2.1 - AI Script Generation (Autonomous Testing)

AI Script Generation (Autonomous Testing) — Appvance IQ documentation.

- AI Script Generation (Autonomous Testing)
- AI 3.0 (AIQ 5.x)
- Getting Started with AI Script Generation
- The Blueprint Dashboard
- The Coverage Map
- Resource Considerations & Performance

AI Script Generation (Autonomous Testing)

Contents

- AI 3.0 (AIQ 5.x)

Autonomous testing is all about application learning. Appvance AI creates a file called AISG (`.abp` file) which contains more information than any HTTP interactions (HTTP Traffic).

Terminology Note

The terms “blueprinting” and “blueprint” have been deprecated with the introduction of GENI. Those terms were used to describe the process and result of the exploration of your applications by Appvance’s AI Script Generation. The term “Digital Twin” will now be used when describing the results of the AISG process.

Because of the advancements of GENI, what is now created by the AI Script Generation process is not a mere representative map (“blueprint”) of your application, it is a true Digital Twin of your application.

In this process of creating the Digital Twin, Appvance AI explores the application to find out as much about it as it can, and dig as far as it can. It then identifies areas for input and prompts for that input, which can be provided for example via a CSV. Using that data, or input, Appvance AI then explores further into the application and produces a Digital Twin which is saved as a .abp file.

- Serves as a coverage map or AISG of the application.
- Knows all the functionality and features, how they are used, how they are accessed, and where they are in the application.
- Only needs initial parameters and data for the application’s text boxes, selects, checkboxes, radio buttons, and other user controls.
- Performs Auto validations.

Implementations

There are different implementations of AI depending on which version of AIQ that you are using.

- If you are using AIQ 5.0 and later, see AI Script Generation 3.0 in AIQ.
- If you are using AIQ 4.10 or earlier, see AI 2.0 in AIQ

Salesforce Examples

- If you are using AIQ 5.0 and later, see AI 3.0 Example - Blueprints for Salesforce Applications
- If you are using AIQ 4.10 or earlier, see AI 2.0 : Blueprints for Salesforce Applications

---

Last updated: March 25, 2025

If you have any feedback on this topic, please contact us at documentation@appvance.ai

AI 3.0 (AIQ 5.x)

AI Script Generation 3.0 in AIQ

AIQ release 5.0.0 contains a comprehensive update to the AIQ AI engine that results in enhanced performance, reporting, and bug-finding accuracy.

Some highlights of the improvements include:

- Optimized performance means that your AISG will run faster.
- Calculated page states are more accurate and reliable when executing custom actions on specific pages.
- User interface has been re-factored to make it more intuitive and accessible to new users.
- You can data drive your AISG by adding data sets prior, during, and after AISG execution. You do not have to reset or restart the AISG if you forgot to add a data set when you added an execution.
- Existing AISG that were created in older versions of AIQ will automatically be upgraded to the new format when you open them for the first time in AIQ 5.0. See Upgrading Blueprint Files to AI Script Generation Files for more information.
- AISG Designer is the new AIQ 5.x AISG user interface. Unlike the updated Web Designer interface, there is no corresponding "classic" version of AISG Designer in AIQ 5.0.

In AIQ 5.0.0 AISG is only supported on Windows test node.

Autonomous Testing Options

This is the Autonomous Testing menu in AIQ release 5.0.0.



The following sections detail working with those options:

- AISG Designer - Working with AI Script Generation
- Coverage Map - Exploring the Coverage Map

Changes from Previous Releases

- Mobile AISG is not enabled in the AIQ release 5.0.0. This function is being enhanced and will be reintroduced in a future release of AIQ.
- The Blueprint Template Editor has been removed as part of the AISG enhancements.
- The Compare AISG option is not enabled in AIQ release 5.0.0. This function is being enhanced and will be reintroduced in a future release of AIQ.
- The Generate Scripts option is not enabled in AIQ release 5.0.0. This function is being enhanced and will be reintroduced in a future release of AIQ. However, you can now generate a DS script from the Coverage map, or download all actions in the AISG as a DS script.

A dedicated training course on AISG is available as part of Appvance's training offerings. See the Appvance Training page for more information.

For your convenience, the information about the new AISG functionality is included in the **AIQ 5.0.0 Release Summary**. This document also contains the AIQ 5.0.0 Release Notes, which includes information about the AIQ 5.0.0 release.

