PEGA PLATFORM

Pega Robotic Automation
Pega Robot Studio

Using the Robot Inspector

19.1

PEGA

Trademarks

Notices

Pegasystems Inc.
One Rogers Street
Cambridge, MA 02142-1209
USA
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pega.com

Updated: October 11, 2019

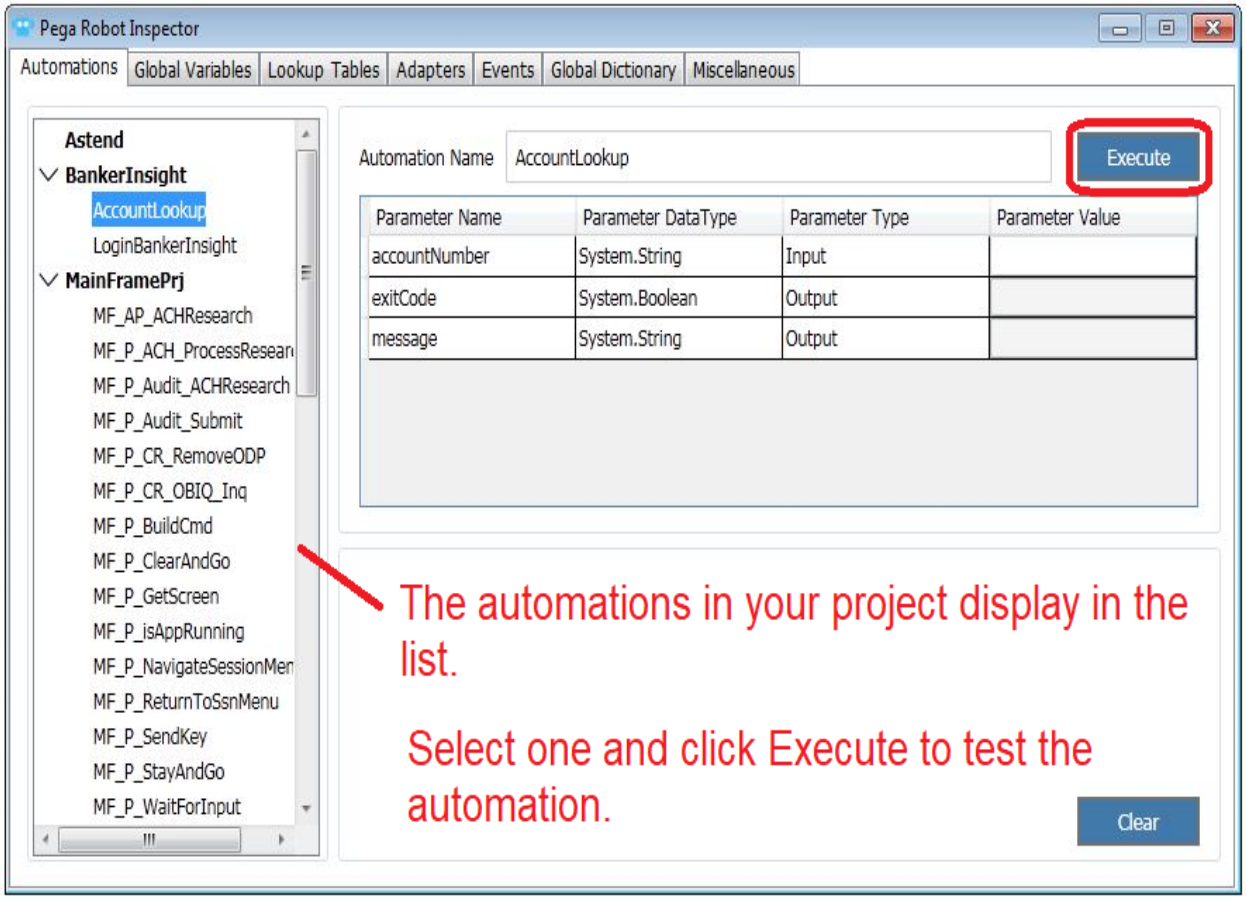
Feedback

If you have suggestions or comments for how we can improve our materials, send an email to Robotics-Documentation@pega.com.

CONTENTS

Using the Pega Robot Inspector

The Pega Robot Inspector provides a user interface that helps you understand the current state of your automation solution while it is running. It provides visibility into global values, including lookup tables and real-time visibility of matching. It also allows you to do targeted testing of discreet automations within your solution, while simulating input parameters and giving visibility into output parameters. This allows you to test a complete workflow or individual parts of a workflow without completing the entire use case or process.



Even for simpler solutions, it might be necessary to run through several automations that you know are working before you can get to the area you want to test. By testing pieces individually, you don't have to run through everything in your solution whenever you want to test one change or new piece of functionality.

The Robot Inspector provides a way to test solutions without having to build a separate testing interface. The user interface (UI) provides comprehensive functionality to help test most of the items within typical solutions.

