



ScenarioBuilder

User Guide

Contents

WELCOME TO SCENARIOBUILDER	8
SCENARIOBUILDER FEATURES	8
ABOUT THIS GUIDE	9
END USER LICENSE AGREEMENT	10
BASIC TERMINOLOGY	12
SCENARIO DESIGN BASICS	13
DESIGN OUTLINE 1 – AIRLINE TICKET WEBSITE	13
DESIGN OUTLINE 2 – SEARCH FOR AN ACCOUNT IN CRM	15
DESIGN OUTLINE 3 – ENTER HOURS IN A TIMESHEET APPLICATION	16
LAUNCH SCENARIOBUILDER	18
PROJECTS	19
MANAGE PROJECTS FROM SCENARIOBUILDER SPLASH SCREEN	19
OPEN PROJECTS.....	20
CREATE NEW PROJECT	20
MANAGE PROJECTS FROM SCENARIOBUILDER	22
CREATE NEW PROJECT	23
SWITCH PROJECTS.....	24
MOVE PROJECT	25
COPY PROJECT TO	25
DELETE PROJECT	26
PROJECT PROPERTIES	28
EXPORT PROJECT	29
IMPORT PROJECT	30
LOAD PROJECT FROM VERSION CONTROL	31
PROJECT OPTIONS	32
NEW SCENARIO	33
START RECORDING.....	34
OPEN WEB APP	34
START APPLICATION	35
MANUAL SCRIPTING.....	35
USER INTERFACE	36
DEFAULT LAYOUT.....	36
HOME TOOLBAR	38



VIEW TOOLBAR	41
RUN TOOLBAR	42
HELP TOOLBAR.....	43
SCENARIO RECORDER.....	45
CAPTURING IMAGES.....	46
SCENARIO RECORDER MENU	48
CREATING YOUR FIRST SCENARIO WITH SCENARIO RECORDER.....	51
IMPORT AND EXPORT SCENARIOS	53
EXPORT SCENARIO	53
IMPORT SCENARIO	54
ACTIONS	56
FAVORITES.....	56
APPLICATION	58
START APPLICATION	58
CLOSE APPLICATION.....	59
BEGIN TRANSACTION	59
END TRANSACTION	60
EXECUTE COMPONENT.....	60
DEFINE COMPONENT.....	61
RUN COMMAND LINE	62
OPEN WITH FILE ASSOCIATION	62
WEB SERVICE CLIENT.....	63
OPEN URL.....	65
PLAY SCRIPT.....	65
PLAY SCRIPT SECTION.....	66
IMAGES	67
CLICK ON IMAGE	67
FIND IMAGE.....	70
FIND LINKED IMAGES	74
FIND TEXT IN IMAGE	75
TAKE A SCREENSHOT.....	78
KEYBOARD.....	79
KEY COMBINATION.....	79
TYPE TEXT	80
FUNCTION KEYS	81
MOUSE.....	82
MOUSE CLICK	82
MOUSE MOVE AND CLICK	82



MOUSE MOVE	83
WINDOW	84
WINDOW ACTIONS.....	84
MOVE WINDOW	85
RESIZE WINDOW	86
ALL WINDOWS ACTION	87
MS EXCEL	89
DEFINE A WORKSHEET	89
FOREACH LINE.....	90
GOTO LINE	90
READ LINE	91
WRITE LINE.....	91
MANAGE PASSWORDS	92
GENERATE PASSWORD	92
SAVE PASSWORD.....	93
RETRIEVE PASSWORD	93
CONTROL ACTIONS	94
IF	94
LOOP.....	96
SWITCH.....	96
CASE	97
CLIPBOARD ACTION	97
READ TEXT INTO A VARIABLE.....	98
RENDEZVOUS POINT	100
EXIT	100
FUNNEL POINT	100
LABEL.....	101
ON FAILURE.....	101
RANDOM ACTION	102
RESUME	102
FIND CSV LINE	102
GO TO	103
RAISE FAILURE	103
WRITE TO FILE	103
VARIABLES.....	105
SET STRING VARIABLE	105
SET DATE VARIABLE	106
SET TIME VARIABLE.....	107
SET NUMBER VARIABLE.....	108
STRING FUCTIONS	109
DATE FUCTIONS	112
TIME FUCTIONS.....	113



INCREMENT A VARIABLE	114
LOAD AN IMAGE INTO A VARIABLE	114
FILES	116
FILE EXISTS	116
FILE UPDATE DATE	117
FILE SIZE	117
FILE VERSION.....	118
MISCELLANEOUS	119
TIMER.....	119
COMMENT	120
DIALOG BOX.....	120
PAUSE EXECUTION	121
WAIT.....	121
CALL DLL	121
LINE FILLER	122
READ SERIAL PORT	122
WRITE SERIAL PORT	122
COMPONENTS	124
DEFINE COMPONENTS	125
SAVE COMPONENTS	126
DELETE A COMPONENT	127
COPY A COMPONENT TO ANOTHER PROJECT	128
EXPORT AND IMPORT COMPONENTS	129
EXPORT COMPONENT	129
IMPORT COMPONENT	130
CONVERT COMPONENTS TO SCENARIOS	133
PROJECT SCRIPTS	134
PLAY SCRIPT	134
WORKFLOWS	137
CLIPBOARD	138
SCENARIO EXPLORER	139
WORKING WITH IMAGES.....	140
THINK LIKE A REAL USER	141
IMAGE ACTIONS KEEPS SCENARIOS SYNCHRONIZED	141



- COMBINE IMAGE AND LOOP ACTIONS TO PAGE DOWN UNTIL IMAGE IS FOUND 142
- WAIT FOR IMAGE TO DISAPPEAR BEFORE ADVANCING 145
- IMAGE VARIATIONS 146**
 - USE MULTIPLE IMAGES FOR AN IMAGE ACTION 146
 - ADDING ALTERNATE IMAGES THROUGH THE SCENARIO RECORDER..... 147
 - SEARCH FOR IMAGE IN BLACK AND WHITE 147
 - IMAGE SEARCH AREA..... 149
 - PLAY WITH IMAGE RETAKE 150
 - PLAYBACK SCREENSHOT 152
 - SMART CONFIGURATOR 153
- OCR CONFIGURATOR..... 158**
- CONDITIONAL STATEMENTS..... 161**
 - “IF” ACTION 161**
 - “ IF “ ACTION WITH CSV FILE 162
 - “ IF ” ACTION WITH EXECUTION STATUS VARIABLE 164
 - CASE/SWITCH..... 165**
- WORKING WITH SCENARIO PARAMETERS..... 167**
 - ADDING PRE-DEFINED VARIABLES TO A SCENARIO 167**
 - SET A VARIABLE 167
 - INCREMENT A VARIABLE..... 169
 - EXECUTION STATUS VARIABLES 170**
 - USING A CSV FILE TO STORE VALUES FOR SCENARIO PARAMETERS 171**
 - CREATE A NEW SCENARIO PARAMETER CSV FILE 171
 - EDIT AN EXISTING SCENARIO PARAMETER CSV FILE..... 173
 - ATTACH A SCENARIO PARAMETER CSV FILE 173
 - ADD SCENARIO PARAMETERS 174
 - LAST LINE USED IN SCENARIO PARAMETER CSV FILE 175
- ON FAILURE 176**
- ACTION CONTEXT MENU 177**
- SCENARIO PROPERTIES 180**
- SCENARIO PLAYBACK 183**
 - EXECUTION REPORT 183**
- RUNNING SCENARIOS FROM THE COMMAND LINE 185**



OPTIONS	188
WORK SPACE OPTIONS	188
BASE OPTIONS	189
GLOBAL HOTKEYS OPTIONS.....	189
GENERAL OPTIONS	190
RECORDER OPTIONS	191
IMAGES	192
GETTING HELP AND SUPPORT	193
SEND US FEEDBACK	193



Welcome to ScenarioBuilder

ScenarioBuilder from Automai is an easy-to-use tool that simplifies test development. Used in conjunction with AppLoader, AppsWatch, AppVerify, ScenarioBuilder makes it easy to create a wide range of test scenarios for your testing needs.

