

Automated Testing of Human Machine Interfaces (HMIs) with Keysight Eggplant Test

Testing HMI interfaces in vehicles

Using Keysight Eggplant Test for HMI Testing

Human Machine Interfaces (HMIs) are a critical component of the products provided by a wide variety of industries. HMIs include electronic devices, such as mobile phones, and physical components, such as a radio knob or hand-held point of sale (POS) device. Consider the example of HMIs in an automobile. Auto HMIs encompass a wide range of features and components that help vehicle occupants control and access vehicle functions, information, and entertainment systems. Auto HMIs include dashboard displays, touchscreens, physical controls, voice recognition, gesture control, steering wheel controls, heads-up displays (HUD), advanced driver assistance, and in-vehicle infotainment systems. Ensuring these interfaces are easy to use and do not distract drivers is essential for a safe and exceptional in-auto experience.

You can use Keysight Eggplant Test (hereafter called “Eggplant”) to test any auto HMI. Our realistic testing approach simulates real-world driving conditions and user interactions, providing you with valuable insights to fine-tune your HMI system for an exceptional in-car experience. By driving a robot to push buttons on a steering wheel or simulating interactions on a touchscreen, Eggplant can validate HMIs by “seeing” and interacting with them the way a user does. Eggplant’s innovative testing solution combines computer vision and robotics to mimic a user’s view of your HMIs, providing invaluable visual verification testing without manual intervention.



This application note illustrates how you can use Eggplant to test your auto HMIs by modeling their user journeys.

This note includes the following sections:

[Setting Up Eggplant](#)

[Modeling an HMI System](#)

[Running a Model](#)

Setting Up Eggplant

Before you build your first model, you must set up your test environment. You want to install and configure Eggplant's Digital Automation Intelligence (DAI) web server application. This is Eggplant's modeling tool. It is available as an on-premise or hosted application (DAI Eggplant Cloud). Please see this [DAI Documentation](#) for information about installing the on-premise version. For more information about DAI Eggplant Cloud, please see its [documentation](#).

You will also need to install Eggplant Functional to write and edit your tests and configure your systems under test (SUTs), which could involve setting up robots.

Please contact your technical success manager for more information about setting up your test environment.

Once your DAI test environment is set up, you can log in and build a model of your auto HMI.

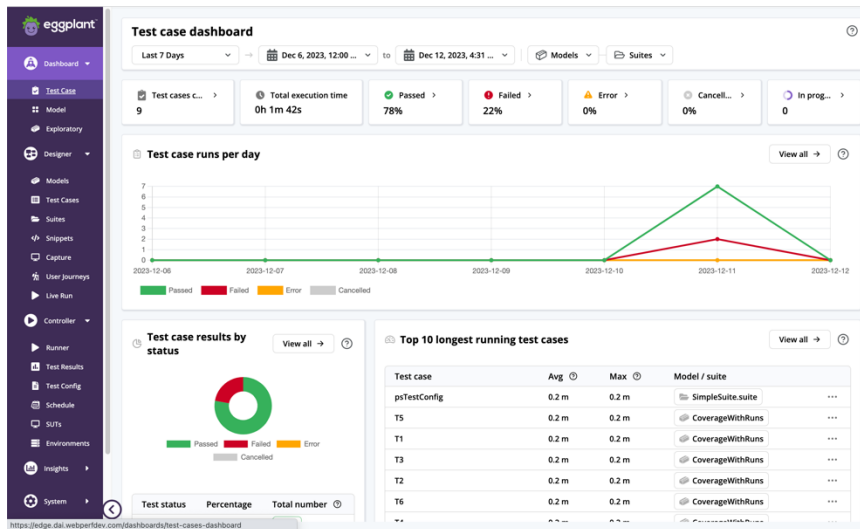


Figure 1. Eggplant DAI Model Dashboard

Modeling Auto HMI

Eggplant DAI is the Eggplant tool you use to model user journeys. Once you model your HMI, you can direct DAI to run specific tests by defining them in test cases. Or you can put DAI in exploratory mode and let it randomly identify and test workflows. You can also use DAI to schedule and run scripts that aren't models, but this application note does not cover that topic.

The following example demonstrates how to model an auto HMI workflow in DAI.