The Robot Inspector allows you to launch automations and includes a live view of (and the option to edit) internal data structures (global variables, lookup tables, and the global dictionary). You can start and stop adapters and view the object hierarchy of an adapter to watch controls match and unmatch.

You can select control events to be logged for troubleshooting adapter issues.

Enabling the Robot Inspector in RuntimeConfig.xml

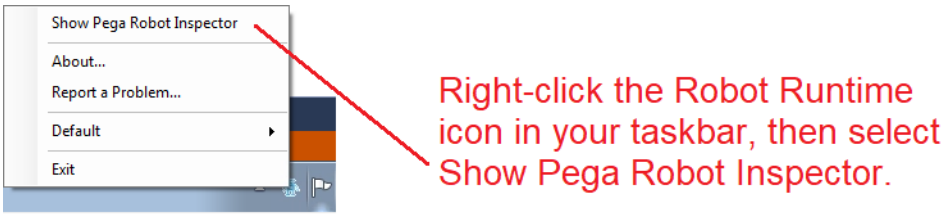
Enable the Robot Inspector by searching for the **LoadRobotInspector** key in the *RuntimeConfig.xml* file. Set the **Show** value to **true** for this element.

If the **LoadRobotInspector** key is not in your *RuntimeConfig.xml* file, add the line to the RuntimeTrayMenu section, as shown in the following example:

```
<RuntimeTrayMenu>
  <MenuItem item="LoadLocalProject" label="Load &amp;Local Project..." show="true" />
  <MenuItem item="LoadWebProject" label="Load &amp;Web Project..." show="true" />
  <MenuItem item="Unload" label="&amp;Unload" show="true" />
  <MenuItem item="Separator" show="true" />
  <MenuItem item="RecentProjects" label="&amp;Recent Projects" show="true" />
  <MenuItem item="LoadMostRecent" label="Load &amp;Most Recent Project" show="true" />
  <MenuItem item="LoadRobotInspector" label="Show Pega Robot Inspector" show="true" />
  <MenuItem item="Separator" show="true" />
```

Launching the Robot Inspector

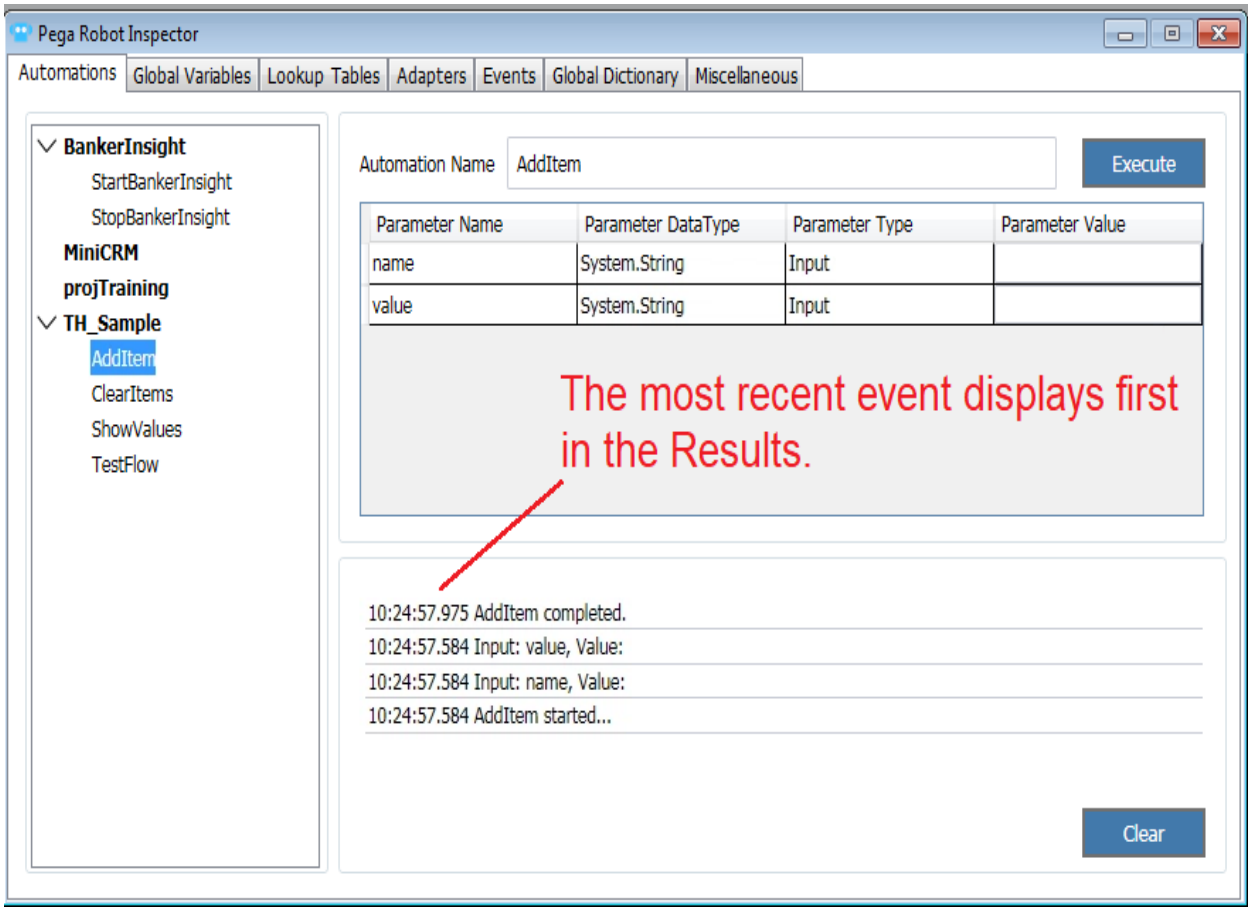
When you launch the Robot Inspector, it automatically discovers the projects that you have in the open solution. So, open and start your project; then load the Robot Inspector.