With ScenarioBuilder's intuitive point and click user interface, users of all skill levels can create any sequence of end user actions, which can be played back in test scenarios. No coding or complex scripting languages are required. Scenario Recorder captures mouse clicks and keystrokes in real time, further simplifying the process. As a result, you can develop even the most complex test scenarios in a fraction of the time required with other load testing and application monitoring solutions.

ScenarioBuilder also features a handy playback tool that allows you to verify your test scenario right on your desktop. You can test your scenario along the way, as well as verify the final scenario before sending it to the Controller/Base for execution in the test environment.

[Click here to watch ScenarioBuilder introduction video guide.](#)

ScenarioBuilder features

With ScenarioBuilder, you can emulate real user activities, including mouse events, keyboard inputs, and more. Here's a sampling of the user *Actions* you can create in a test Scenario:

- Type text, including function keys
- Mouse moves and clicks
- Move and resize windows
- Find and click on an image (for Windows objects that do not have a "handle", ScenarioBuilder's image Actions provide the flexibility to navigate to virtually anywhere on the desktop)
- Define a specific image "Search Area" on the desktop

In addition to basic user Actions, ScenarioBuilder includes advanced conditional statements and variables, so you can easily create complex scenarios without any scripting:

- If/Then/Else statements
- Loops
- Case/Switch statements



- Variables retrieved from a CSV file allow you to change values with each scenario execution
- Determine Actions to take if your scenario is interrupted by timeouts, pop-ups, or other failures

ScenarioBuilder's tools and options provide flexible, user friendly functionality

- Create *Components* to encapsulate Action sequences for reuse throughout the current, or in future scenarios
- Record Actions in real time with Scenario Recorder
- Edit and update recorded scenarios in either "Graphical" or "Tree" views

About this guide

Designed for both the novice and the experienced user, this guide will introduce you to ScenarioBuilder's features, user interface, toolbars, layout and also walks you through all the steps involved in creating test scenarios.

First time users can also review the [Getting Started](#) document to learn how to create a basic test scenario.

This guide covers building test scenarios, playing them back for verification, and sending test scenarios to the Controller/Base.

For more details on executing test scenarios from AppLoader, AppsWatch, AppVerify and RPA applications, gathering and interpreting results from them, please refer to the [AppLoader](#), [AppsWatch](#), [AppVerify](#) and [RPA](#) User Guides.



End user license agreement

IMPORTANT -- PLEASE READ CAREFULLY BEFORE USING THIS SOFTWARE PRODUCT: This End-User license Agreement ("EULA") is a contract between (a) you (either an individual or the entity you represent) and (b) Automai, Inc. ("AUTOMAI") that governs your use of the software product ("Software").

RIGHTS IN THE SOFTWARE ARE OFFERED ONLY ON THE CONDITION THAT YOU AGREE TO ALL TERMS AND CONDITIONS OF THIS EULA. BY INSTALLING, COPYING, DOWNLOADING, OR OTHERWISE USING THE SOFTWARE, YOU AGREE TO BE BOUND BY THIS EULA. IF YOU DO NOT ACCEPT THIS EULA, YOU MUST DESTROY ALL COPIES OF THE SOFTWARE AND ACCOMPANYING DOCUMENTATION WITHIN 1 DAY OF PURCHASE FOR A REFUND FROM AUTOMAI.

1. **THIRD PARTY SOFTWARE.** The Software may include, in addition to AUTOMAI proprietary software ("AUTOMAI Software"), software under licenses from third parties ("Third Party Software" and "Third Party License"). Any Third Party Software is licensed to you subject to the terms and conditions of the corresponding Third Party License. Generally, the Third Party License is located in a file such as license.txt or read.me file; you should contact AUTOMAI support if you cannot find any Third Party License.

2. **LICENSE RIGHTS.** You will have the following rights provided you comply with all terms and conditions of this EULA: a. **Use.** AUTOMAI grants you a license to Use one copy of the AUTOMAI Software. "Use" means installing, copying, storing, loading, executing, displaying, or otherwise using the AUTOMAI Software. You may not modify the AUTOMAI Software or disable any licensing or control feature of the AUTOMAI Software. Additional restrictions on Use may appear in the User Documentation. You may not separate component parts of the AUTOMAI Software for Use. You do not have the right to distribute the AUTOMAI Software. b. **Copying.** You may copy the AUTOMAI software for back-up and archival purposes, provided any copy contains all the original software's proprietary notices.

3. **UPGRADES.** To Use AUTOMAI Software provided by AUTOMAI as an upgrade, update, or supplement (collectively "Upgrade"), you must first be licensed for the original AUTOMAI Software identified by AUTOMAI as eligible for the Upgrade. To the extent the Upgrade supersedes the original AUTOMAI Software, you may no longer use such AUTOMAI Software. This EULA applies to each Upgrade unless AUTOMAI provides other terms with the Upgrade. In case of a conflict between this EULA and such other terms, the other terms will prevail.

4. **TRANSFER.** a. **Third Party Transfer.** The initial end user of the AUTOMAI Software may make a one-time transfer of the AUTOMAI Software to another end user. Any transfer will include all component parts, media, User Documentation, this EULA, and if applicable, the Certificate of Authenticity. The transfer may not be an indirect



transfer, such as a consignment. Prior to the transfer, the end user receiving the transferred Software will agree to this EULA. Upon transfer of the AUTOMAI Software, your license is automatically terminated. b. Restrictions. You may not rent, lease or lend the AUTOMAI Software or Use the AUTOMAI Software for commercial timesharing or bureau use. You may not sublicense, assign or otherwise transfer the AUTOMAI Software except as expressly provided in this EULA.

5. PROPRIETARY RIGHTS. All ownership and intellectual property rights in the Software and User Documentation are owned by AUTOMAI and/or its suppliers and are protected by law, including applicable copyright, trade secret, patent, and trademark laws. You will not remove any product identification, copyright notice, or proprietary restriction from the Software.

6. LIMITATION ON REVERSE ENGINEERING. You may not reverse engineer, decompile, or disassemble the Automai Software, except and only to the extent that the right to do so is allowed under applicable law.

7. CONSENT TO USE OF DATA. Automai and its affiliates may collect and use technical information you provide in relation to (i) your Use of the Software or the AUTOMAI Product, or (ii) the provision of support services related to the Software or the AUTOMAI Product. All such information will be subject to AUTOMAI's privacy policy. AUTOMAI will not use such information in a form that personally identifies you except to the extent necessary to enhance your Use or provide support services.

8. LIMITATION OF LIABILITY. Under no circumstances shall AUTOMAI or its suppliers or resellers be liable to licensee or any other person for any special, incidental, indirect, or consequential damages, including damages for loss of goodwill, lost profits, work stoppage, business interruption, personal injury, computer failure, loss of privacy, or any and all commercial damages or losses, even if AUTOMAI or any supplier has been advised of the possibility of such damages. Further, in no event shall AUTOMAI's liability under any provision of this agreement exceed the license fee paid to AUTOMAI. Some states or other jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

9. U.S. GOVERNMENT RESTRICTED RIGHTS AND COMPLIANCE WITH EXPORT LAWS. This software is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause of DFARS 252.227-7013 or subparagraphs (c)(i) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. You will comply with all laws, rules, and regulations (i) applicable to the export or import of the Software, or (ii) restricting the Use of the Software.

10. RESERVATION OF RIGHTS. Automai and its suppliers reserve all rights not expressly granted to you in this EULA.



Basic terminology

The following terms are used in ScenarioBuilder:

Project: A project is a folder that contains the elements of your scenarios (Images, Variables, scenario XML files, Components and Documents). Upon installation, ScenarioBuilder creates the “Default Project” folder, where scenarios will be stored *unless* a new Project is created and “Switched To”. Creating a Project for each business application being tested not only keeps test cases organized, but allows you to share resources among related scenarios. For example, imagine a separate scenario needs to be created to test various features of an application. Each of these features uses a common icon. The time-saving benefits of sharing a single image for each scenario are two-fold; (1) the image can simply be “loaded” rather than “captured” each subsequent time it is needed; (2) in the event that the “Submit” button is modified or changed in any way, updating the image once will update it for all of the scenarios within the Project. To create a Project, click File→New→New Project or File→Project Manager→New Project. A Project folder will be created, with subfolders for Images, Variables, scenario XML files, Components and documents.