Building a model

When you design a model for auto HMIs, you need to consider the different tasks a driver performs, which include using the different HMIs, to create user journeys. See [Building Models](#) in the DAI documentation for more information.

Design your model

1. To begin building your model, navigate to the **Models** page by choosing **Designer > Models** from the navigation menu on the left.

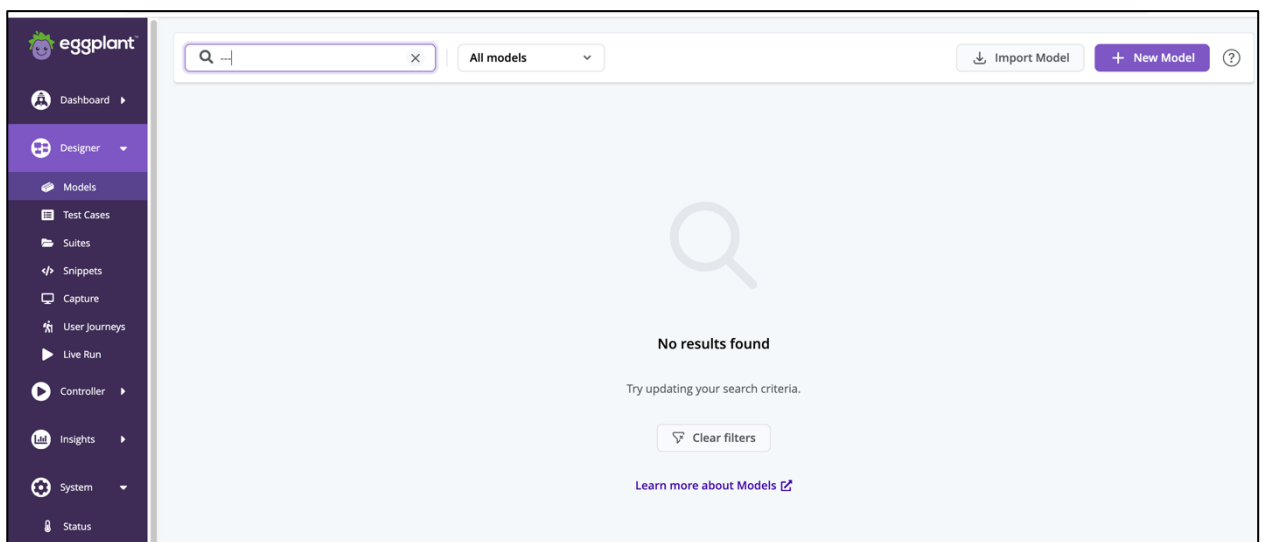


Figure 2. The DAI Model page

2. Create a new model by clicking the + **New Model** button in the upper-right corner of the DAI Model page. The New Model page opens.