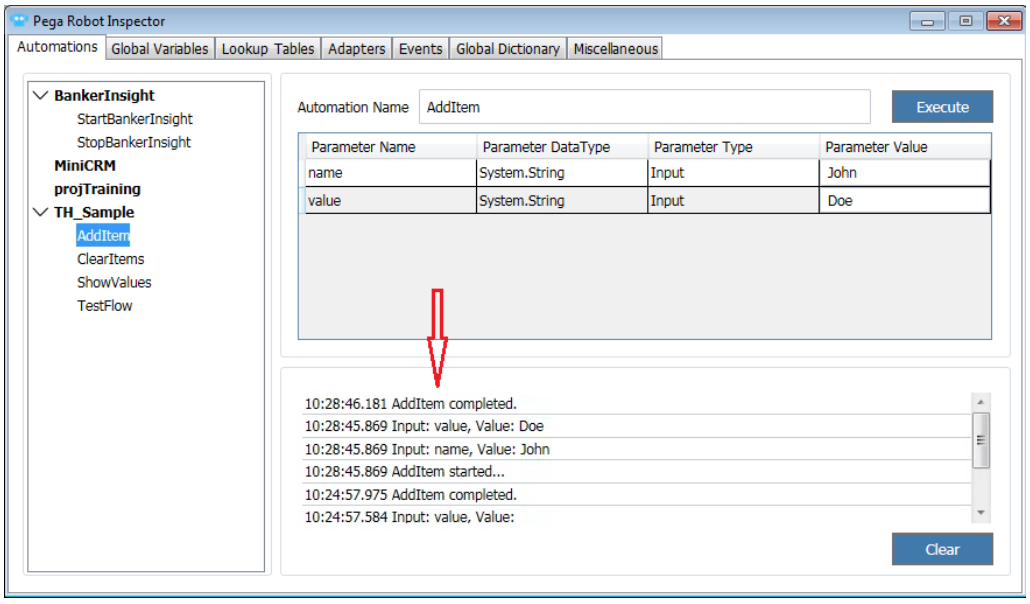
Using the Automations tab

The **Automations** tab allows you to select automations within your solution and run them. This capability makes it much easier to unit test your solution at run time. The tab lists automations, separated by project, that contain entry points and allows you to set input parameters if applicable. If an automation doesn't have an entry point, it does not display in the list of automations.

To run an automation, highlight it in the list on the left and click **Execute**. If input parameters are required, you are prompted to enter them. The dialog remembers the parameters you entered previously in the current session.