Last updated: April 01, 2025

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Getting Started with AI Script Generation

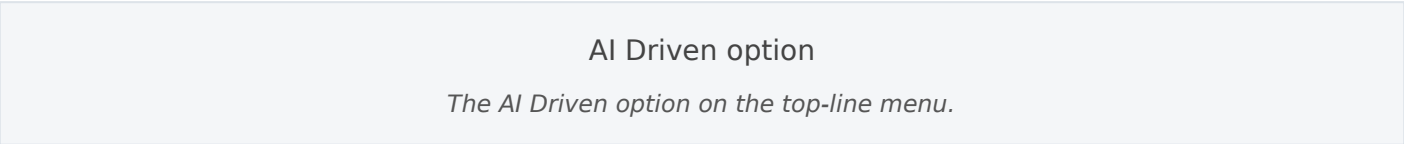
This page walks through the prerequisites and the different ways to launch and create an AI Script Generation in Appvance IQ.

> **Prerequisite:** before you can work with AI Script Generation, you must have a **repository configured** to store your AISG files. *Storing files locally is not supported.*

Launching AISG

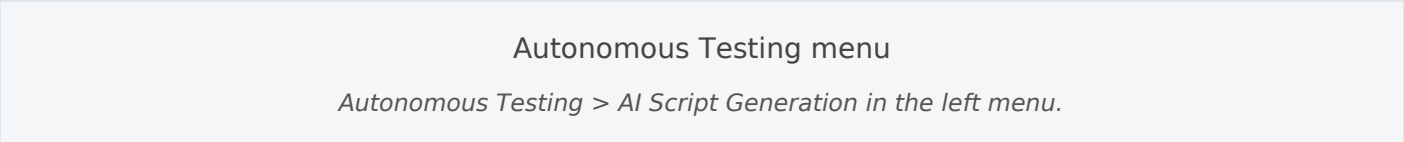
You can access AI Script Generation in two ways from the AIQ home screen:

- From the top-line menu, click **AI Driven**.
- From the left menu, expand **Autonomous Testing** and select **AI Script Generation**.



AI Driven option

The AI Driven option on the top-line menu.



Autonomous Testing menu

Autonomous Testing > AI Script Generation in the left menu.

Ways to begin

- **Upgrade** an existing pre-AIQ 5.0 Blueprint file to the AI Script Generation format.
- **Create a New Web AI Script Generation** — for web applications.
- **Create a New Mobile Digital Twin** — for mobile apps.
- **Use an AI Script Generation Template** — export an existing AISG as a reusable `.abpt` template.

Upgrading pre-5.0 Blueprints

1. Navigate to **Autonomous Testing > AI Script Generation** (or click **AI Driven**).
2. Click **Open** and navigate to the repository containing your AISG files.
3. Open a blueprint. The **Create New Template** dialog informs you that the blueprint must be updated.
4. Click **Confirm**. Once the AISG file loads, the upgrade is complete — SmartTags and validations carry over automatically.

Open repository

Opening the repository that holds your blueprint files.

Confirm upgrade

The Create New Template confirmation dialog.

Upgraded blueprint

The upgraded AI Script Generation file, ready to execute.

What happens behind the scenes on upgrade

- A backup of the old blueprint is created as `<blueprint_name>-backup.abp`.
- A blueprint template file (`<blueprint_name>.abpt`) is created.
- A new folder named after the blueprint is created, and all linked files (SmartTags, validations) are copied into it.
- A new blueprint file with the same name is created from the template.

Source: Launching AI Script Generation (docs.appvance.net)

The Blueprint Dashboard

The **Blueprint Overview** (or **Blueprint Dashboard**) shows what the AI found while the Blueprint is running or after it has run. From here you interpret the results of an AISG execution.

Blueprint Dashboard

The Blueprint Dashboard in AIQ 5.x.

> **Note:** because AISG runs in **headless mode**, you no longer watch browsers navigate the application in real time (as you could before AIQ 5.0). Results are reviewed on this dashboard instead.

Blueprint Overview

A summary of *all* executions of the blueprint:

- **Unique States** — number of unique page states discovered across all executions (total application coverage).
- **Unique Pages** — number of unique pages discovered across all executions.
- **Duration** — total runtime of all blueprint executions.

Execution Details

A summary of a *particular* execution:

- **Completed Actions** — actions performed by the blueprint.
- **States** / **Pages** — page states and pages discovered in this execution.
- **Duration** — runtime of this execution.