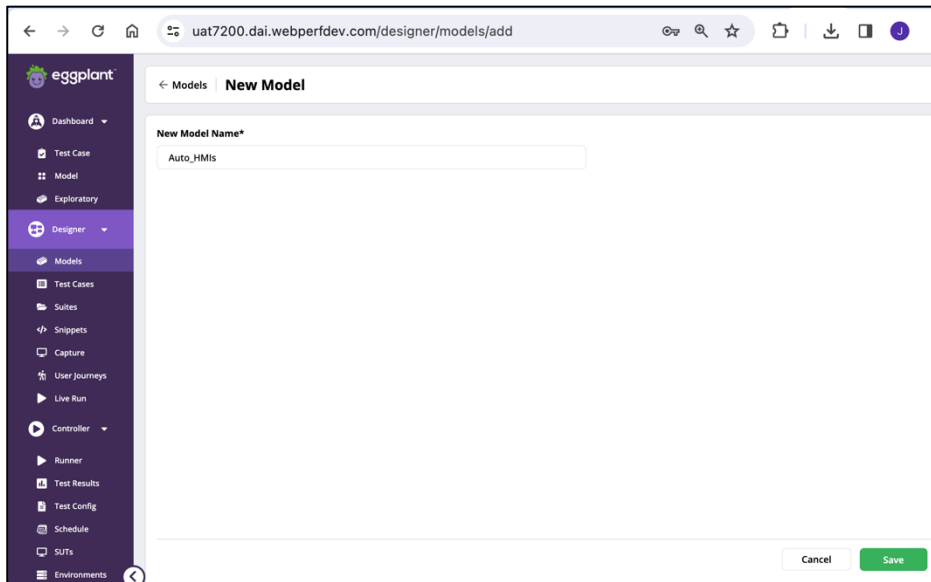


Figure 3. The **New Model Name** field on the DAI New Model page

3. Enter a descriptive name for your model, such as “Auto_HMIs” in the example above.
4. Click the **Save** button in the lower right corner of the **New Model** page. You may need to scroll down and right to see it. The Designer workspace opens with your new model. A green “Initial” state is automatically added to all new models.

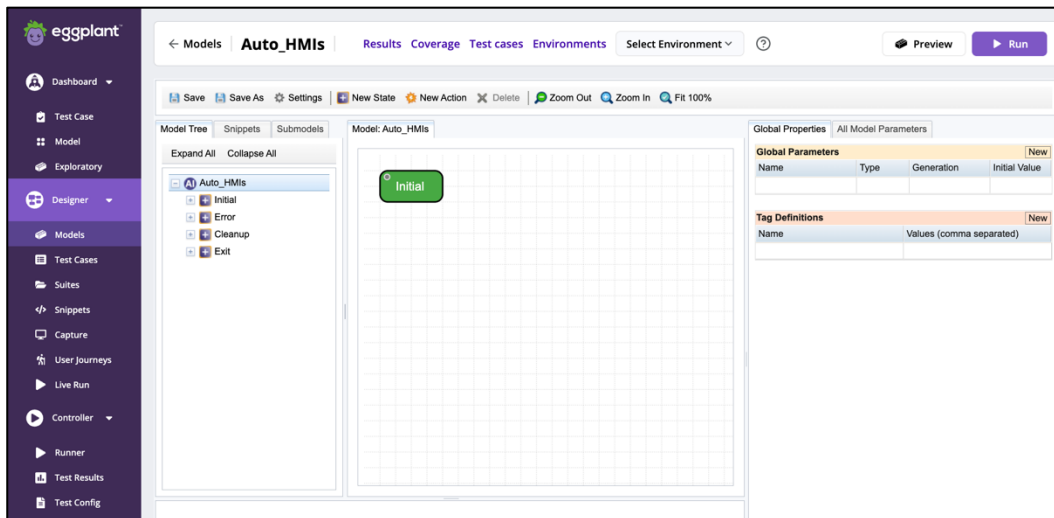


Figure 4. A new model in the DAI Designer

5. Change the name of the “Initial” state by selecting it, and when the **State Properties** tab opens to the right of the workspace, type the new name in **Name** field and press Enter/Return to save it. The example state name below is “LaunchAutoHMIs.” The example model uses this state to launch the auto HMI tests. The next few steps will show how to add states and actions to model radio HMIs in an auto.