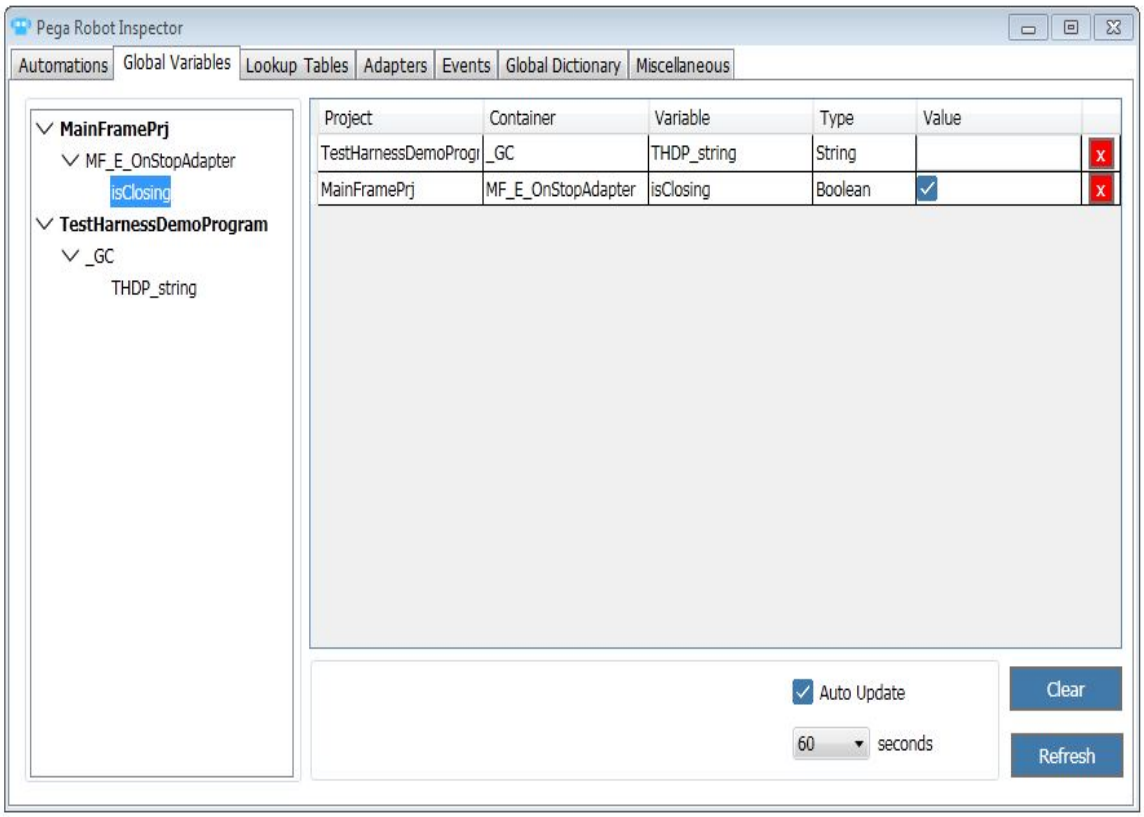
When you enter a Parameter Value and execute the automation, you see the entered values in the Results list.



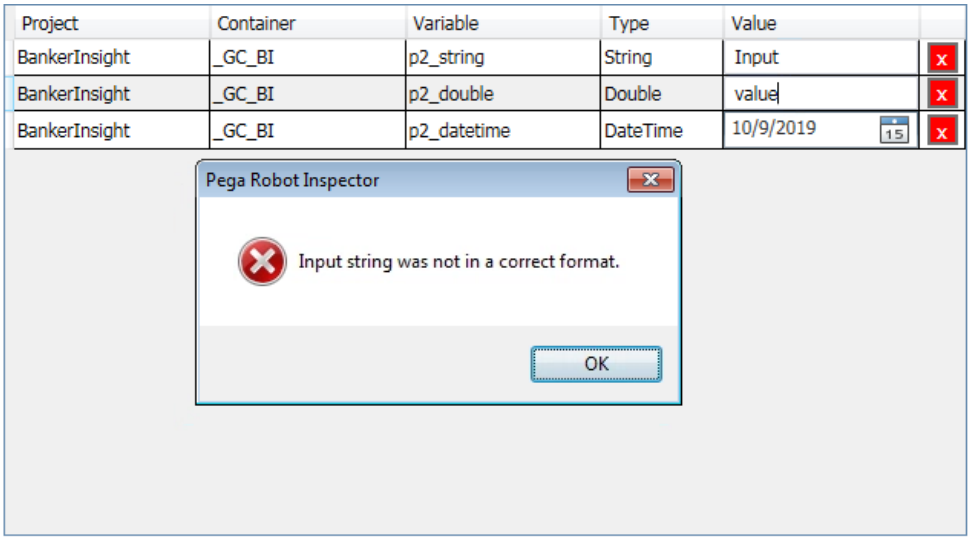
Using the Global Variables tab

Use the **Global Variables** tab to monitor and change global variables within your solution at run time. You can add values to the watch list on the **Global Variables** tab, and then execute a process from the **Automations** tab to verify that the automation is updating the value as expected. You can see what the values are for all your global variables and modify them if needed. You can set values to known values, or test to see how your automations modify the Global Variable values.

This tab allows for viewing and editing global variables. It is different from the Automation Locals or Automation Watches window in Robot Studio, because you can view the values at run time without being at a breakpoint. Also, you can modify the values. When you no longer want to view a variable, remove it from the list by clicking the red X next to it.



Note that the Robot Inspector still tests for valid inputs, based on the control type.