Scenario: A scenario is an XML file created with ScenarioBuilder. A scenario contains Actions that emulate the activity of a user interacting with an application. Scenarios can be called in other scenarios (known as *nesting*) with the “Play Scenario” Action.

Component: A Component is a user-defined shell within a scenario that contains a single Action or a series of Actions. Think of Components as containers for Actions related to a specific task in the overall process. Breaking up scenarios into Components makes for easier scenario maintenance, and can be helpful in conceptualizing a scenario design. Components are self-contained units that can be called from any scenario within a Project. Additionally, Components can be exported and imported to and from Projects. If a modification is made to a Component that is being used by multiple scenarios, the Component is updated for all scenarios.

Transaction: A Transaction is a section of a scenario that is marked for the purpose of measuring the response time of that section. For example, you can define a Transaction to be “login to CRM” where the beginning of the Transaction is defined when the CRM application is launched, and the end of the Transaction is defined when the CRM interface finishes loading on the screen. Many Actions may occur within a Transaction (such as finding the executable to launch the application, launching the executable, waiting for the login screen, typing in the login name and password, waiting for the application to load completely, etc.).



Action: An Action is a step in a scenario that represents a single event (such as a mouse click or an application launch). Actions are located in a pane on the left side of the ScenarioBuilder window, and can be added to scenarios by double-clicking or dragging into the Scenario Window.

On Failure: The “On Failure” section of a scenario determines what Action(s) to take when a scenario or Component fails. For example, if your application crashes during a test, “On Failure” Actions could shut down and restart the application and attempt to resume the scenario where the crash occurred. A “Raise Failure” Action used in conjunction with an “On Failure” will mark and report the crash as a failure. If “Raise Failure” is omitted, the scenario will still return a “success” status.

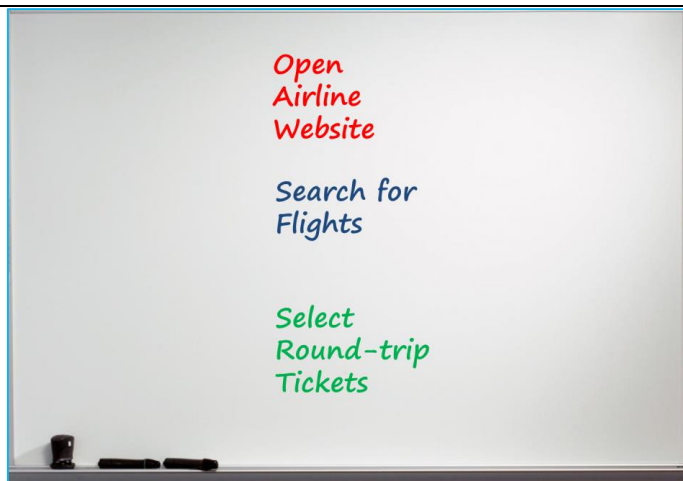
Response Time: Response time is the amount of time taken to execute a Scenario, Component, Transaction or step within a scenario. The system calculates the response time from taking the end point of your Scenario or Transaction and subtracting it from the beginning point of your respective Scenario or Transaction respectively.

Scenario design basics

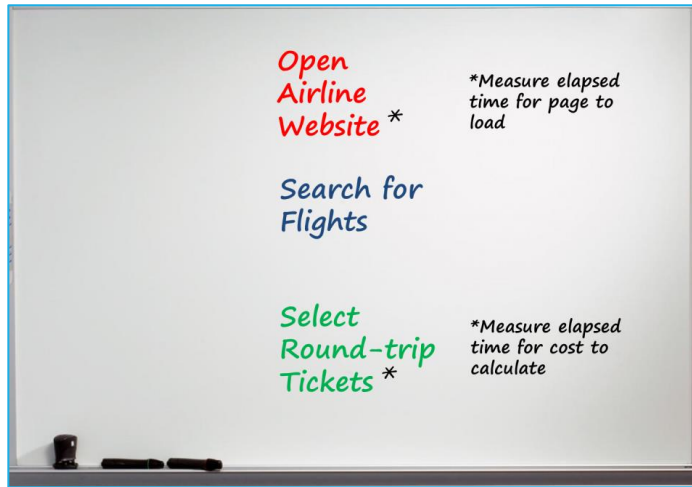
While it’s tempting to jump right in and begin building scenarios (in fact, we encourage you to do so, as it’s a great way to learn about ScenarioBuilder), before embarking on actual test case scenarios, we encourage you to step back and take a breath. The best practice is to start with a diagram outlining the business processes you want to test. The flow chart that you sketch will most likely become the blueprint for your scenario. Here are some examples of how to approach your scenario designs.

DESIGN OUTLINE 1 – AIRLINE TICKET WEBSITE

To create a scenario testing an airline’s website, for example, you might break out three general processes – *Open Airline Website*; *Search for Flights*; *Select Tickets*;



Now, say you want to know not only how long the processes take overall, but how long specific functions within the processes take. For example, the time the website takes to load, as well as the time it takes to calculate a “total trip cost” for the flights specified;



One final question needs to be answered before getting down to the business of building scenarios. Will the outlined business processes be re-used throughout the project or are they one and done? The answer to this question will help you to decide whether to break them up into Components or Scenarios:

Components	Scenarios
<p>Use Components to break up long scenarios into logical sections for editing and troubleshooting.</p>	<p>Use Scenarios for business processes that will be used multiple times throughout your test project (like logins).</p>
<p>Use Components for business processes that are not likely to be re-used in the project (one and done).</p>	<p>Scenarios can be called and played from other scenarios with the “Play Script” Action.</p>
<p>Use Components if a separate parameters value file is needed for a section of your scenario.</p>	



DESIGN OUTLINE 2 – SEARCH FOR AN ACCOUNT IN CRM

Here is another example, this one defines a test of a CRM application. The test breaks down into three segments – *Login to CRM application; Look up an account; Logoff of application*. Again, these segments establish the natural structure for the scenario. Notice the outline also includes the Actions (in plain terms) to be executed within each segment, further taking the guesswork out of scenario building:

<p>Login to CRM</p> <p>Actions include:</p> <ul style="list-style-type: none"> • Launch CRM App • Enter login info • Press enter • Wait for CRM home screen 	<p>If we assume that the Login and Logoff sequences will be used multiple times throughout the test project, it makes sense to create scenarios for them.</p> <p>And if we assume that the Lookup will only occur once in the project, we would use a Component to house that sequence.</p> <p>When working with your scenarios, use the Execute Component command to execute a component in your scenario and use the Define Component command to define the actions involved in it.</p>
<p>Lookup An Account</p> <p>Actions include:</p> <ul style="list-style-type: none"> • Click search icon • Enter account info • Press enter 	
<p>Logoff CRM</p> <p>Actions include:</p> <ul style="list-style-type: none"> • Click logoff • Confirm logoff • Wait for logoff screen 	



Take it one step further and add Transaction markers. If you want to know specifically how long it takes for the application to return the search results when a user looks up an account, insert a Transaction “Begin” and “End” Action in the Lookup segment:

<p>Lookup An Account</p> <p>Actions include:</p> <ul style="list-style-type: none"> • Click search icon • Enter account info • Begin Transaction • Press enter • Wait for CRM home screen • End Transaction 	<p>Use Begin Transaction to establish the starting point of your measurement, and End Transaction to mark the stopping point. Provide a meaningful name for Transactions as that name will be used in graphs and reports.</p>
--	---

DESIGN OUTLINE 3 – ENTER HOURS IN A TIMESHEET APPLICATION

This last example shows how to organize your scenario into a basic chart format using a spreadsheet. This example outlines a scenario for testing a Citrix-based timesheet application. There are three main tasks involved:

- Load Citrix and start the application
- Enter hours
- Close the application

Hierarchy	Scenario/Tasks	Type	Description	What to do on Failure?
1	Timesheet	Scenario	User enters timesheet data and submits for approval	Close TS application
2	Start TS	Component	Load Citrix and logon to the application	
3	Logon	Transaction	Click on signal button and wait until the first form displays	
2	Enter hours	Component	Select everyday of the week and enter the hours worked	Delete entered hours
3	Submit TS	Transaction	Click on submit TS button on wait for successful message	If resource busy, wait 2 seconds then return to submit again
2	Close TS	Component	Logoff and close TS	
3	Logoff	Transaction	Click on logoff link and wait for sign on screen to appear	



Each of the three main tasks in this example has been designated for Components. Additionally, the chart marks Transactions to measure the time it will take to perform key tasks within the Scenario:

- How long does it take to log on to the application?
- How long does it take to submit the hours? (i.e. receive a successful confirmation)
- How long does it take to log off from the application?