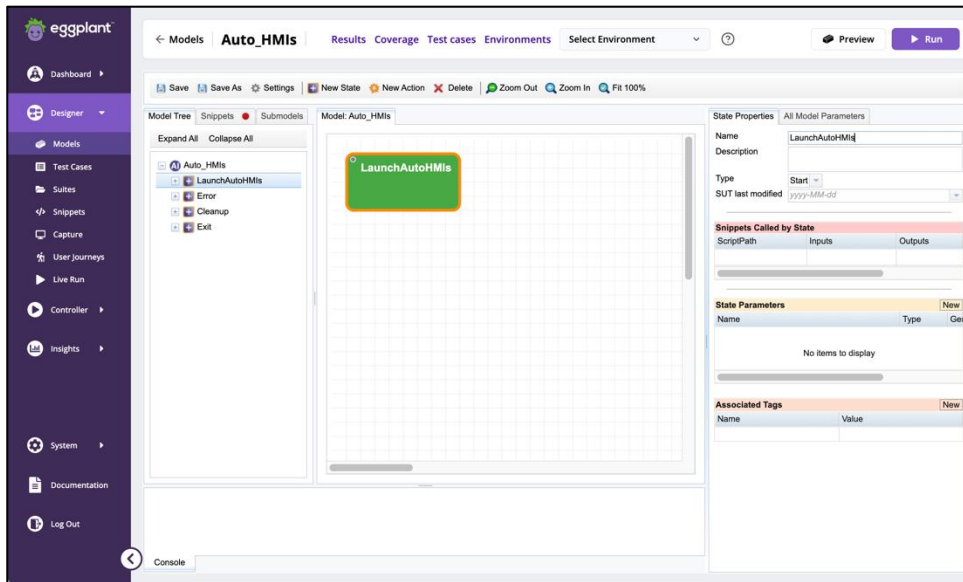


Figure 5. A renamed initial state in a new model

6. Add an action to test the radio by right-clicking the “LaunchAutoHMIs” state and choosing **New Action** from the context menu (or clicking the **New Action** button above the workspace). The action is added inside the state, and the **Action Properties** tab opens to the right of the workspace, next to the **All Model Parameters** tab.

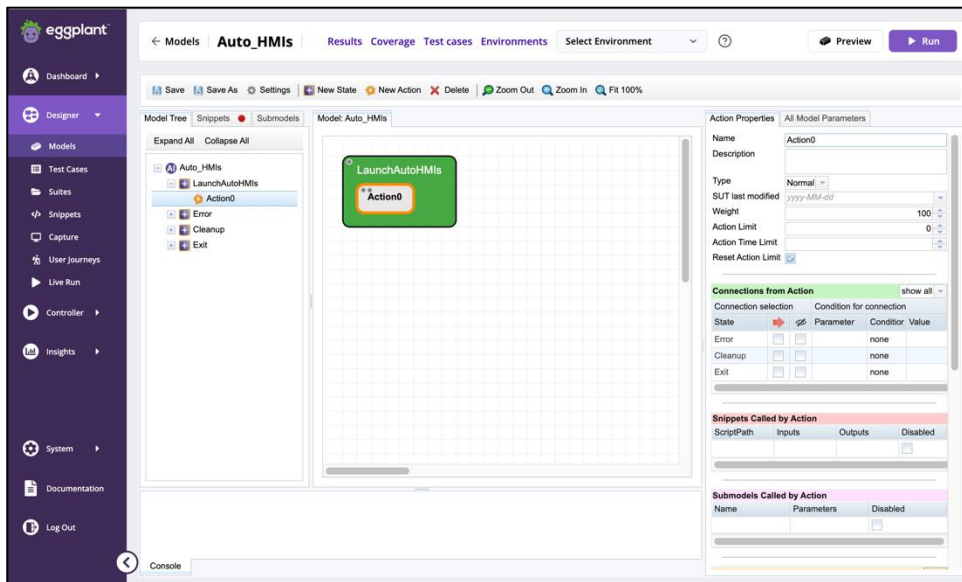


Figure 6. A state with new action

7. Name your new action by selecting it, and when the **Action Properties** tab opens to the right of the workspace, type the new name in the **Name** field. Press Enter/Return to save it. The example name for this action is “TestRadio.” You could add more actions here for the different HMIs you can model for an auto to this state, such as “LaunchHeadsUpDisplay” or “TestPhone”.

8. Add a state for one of your radio HMI's by right-clicking in the workspace and choosing **New State** from the context menu (or clicking the **New State** button on the toolbar above the workspace). When you add a state, the **State Properties** tab opens to the right of the model workspace, next to the **All Model Parameters** tab.