Dashboard controls

Control	Function

>**Coverage Map**	>Launches the Coverage Map — visualizes the paths the AI can take through the application.
>**SmartTags / Validations**	>Visible only if your blueprint uses SmartTags or Validations.
>**Requests**	>All requests made by the blueprint.
>**Failed Actions / Failed SmartTags**	>Details of failures per execution — including the pages where they occurred.
>**New Inputs**	>Opens the Pages information of the execution (replaces “Inputs Found”).
>**Dataset**	>Add execution data (CSV data sets) from the dashboard — even after the run.
>**Browsers**	>Number of browsers running the blueprint (set in Add Execution).
>**Add Execution**	>Adds a blueprint execution — at least one is required per blueprint.
>**Configuration**	>View configuration settings. *Information only — configurations cannot be modified after execution.*

Additional options

- **Rerun failed actions** — retest failed actions (useful if you forgot to add a data set).
- **Export Template** — export the blueprint as a reusable Blueprint Template (`.abpt`).
- **Download failed actions / all actions as DS or JS** — export scripts for further analysis or reuse.

Source: Exploring Blueprint Results (docs.appvance.net)

The Coverage Map

The AISG **Coverage Map** shows the multiple paths the AI can take while interacting with the application being tested. It lets you visualize the **health of the application** and understand all potential user flows — including areas plagued by errors that may need rework.

Coverage Map

The AISG Coverage Map in AIQ 5.2+.

Three coverage map types (AIQ 5.2.0+)

Type	What it shows
>**Original**	>The default visualization — the view used in previous versions of AIQ.
>**Extended Potential**	>Enhanced view with more page states and relationships the AI *learned about* — even paths it did not actually navigate.
>**Actual Path**	>The exact path the AI followed through the application — great for tracking AI training and generating precise DS scripts.

> **Note:** to take advantage of the new Coverage Map types, you must **rerun any existing 5.0.x executions on 5.2.0**. Opening a 5.0.x `.abpt` file alone is not enough.

Node colors and their meaning

Color	Meaning
White	>Not executed
Green	>Executed with no errors
Yellow	>Executed with a non-critical error (e.g. a 500 error)
Red	>Executed with a critical error (e.g. a 404 error)

Color	Meaning
Blue	>The execution path from the start point

Interacting with the map

- **Zoom in / out** to see more detail, including the actions and accessors associated with a node.
- Explore the map **in real time** while the blueprint is still running.
- **Click a node** to see its actions and accessors, plus their exact location in the application.
- **Generate DS Script** — creates a script with the steps the AI took to reach a specific page.
- Use the **Request tab** to see the associated requests.
- **Search URLs** in the map; click a result to see the page state details.

Node details

Node details with actions, accessors, and Generate DS Script.

Filtering requests

You can filter out specific requests or errors (e.g. 404s) from the Coverage Map. The filter is saved to the Blueprint (`.abpt`) file so the requests won't display on future executions. To hide a request, right-click it and select **Hide URL** — it moves to the **Hidden Requests** list, visible per node or per execution.

Source: Exploring the Coverage Map (docs.appvance.net)

Resource Considerations & Performance

Performance of the AI Script Generation process depends heavily on the resources available to your **test node** and the configuration choices you make. Here are the general guidelines from Appvance.

Headless vs non-headless browsers

You can run the AI browsers in **headless** or **non-headless** mode. Headless mode is recommended for increased performance and lower resource usage. As a rule of scale: if you can run **5 non-headless browsers**, you can likely run **10–15 headless browsers** with the same performance.

Mode	Limiting factor	Typical scale
>**Non-headless**	>Memory (RAM) on the test node	>16 GB → 5 browsers · 32 GB → 10 · 64 GB → 20
>**Headless**	>CPU speed and thread count	>10–15 browsers at 2.4–2.6 GHz base

> **Tip:** if performance is slow, try **reducing the number of browsers**. Running more browsers than the machine can handle makes the whole AISG process slower — not faster.

Delay between requests

Delay Between Requests (ms) is an execution option that determines how long AIQ waits for the application to load before reporting a failure while navigating. A longer delay gives the application more time to respond. As a side effect, higher delays can let you run **more browsers** (some go to sleep while waiting). For example: 10 browsers work best at 0 ms delay, but at 20 ms you may run 15. This is not a direct correlation — **benchmark your own environment**.

Types of applications

- **Static** — very static apps with few elements: up to **20 headless browsers** on a standard 8-core / 32 GB node.

- **Dynamic** — highly dynamic apps (moving elements, Shadow DOM, iframes): do some **basic benchmarking**.

Basic benchmarking

1. Run a quick AISG execution — e.g. compare 10 headless vs 5 non-headless browsers.
2. Monitor memory and processor utilization.
3. Use the results to decide on the browser count for your real AISG run.

AIQ limits

50
absolute max browsers (hard limit)

35–50
optimum browsers even with abundant resources

> **Also note (AIQ 5.0):** AI blueprinting is supported only on **Windows test nodes**. Mobile blueprinting, Compare Blueprints, and the Generate Scripts option are disabled in 5.0 — use the Coverage Map DS generation instead.

Source: Resource Considerations (docs.appvance.net)