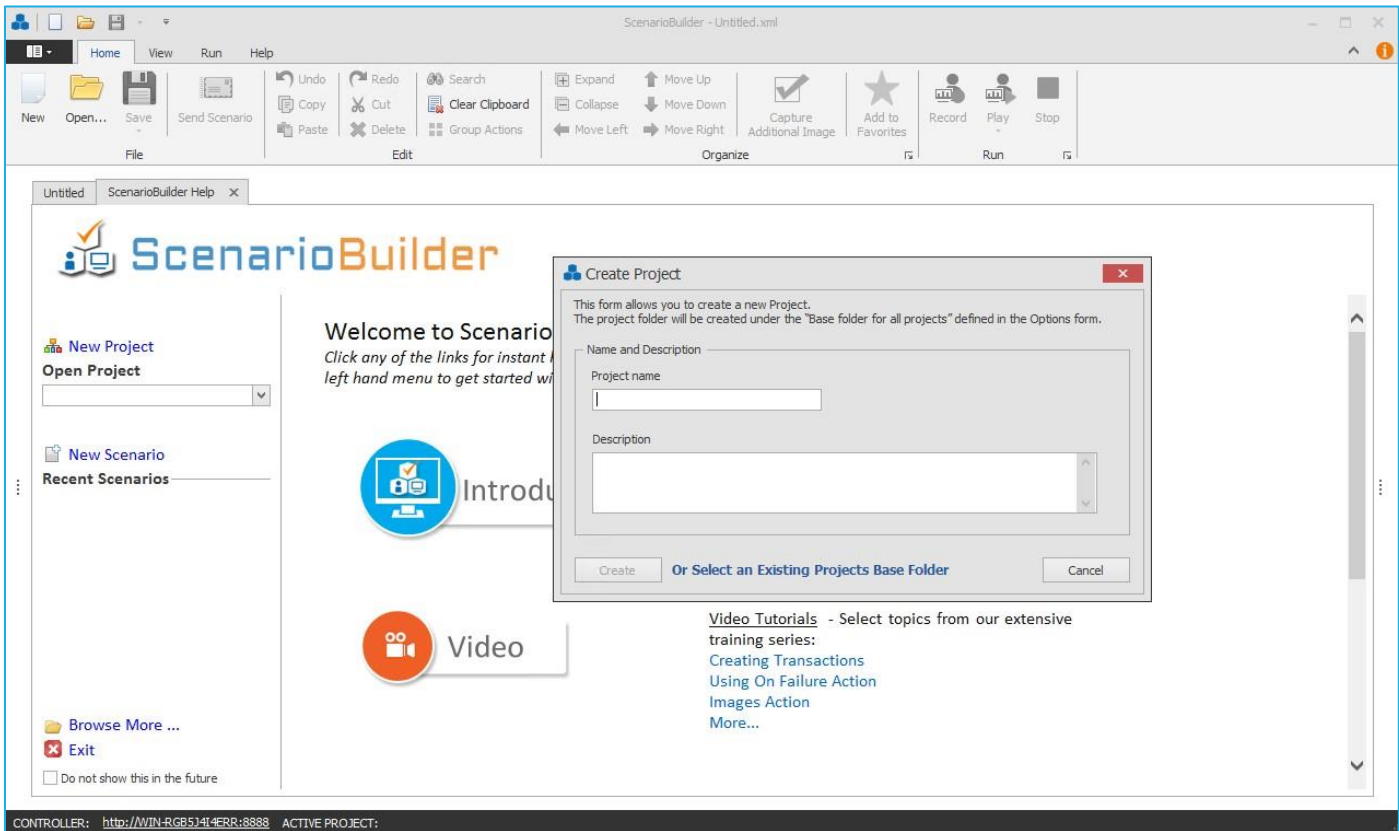


Launch ScenarioBuilder

The ScenarioBuilder “Splash Screen” greets you upon launching the program. Here, you may create a New Scenario, open an existing Scenario, or familiarize yourself with the common Actions used in ScenarioBuilder;



Double-click the ScenarioBuilder shortcut to launch the splash screen.



Projects

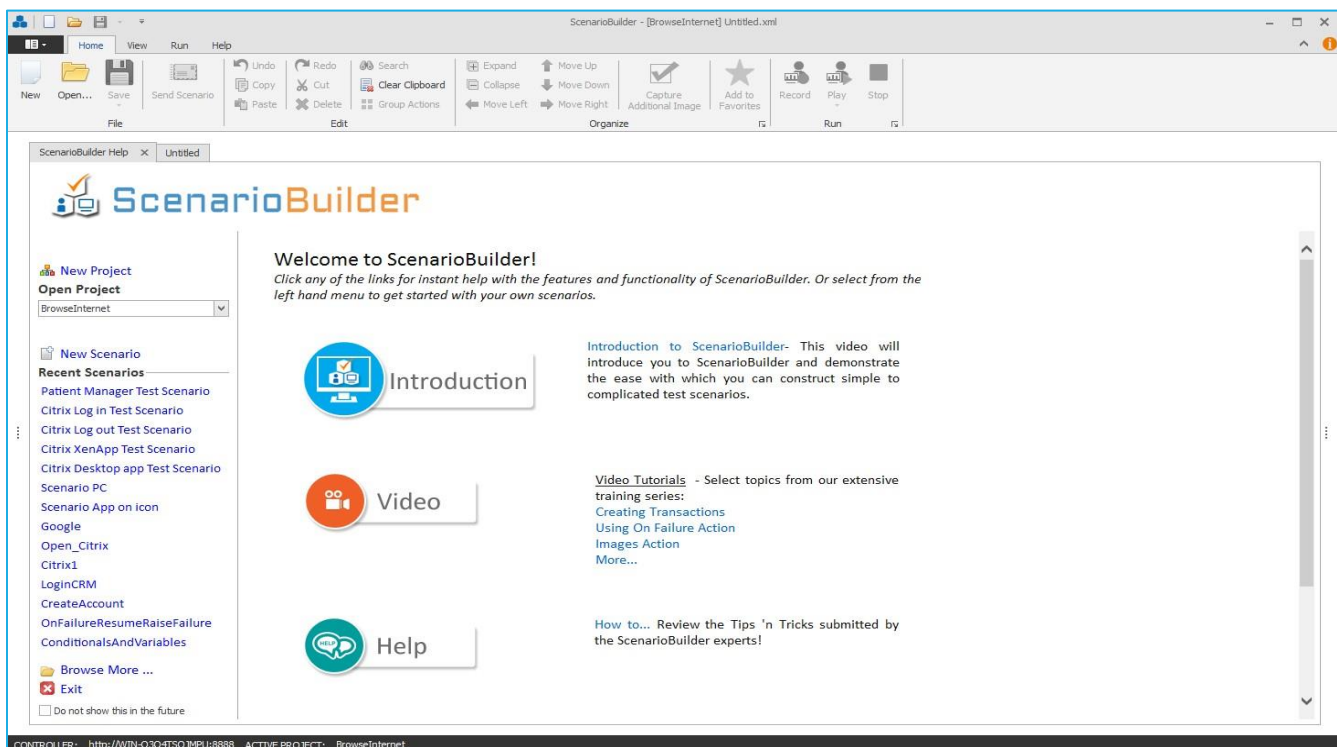
When ScenarioBuilder launches for the first time, it will ask you to create a new project. Upon installation, ScenarioBuilder creates the “Default Project” folder in your user’s Documents/ScenarioBuilder folder, where scenarios will be stored. Additional Project folders can be created from ScenarioBuilder.