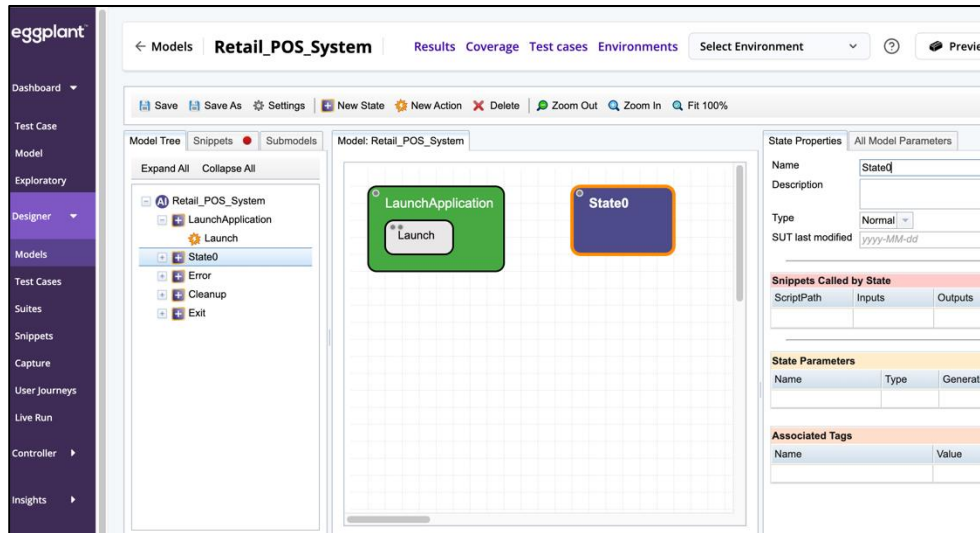


Figure 7. A new state added to a model

9. Provide a name for the state in the **Name** field on the **State Properties** tab and press Enter/Return to save it. The example state name below is “TouchScreen”.

This screenshot shows the 'State Properties' tab for a state named 'TouchScreen'. The 'Name' field contains 'TouchScreen'. Below it are fields for 'Description', 'Type' (set to 'Normal'), and 'SUT last modified' (set to 'yyyy-MM-dd'). There are three expandable sections: 'Snippets Called by State' with a table for ScriptPath, Inputs, Outputs, and Disabled; 'State Parameters' with a table for Name, Type, Generation, and Initial Value; and 'Associated Tags' with a table for Name and Value. Each section has a 'New' button to add entries.

Figure 8. Providing a state name on the DAI model **State Properties** tab

10. Add a new action to the “TouchScreen” state to test the touchscreen controls. For example, name it “TestScreenControls.”

11. Add two more states to the model for different HMIs, such as “DashboardKnobs” and “SteeringWheelControls.”
12. Add two states to the right of the HMI states, which will be for the ON and OFF states of the radio.
13. Add a “TurnON” action to the “RadioOFF” state, and appropriate actions to the “RadioOn” state, such as “Tune,” “VolumeUp,” “VolumeDown,” and “TurnOFF.”
14. Now return to the “Radio” state and start mapping your workflows, beginning with the “TestRadio” action and “RadioTouchScreen” state. Select the “TestRadio” action, right-click, and when the **Action Properties** tab opens, click the checkbox next to “RadioTouchScreen” in the outbound (red arrow) column in the **Connections from Action** part of the tab.

The screenshot shows the 'Action Properties' dialog box for the 'TestRadio' action. The 'Connections from Action' section is expanded, showing a table with columns for State, Connection selection, and Condition for connection. The 'SteeringWh' state is selected with a checked checkbox in the 'Connection selection' column.

State	Connection selection	Condition for connection
RadioOFF	☐	none
RadioON	☐	none
SteeringWh	☒	none
RadioTouch	☐	none
DashboardKnobs	☐	none
Error	☐	none
Cleanup	☐	none
Exit	☐	none

Figure 9. A checkbox showing outbound connection from an action to a state

15. Now repeat this mapping to map the “TestRadio” action to the “SteeringWheel” and “DashboardKnobs” states as shown above.
16. Now map the different control states (“RadioTouchScreen”, “SteeringWheelControls”, and “Dashboard Knobs”) to the radio states (“RadioOn” and “RadioOFF”).

