be used to calculate step durations.



```
Trace Logs v
ions, please refer to the script snapshot or the Analyze reports.
Starting session in -> http://localhost:9090/ds/dyn/
* Note: The following text log shows start times for each step. This times should not be used to calculate durations of the steps. To check on step durations, please refer to the script snapshot or the Analyze reports.
Starting session 9390246 - 05/10/2024 22:24:13
Step 1:
  Navigate to https://demosite.appvance.net/
```

Iteration Summary Reports Optional

A new CICD parameter controls whether the Iteration Summary Report is generated and attached to the `report.html` CICD Dashboard page.

This option allows users who are only interested in the Functional Reports of their pipelines to decrease the time it take for the pipeline reports to be generated and reduce the stress on the results database. This can be a significant saving of time and resources for scenarios with multiple iterations.

The parameters is `includeSummaryReports`. The default value is `true`, which means the reports will be included. If it is set to `false`, the iteration

summary reports will not be attached to and attached to the `report.html` CICD Dashboard page.

Scenario Editor

The following functional changes in the Scenario Editor were added in AIQ release 5.2.0:

- Added option for API tests in the Script Type for Bulk Performance tests.

DataDog APM Integration

When setting up integration with DataDog the parameter `live=true` must be replaced with `live=false` otherwise DataDog will only show the last 10 minutes.



See [Integration with Datadog \(APM\)](#) for more information on the integration.

AIQ 5.0.x Enhancements

Depending on what AIQ release you are upgrading from to AIQ 5.2.0, there may be additional functional enhancements included in this release.



Refer to the [release notes](#) for all the AIQ releases since your current release for more details.

The following enhancements and functional changes were add in update releases of AIQ in 5.0.1 and later 5.0.x releases.

Scenario Execution Data

Scenario execution time data older than 90 days will no longer be stored. This will result in performance improvements because AIQ is no longer storing large amounts of historical data.



This change was introduced in the 5.0.1 release of AIQ.

Added Support for Encryption of JavaScript DPL and Hash DPL

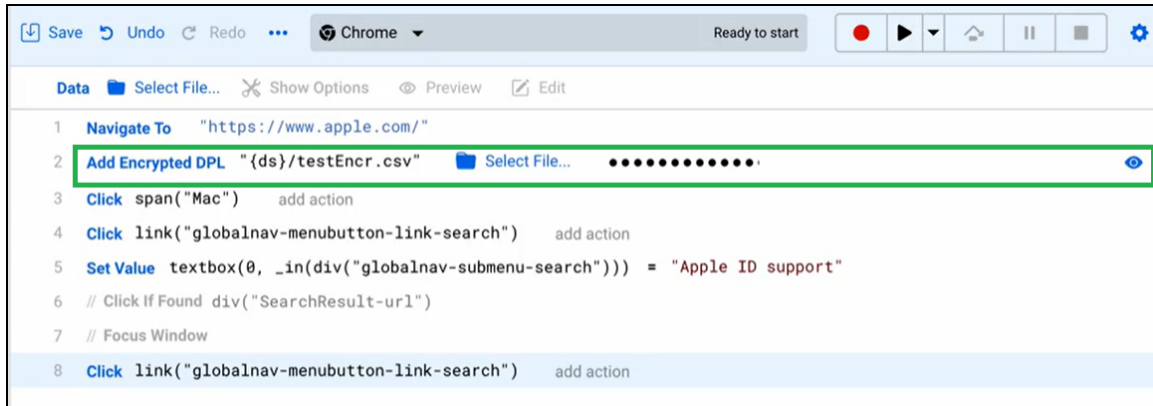
With this improvement, every time a Hash DPL or JavaScript DPL .CSV file is saved using AIQ, the file will be saved encrypted by default. This improvement is to increase user test data security. Existing unencrypted .csv files will continue to work within scenarios and scripts along with newly created encrypted .csv files.



This change was introduced in the 5.0.6 release of AIQ.

Added Add Encrypt/Decrypt DPL to Actions

Similar to the available actions of Added Hash DPL, Add JS DPL, and Add Synthetic DPL in a test script, you can now Add Encrypt/Decrypt DPL as well.



Improved Resource Management in Scenario Editor

Improvements have been made to resources management in the Scenario Editor. To make it clearer to users, the **Test Cases** panel has been moved and renamed to **Test Case Definitions**.

The behavior of AIQ when removing a test case definition or resources has been improved:

- If you remove a test case definition, the associated resources will only be removed if they are not being used by any other test case definition of any iteration.
- If you remove a resource, AIQ will inform you if that resource is also used in another iteration or test case definition. If that is the case, you will not be able to remove the resource.



This change was introduced in the 5.0.6 release of AIQ.

Enable Extraction from the Whiteboard within a SQL Script

In SQL there are ways to store results in the locker, and now you can inject items from the locker.

Sample Code in a Services Workbench Script

```
addToLocalWhiteboard("DefId", -1081701662);  
processSQLFile("readFromWhiteboard.sql");  
var value = getFromLocalWhiteboard("DefinitionValue");  
log("Definition value is "+value); // This should print 'Testcase'
```



This change was introduced in the 5.0.6 release of AIQ.