The Project will have five folders: Documents, Scenarios, Components, Images, and Variables folders.

- Save your documents in the Documents folder
- Save your Scenarios in the Scenarios folder
- Save your Components in the Components folder
- Save your Images in the Images folder
- Save your CSV files in the Variables folder

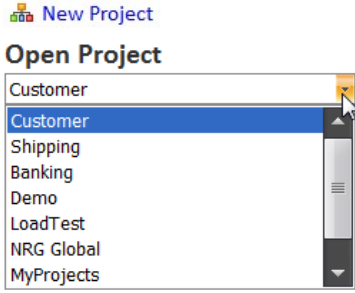
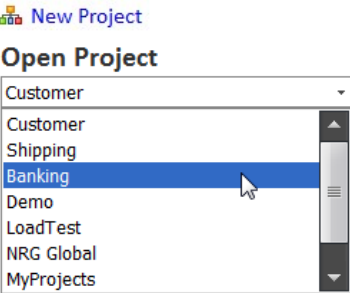
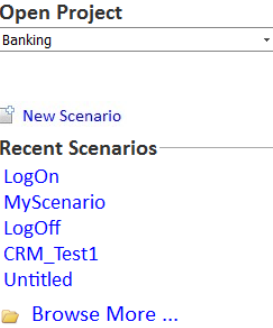
Manage projects from ScenarioBuilder splash screen

Each time ScenarioBuilder is launched, it opens to the Splash Screen (unless you have disabled via the “Do not show this in the future” box at the bottom of the Splash Screen or via the “Splash Screen Settings” in the ScenarioBuilder “Options” menu). The Splash Screen displays the Current Project, and Recent Scenarios from that Project:




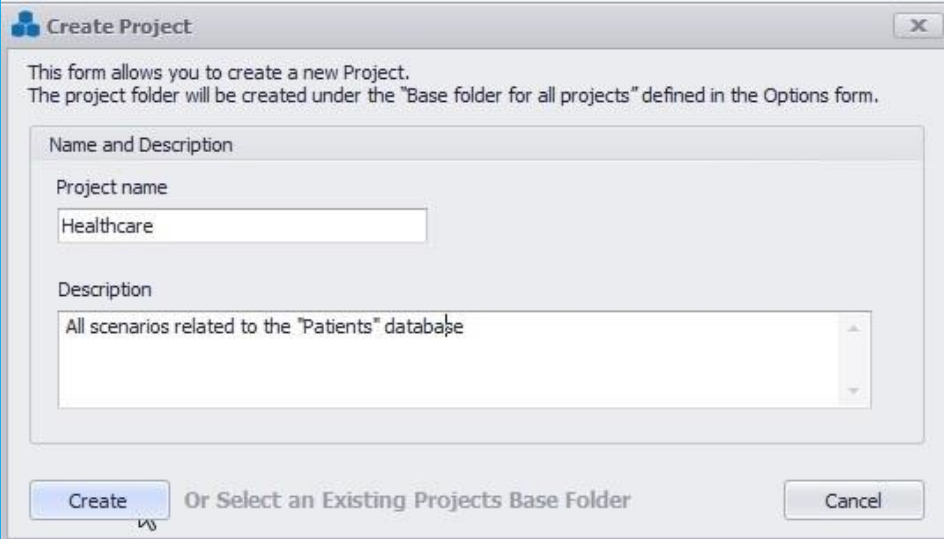
 You can also open your previous project folder by clicking on the “Select an Existing Projects Base Folder” button.

OPEN PROJECTS

<p>Click the “Open Project” drop down box to display all of the Projects from the current “Base Folder for all Projects”;</p>	<p>Select a Project;</p>	<p>The Splash Screen refreshes to display scenarios available from the newly selected Project;</p>
		

CREATE NEW PROJECT

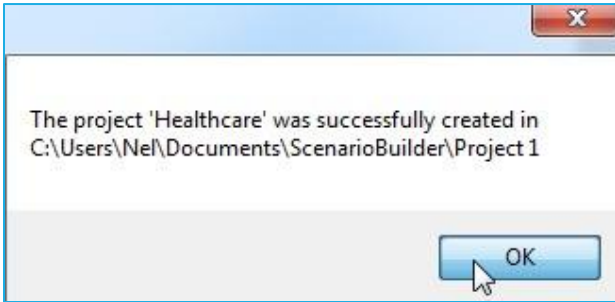
Click  **New Project** . “Create Project” form opens;



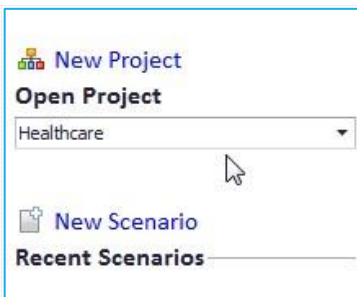

Project name: Enter a meaningful name for your Project.

Description: Enter an optional description to help identify this Project.

Click “Create” to submit the Project. Confirmation pop-up opens;

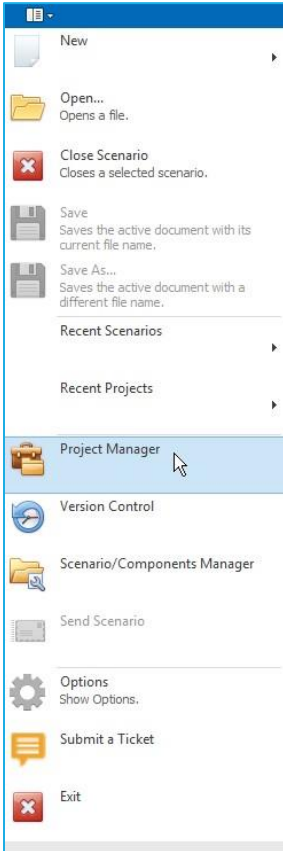


Click “OK” to complete the process. The Splash Screen updates to reflect the new Project;

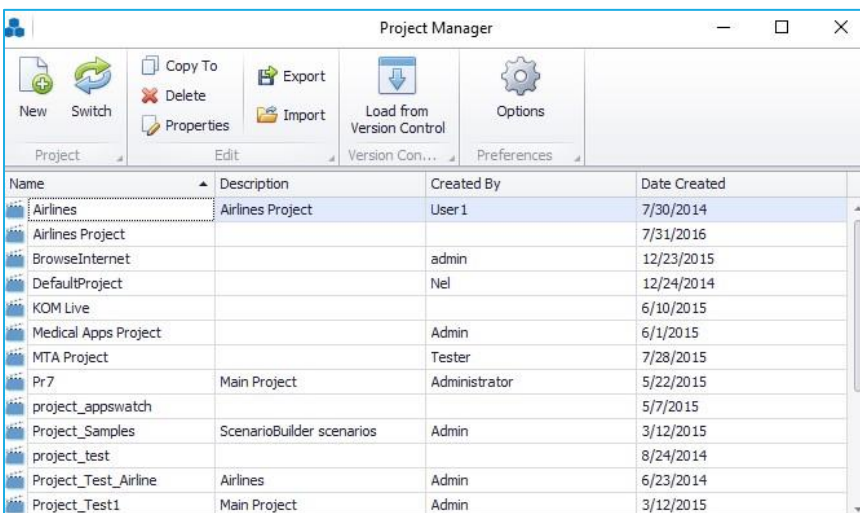


Manage projects from ScenarioBuilder

From the ScenarioBuilder “File” menu, select “Project Manager”;