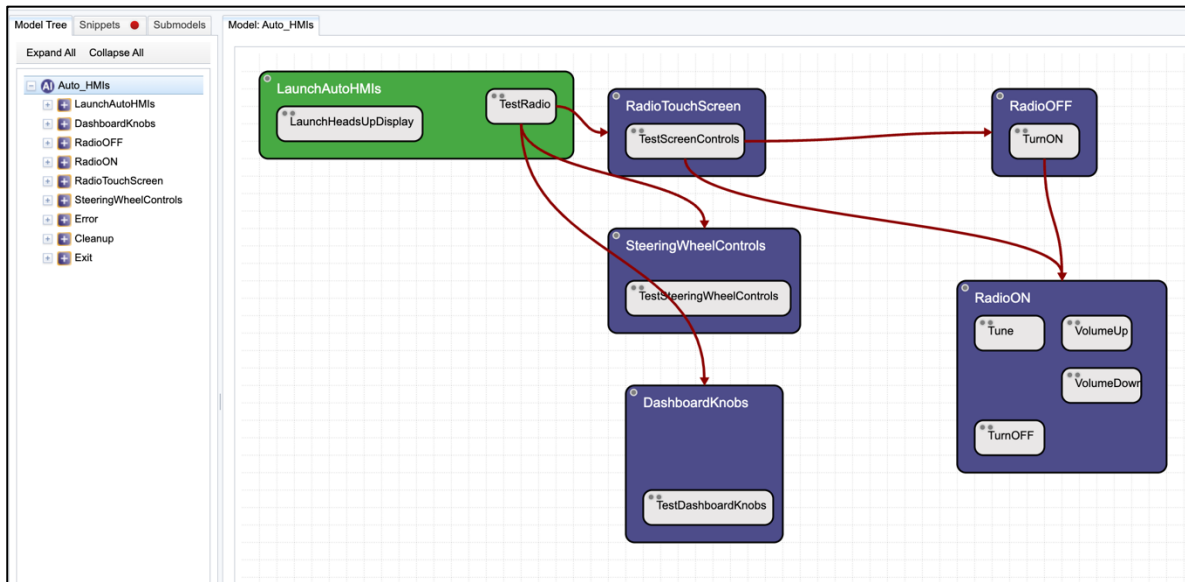


Figure 10. Part of an HMI model with states and actions mapped

That completes an example of how to model a portion of a car HMI. You can continue adding states, actions, and connections to expand your model. Proceed to the next section to run the model.

Running a Model

After you create your model, you can do a “dry run” of it. A “dry run” runs a test without running any snippets (“scripts”). This enables you to test the user journeys (or pathways) through a model, like the ones we just mapped.

1. In the DAI web application, choose **Designer > Models**. The list of models known to your DAI Server displays. If you have many models, you can search for the one you want using the search bar at the top of the page.