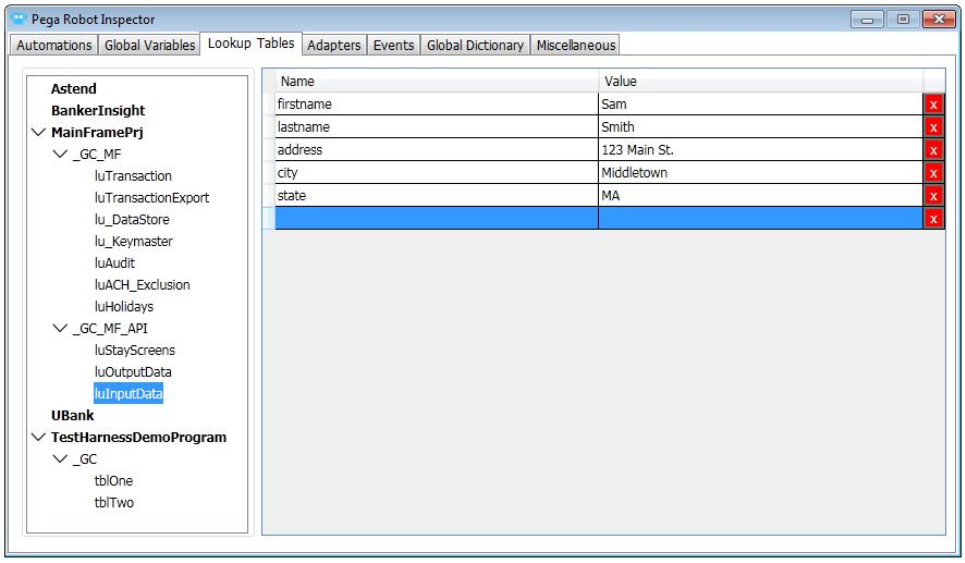
Auto Update

The changes on this tab are not live, but updated either on the interval selected or by clicking **Refresh**. The Interval in the Auto Update area is the time in seconds before Robot Studio polls the data to see if an update is needed. You can uncheck the **Auto Update** box to disable this feature. Click **Refresh** to update information on this tab.

Using the Lookup Tables tab

Lookup tables are global values within your solution that contain a data structure too complex to be represented on the **Global Variables** tab. Within an automation, lookup tables can store any table data and are commonly used to store information read from an Excel spreadsheet, results from a database query, or key value pairs that represent configuration data within a solution.

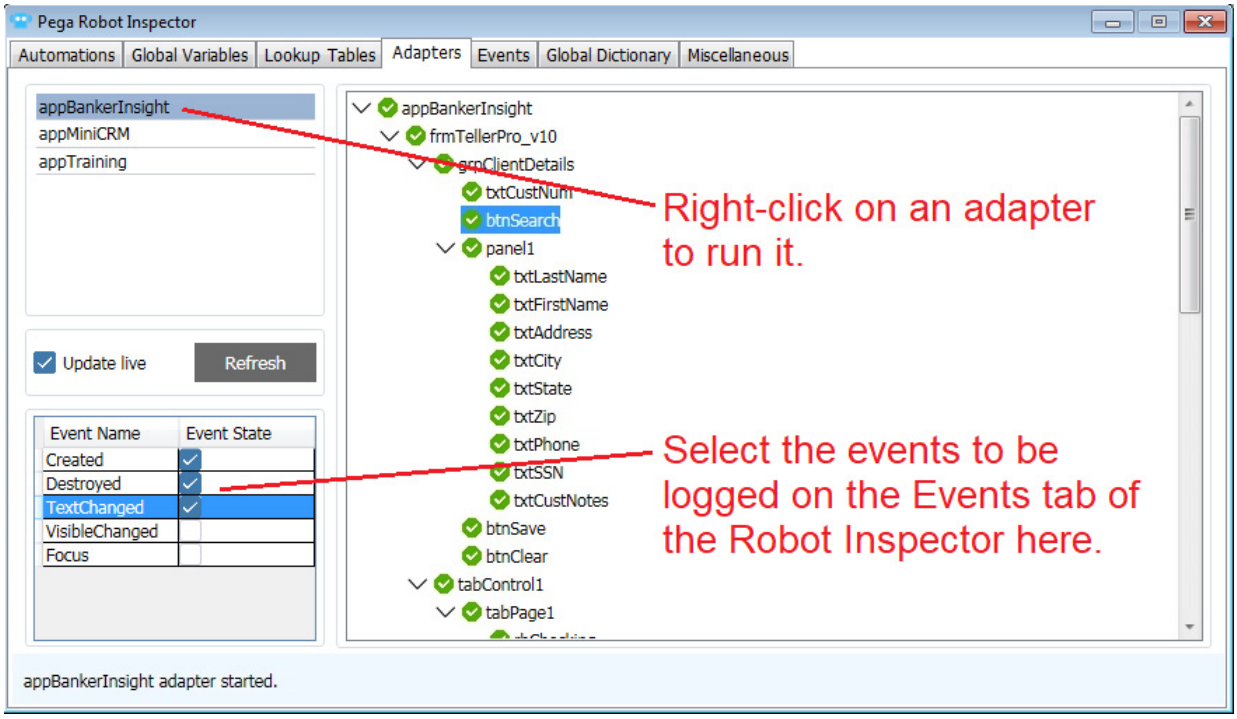
On the **Lookup Tables** tab, you can view and edit the contents of your lookup tables. You can also import or export XML or CSV files, to load or extract data to or from a table at run time. Editing data is done by changing a cell value and then clicking outside of that cell. You can add new records to a table as well.