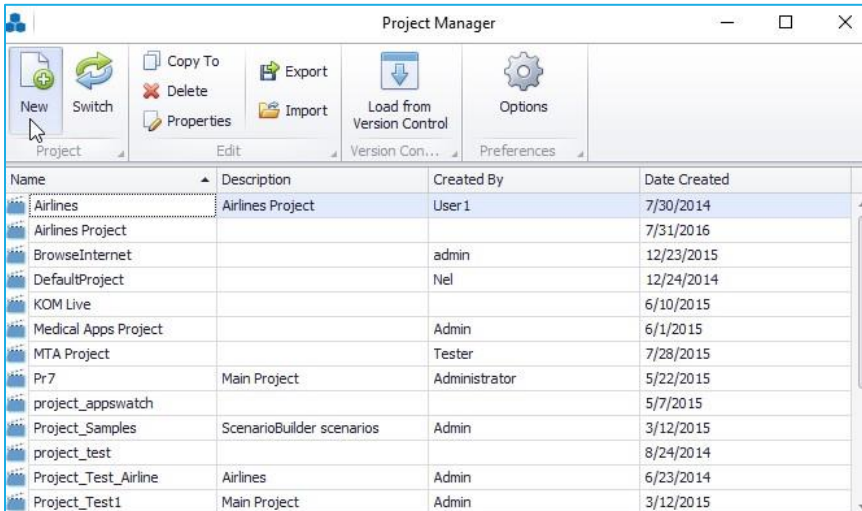


The “Project Manager” pop-up window opens displaying all of the Projects from the current “Base Folder for all Projects”;

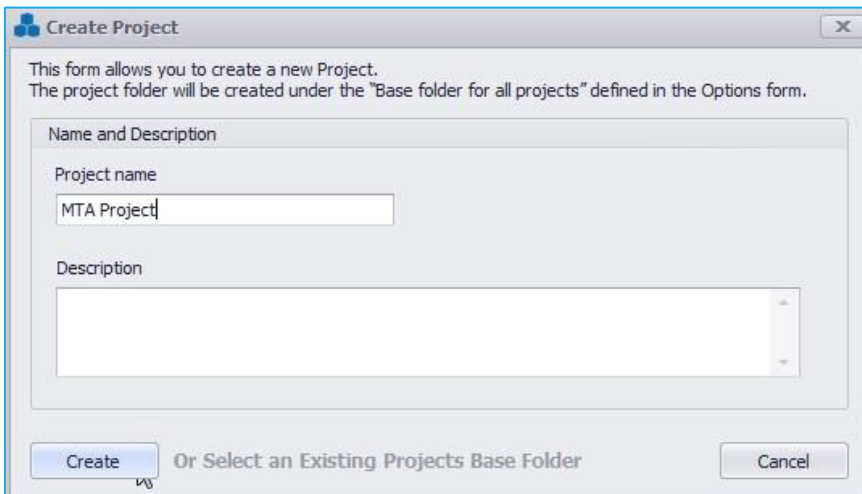


CREATE NEW PROJECT

Click on “New” button;



Fill out the following fields for the new project and click on “OK” button.

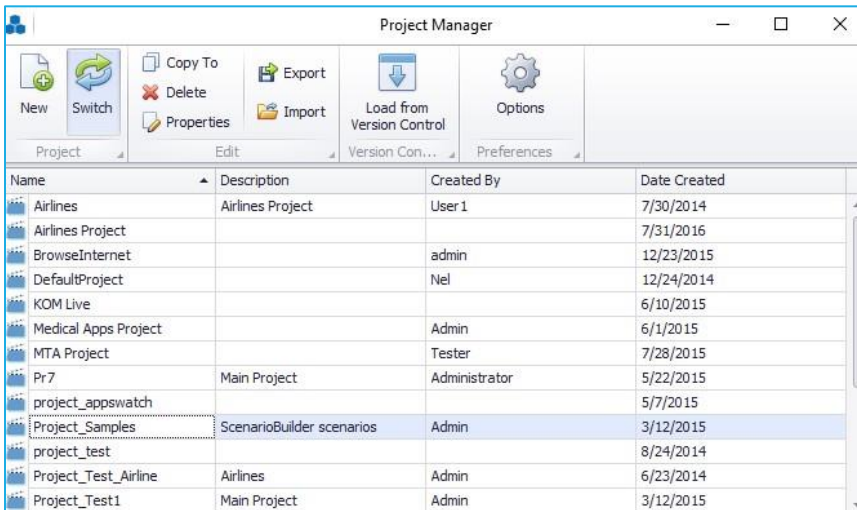


The new project will be added to the projects list.

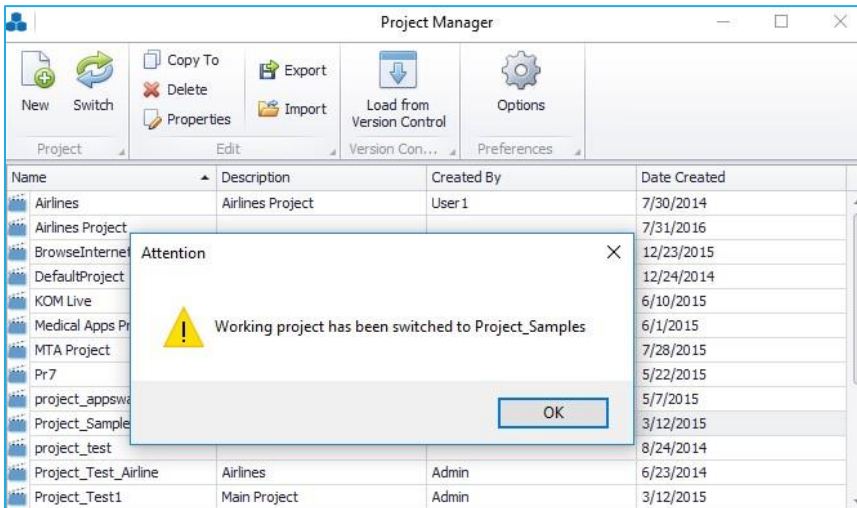


SWITCH PROJECTS

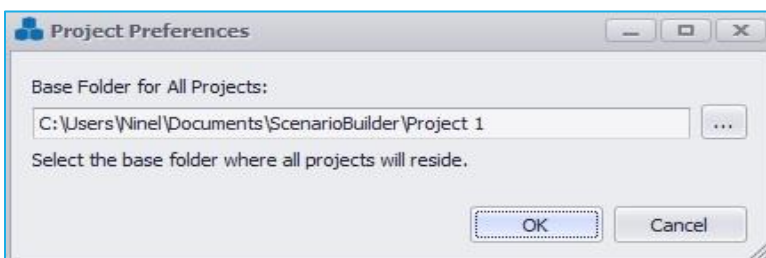
Click the desired project to select it and click on “Switch” button.



Confirmation pop-up opens; Click “OK” to complete the process.



If desired project resides in another “Base Folder”, click on “Options” button to change the location of the “Base Folder”;