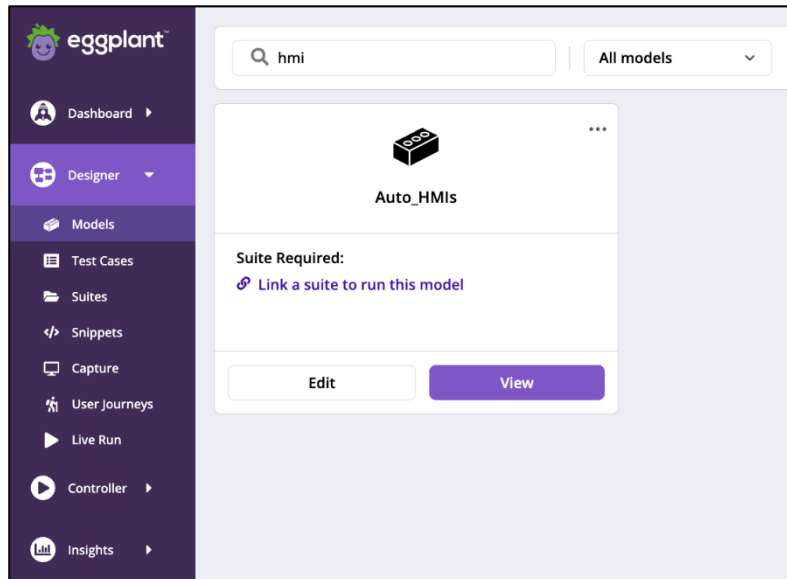


Figure 11. DAI Models page filtered for an HMI model

2. When you find your model, click **View**. The model opens in a preview mode.

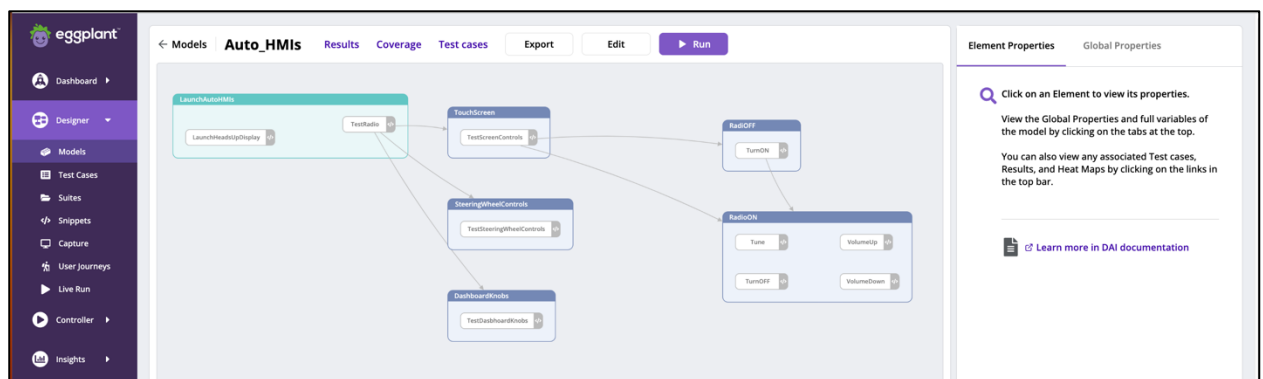


Figure 12. DAI Model open in Designer preview mode

3. Click the **Run** button at the top of the preview. The Live Run model page opens. This page lets you specify how you want to run your model. The defaults are **Model Based**, your selected model, exploratory test mode, and a dry run (without snippets). See **Live Run** in the DAI documentation for more information.

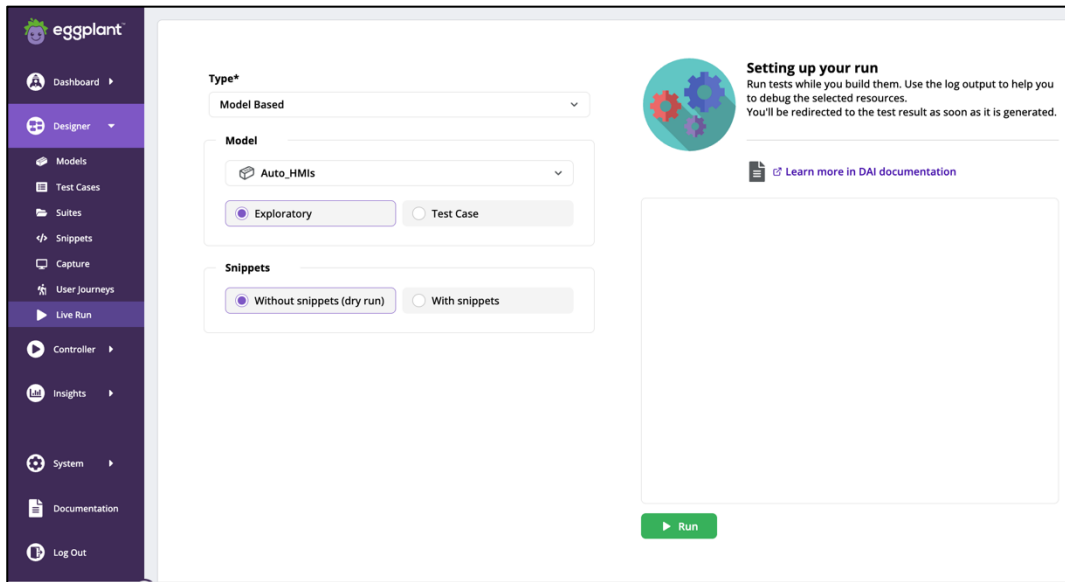
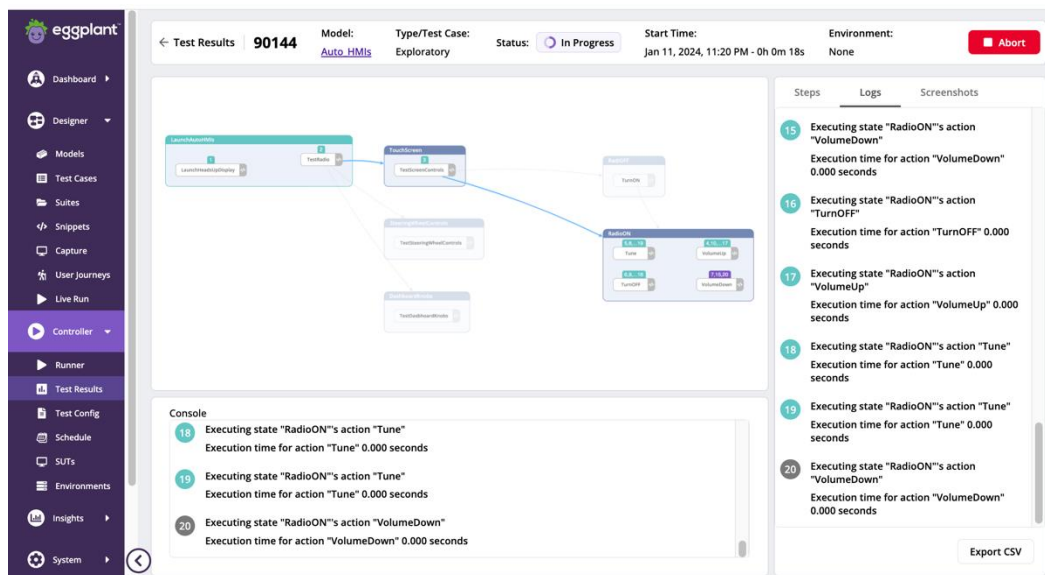