Using the Adapters tab

The **Adapters** tab lists all of the adapters used in your solution and provides a live view of the matching status for each target control the adapter contains. You can start or stop an adapter to test how it runs. You can log several key events to the **Events** tab (for example, Created, Destroyed, TextChanged) to see the order these happen and learn more about application behavior.

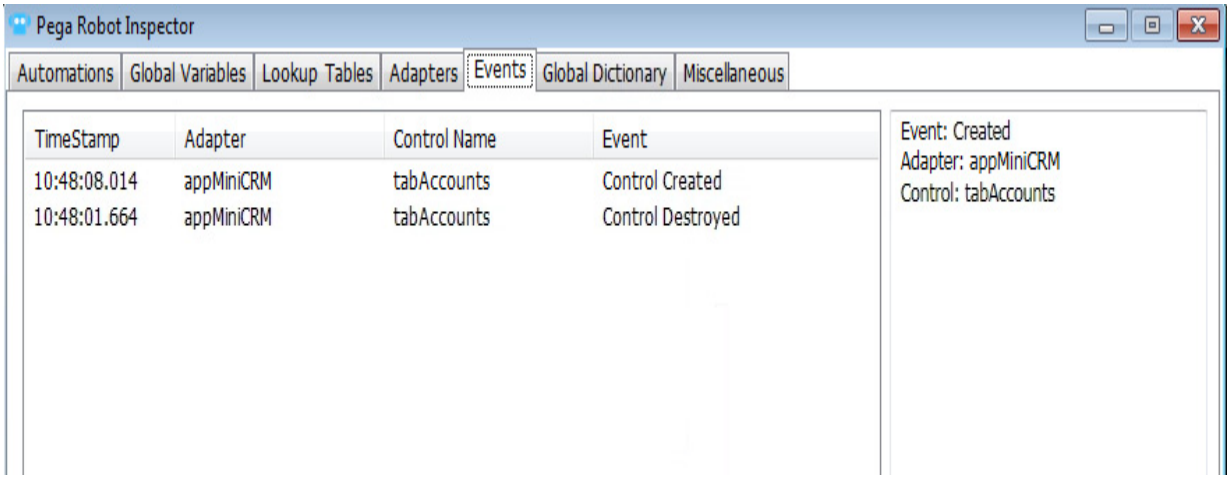
To track events, click an adapter or adapter control and select which events you want to be logged.



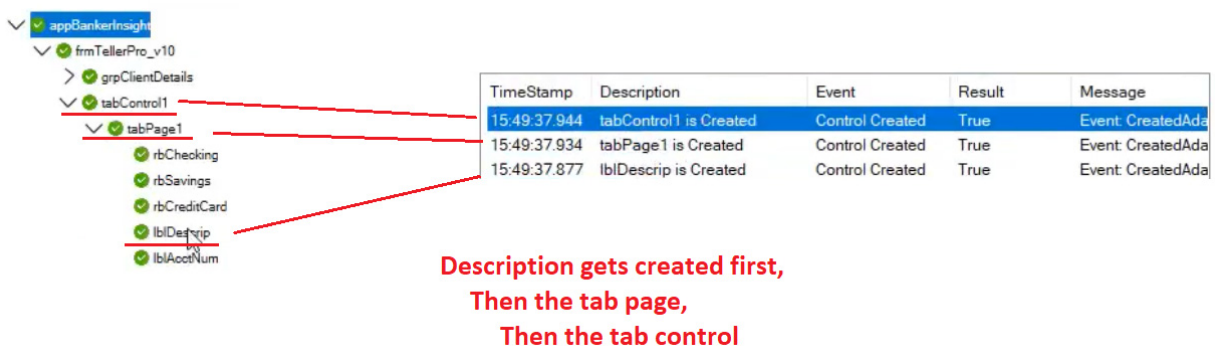
Using the Events tab

On the **Events** tab, you can see when a control or group of controls gets created and destroyed while your automations are running, and the order of these events.

You can specify the events that are logged on the **Events** tab of the Robot Inspector. (You specify the controls and events to log on the **Adapters** tab.) This can help you determine when a control is created or destroyed, or when a data value gets modified.



In Pega Robotic Automation, controls are created from the lowest to the highest in the object hierarchy once you navigate to the page or screen they are on.