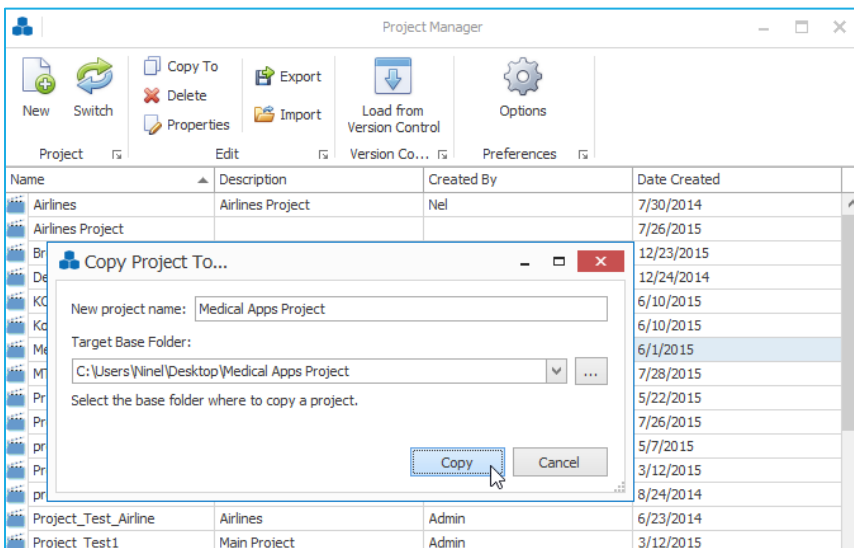
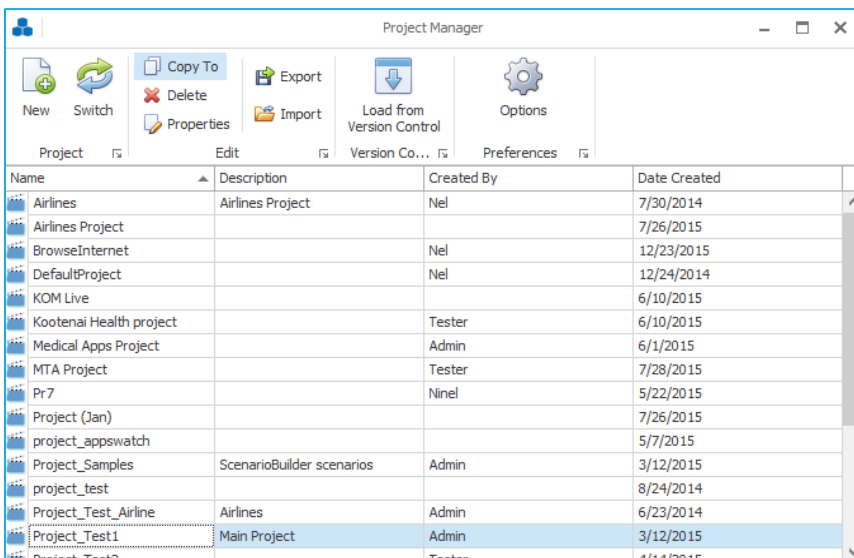


MOVE PROJECT

Creating a zip file from the ScenarioBuilder base folder, copying and pasting it to desired location, is the easiest way to move your ScenarioBuilder project folder from one location to another location.

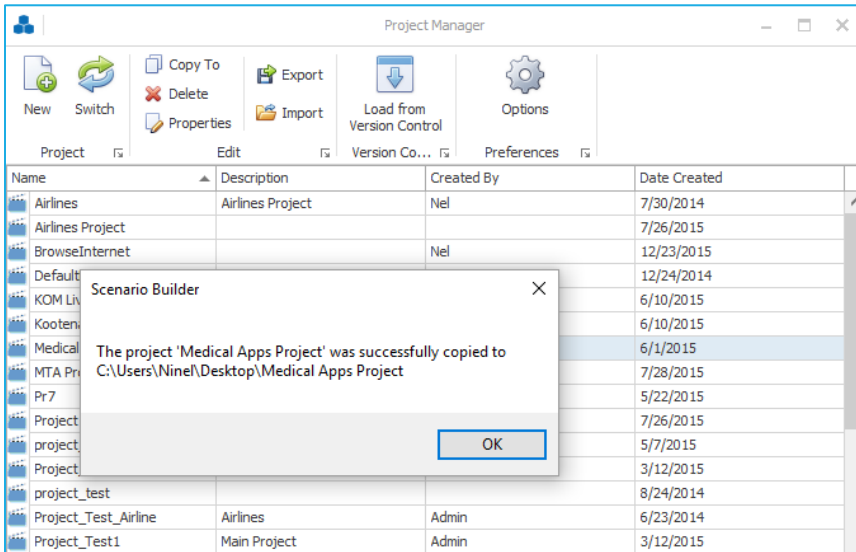
COPY PROJECT TO

Click on desired project to select it and then click on “Copy To” to copy the project to a new Base folder.



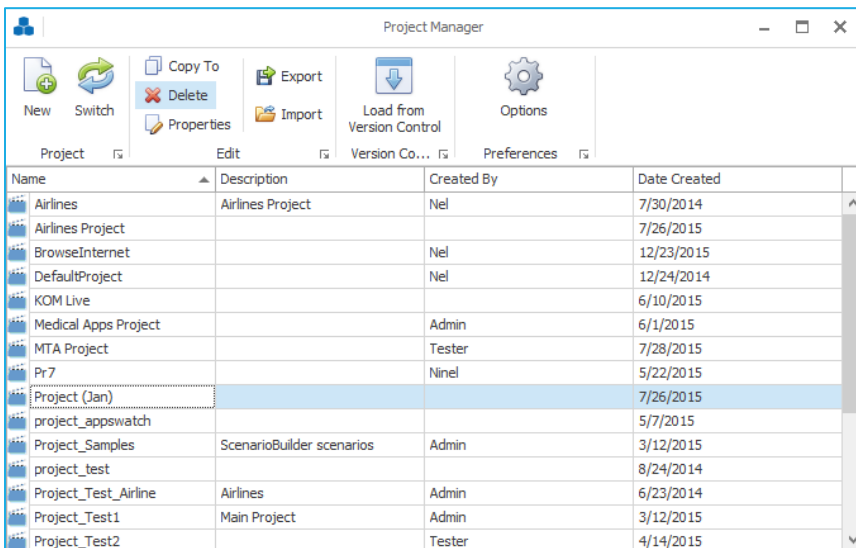
Confirmation pop-up opens; Click “OK” to complete the process.





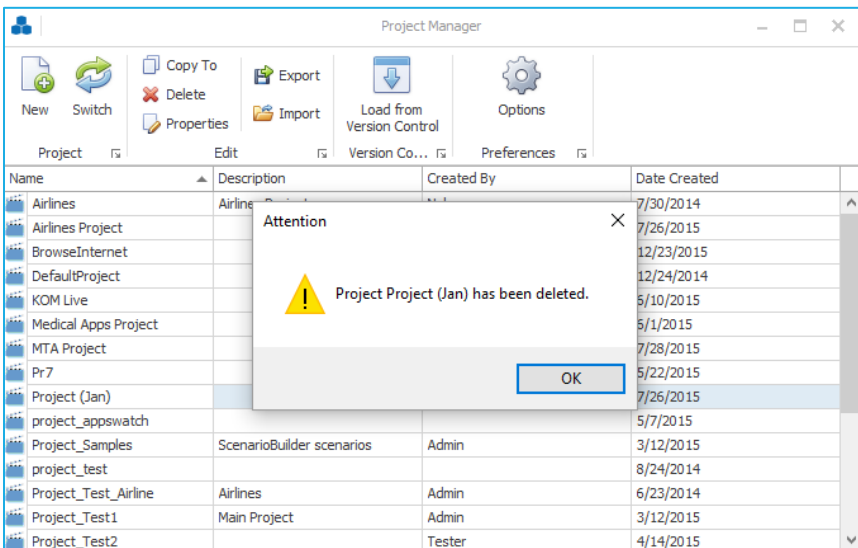
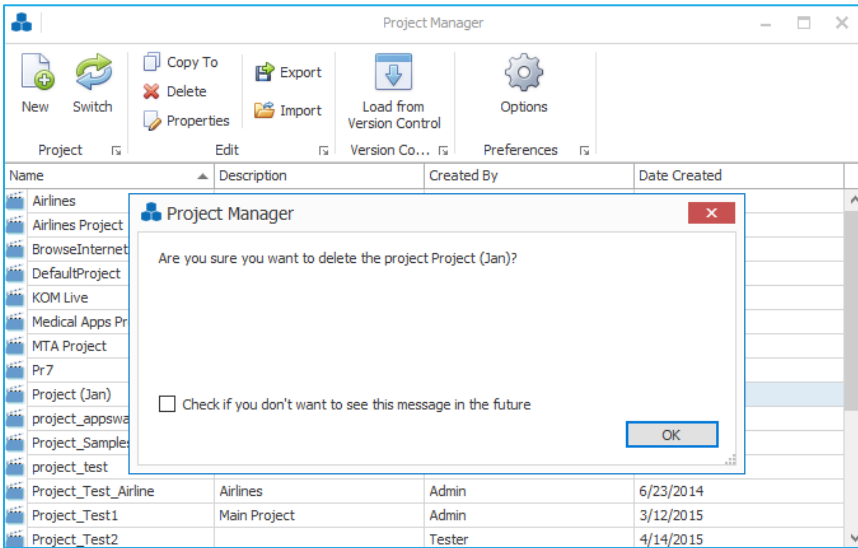
DELETE PROJECT


Click on desired project to select it and click on “Delete” to delete it.