Figure 13. The Run model configuration page

4. Click the **Run** button at the bottom right of the Run model configuration page. DAI begins a “dry run” of the model, randomly exploring its pathways. DAI logs its progress to the console below the model preview and the steps completed to the right.



We only mapped a few paths through radio HMIs. If you model more of the radio and other auto HMI user journeys—creating more states, actions, and connections—you can create more paths for DAI to explore.

Please see the [DAI documentation](#) for more information about modeling with DAI, including how to run with snippets.

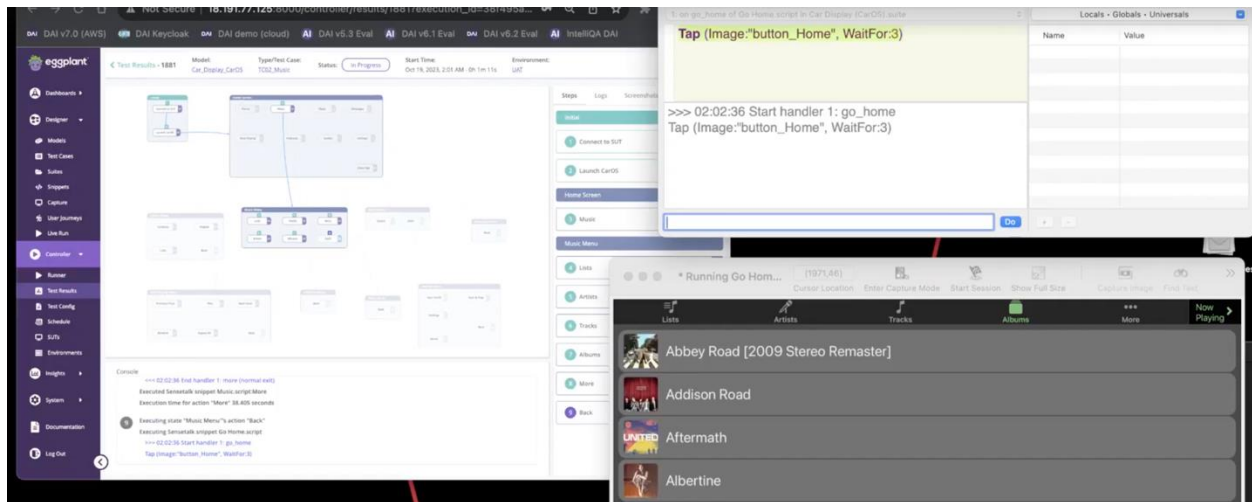


Figure 15. DAI running auto HMIs model

Above is an example of an HMI model. In this case, the model runs snippets against a music app in an auto on a connected music app in an auto.

To learn more about Eggplant products and solutions, please reach out to [us](#) today!



Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.

This information is subject to change without notice. © Keysight Technologies, 2014, Published in USA, January 29, 2023, 3124-1064.EN