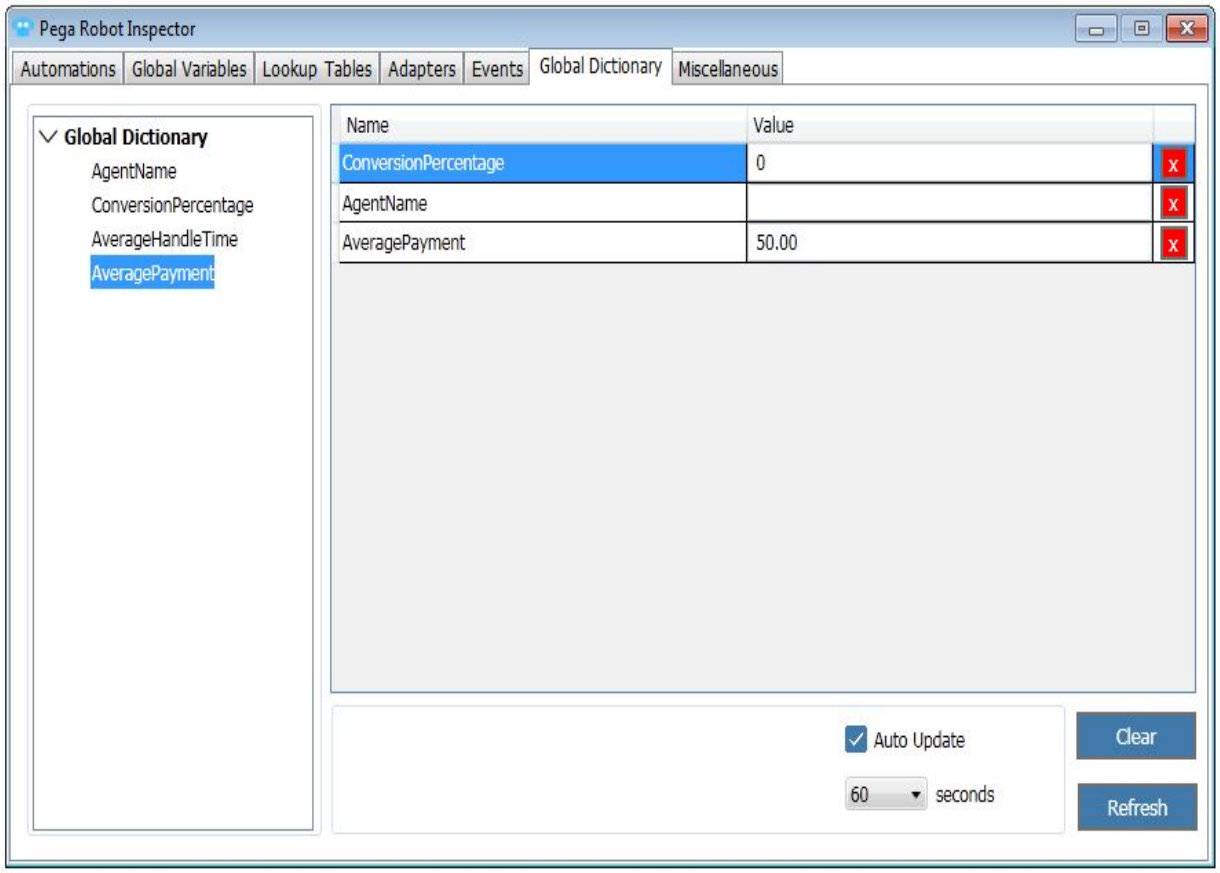


Using the Global Dictionary tab

Similar to the **Global Variables** tab, this tab is used to view and edit Global Dictionary variables used within the Interaction Framework. This tab is not displayed in the Robot Inspector if your loaded solution does not contain both a Global Dictionary component and an Interaction Manager component.

On the **Global Dictionary** tab, you can view the values at run time without being at a breakpoint, and you can modify values. When you no longer want to view a variable, remove it from the list by clicking the red X.

When you add a Global Dictionary component to a solution, the Global Dictionary variables listed in the *`Interaction.xml`* file are available for viewing and editing. Global Dictionary variables can be used across projects and are visible to any solution that has a Global Dictionary component.



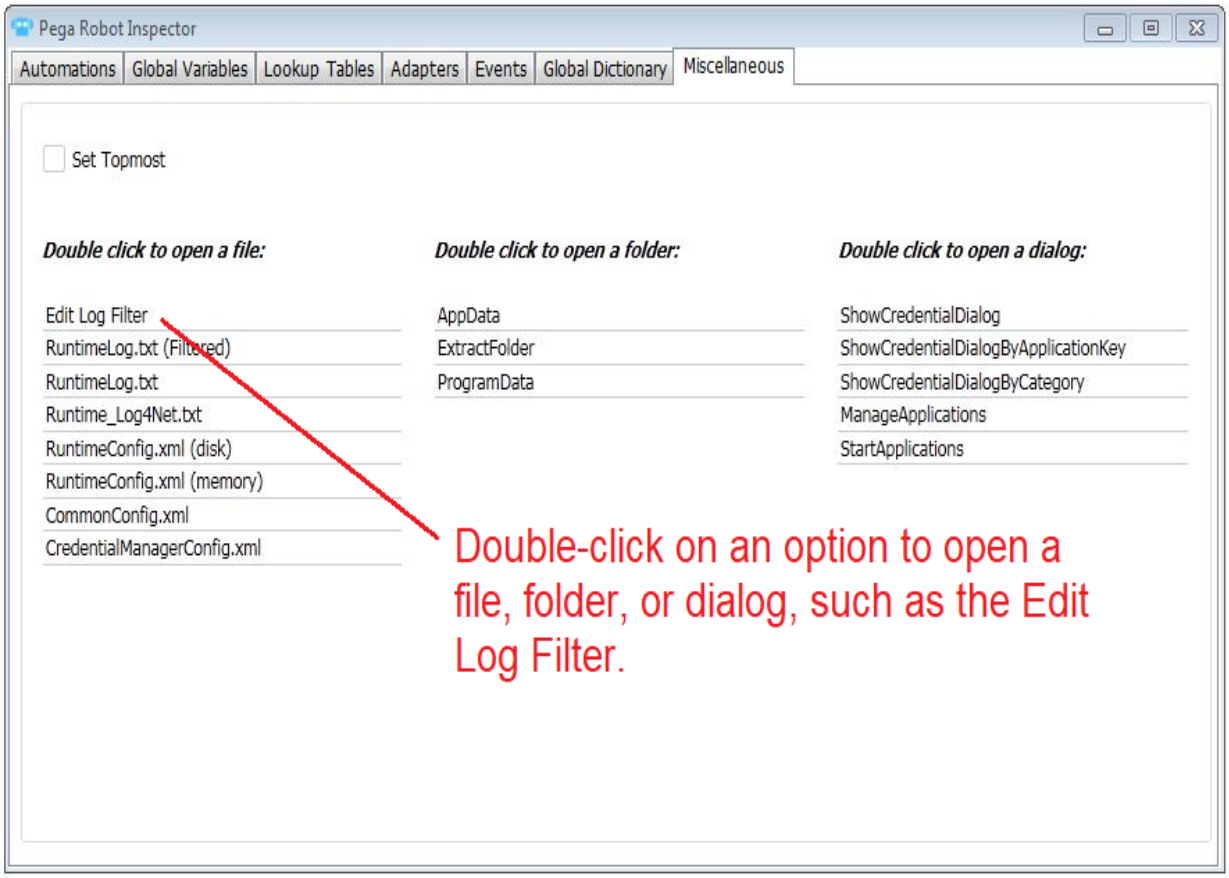
Values that are modified in the Robot Inspector are available to all projects in the entire solution.

Auto Update

The changes on this tab are not live, but updated either on the interval selected or by clicking **Refresh**. The Interval in the Auto Update area is the time in seconds before Robot Studio polls the data to see if an update is needed. You can uncheck the **Auto Update** box to disable this feature. Click **Refresh** to update information on this tab.

Using the Miscellaneous tab

Use the **Miscellaneous** tab to get access to configuration and logging files stored on your system, folders that are commonly used when testing, and out-of-the-box dialogs used for Start My Day and Assisted Sign On implementations.



Check the **Set Topmost** option to have the Robot Inspector always display in front of Robot Studio.

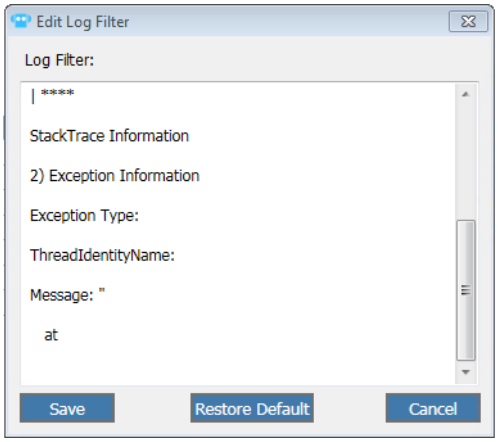
Double-click to open a file

From the Robot Inspector, you can open these files, which are configuration or log files that are commonly accessed when testing and debugging automations.

Edit Log Filter

The system provides the ability to apply filters to Robot Runtime logs that make debugging quicker and easier. You can edit the filters to satisfy your needs by customizing the log filter file. Do this by selecting 'Edit Log Filter' and updating the file. Note that you are creating a log filter file that persists between run-time sessions. You can revert back to the preset filters by clicking **Restore Default**.

The Log Filter provides a list of string values that, when matched in the *RuntimeLog.xml* file, cause a log line to be included in the output file. Each entry in the log filter must be followed by a blank line. The default filter lists log lines that match the Yellow lines (ExecutionLink), the Blue lines (propagating), control created events (Control is matched), control destroyed events (Control is detached), key lines in an Exception link, Script log lines, and any user-added log entries (log entries with four asterisks in a row ****).



Double-click to open a folder

Double-click these options to open frequently used folders:

- **AppData:** *RuntimeConfig.xml*, Runtime logs
- **ExtractFolder:** Projects are extracted here and stored while being run
- **ProgramData:** *CommonConfig.xml*, *CredentialManagerConfig.xml*, and other machine-based configuration files.

Double-click to open a dialog

If you have an ASO Manager or StartMyDay component in your solution, it displays here.

When using the ASO Manager and StartMyDay components, you can use these links to access the available dialogs while testing, without the need to build a user interface for that purpose. This often happens when starting a new project.