Confirmation pop-up opens; Click “OK” button to finalize the deletion of the project.





 Project deletion results in the deletion of all scenarios, Components, image files, variables files and documents contained in that Project. **Use extreme care when deleting Projects.**



PROJECT PROPERTIES

Click on desired Project to select it and then click on “Properties” to view Project properties.

Name	Description	Created By	Date Created
Airlines	Airlines Project	Nel	7/30/2014
Airlines Project			7/26/2015
BrowseInternet		Nel	12/23/2015
DefaultProject		Nel	12/24/2014
KOM Live			6/10/2015
Medical Apps Project		Admin	6/1/2015
MTA Project		Tester	7/28/2015
Pr7		Ninel	5/22/2015
project_appswatch			5/7/2015
Project_Samples	ScenarioBuilder scenarios	Admin	3/12/2015
project_test			8/24/2014
Project_Test_Airline	Airlines	Admin	6/23/2014
Project_Test1	Main Project	Admin	3/12/2015
Project_Test2		Tester	4/14/2015
Project_Test3		Tester	10/20/2014

Here you can modify the fields.

The 'Project Properties' dialog box is open, showing the following fields:

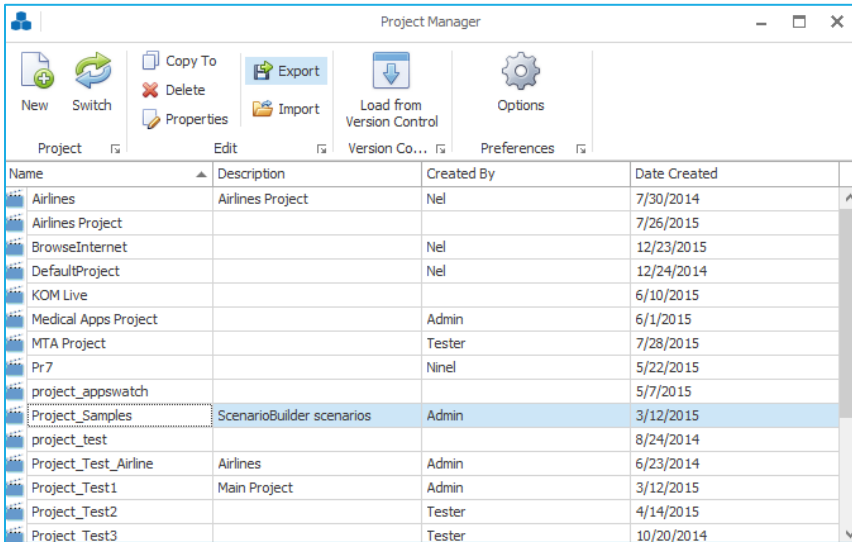
- Project Name: Project_Samples
- Created By: Admin
- Description: ScenarioBuilder scenarios

Buttons: OK, Cancel

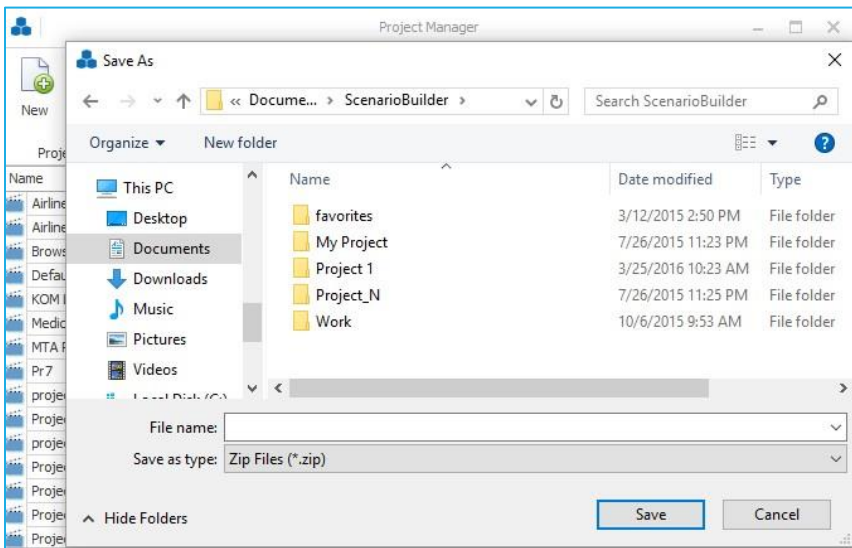


EXPORT PROJECT

Click on desired project to select it and then click on “Export” to export the whole project to a zip file.

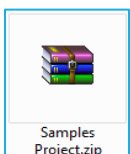


Navigate to an existing directory or create a new folder for your export destination.



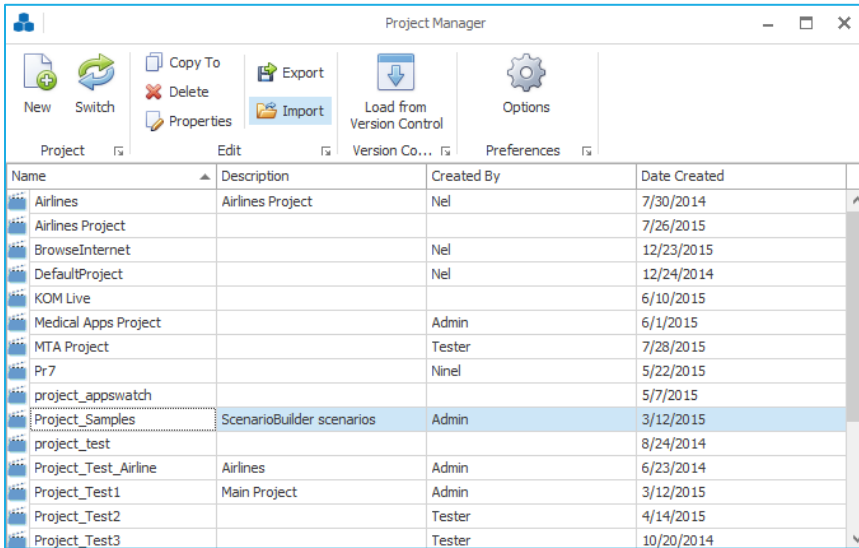
Enter a name for exported file and click on “Save” button.

Your exported project(s) will be saved as a .zip or .rar folder in the folder you selected;

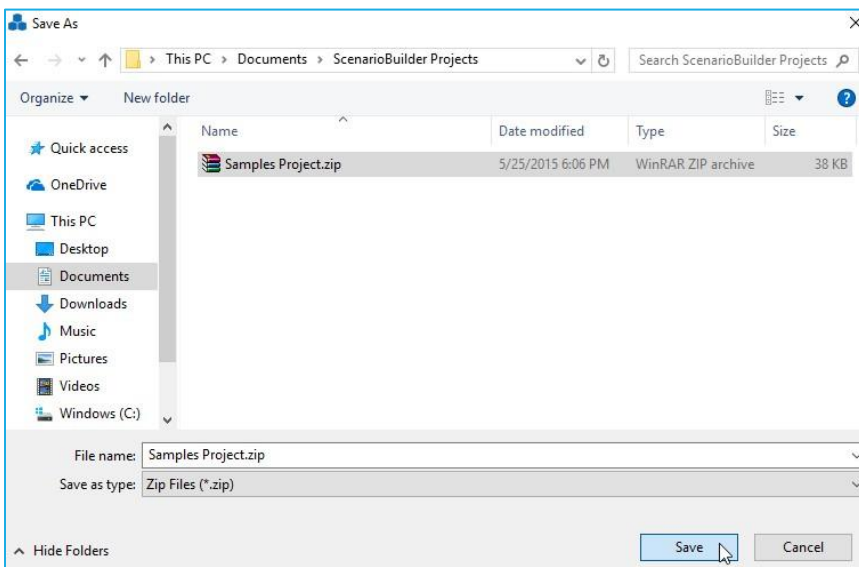


IMPORT PROJECT

Click on “Import” to import a project to the projects list.



Browse to the folder containing the project file;



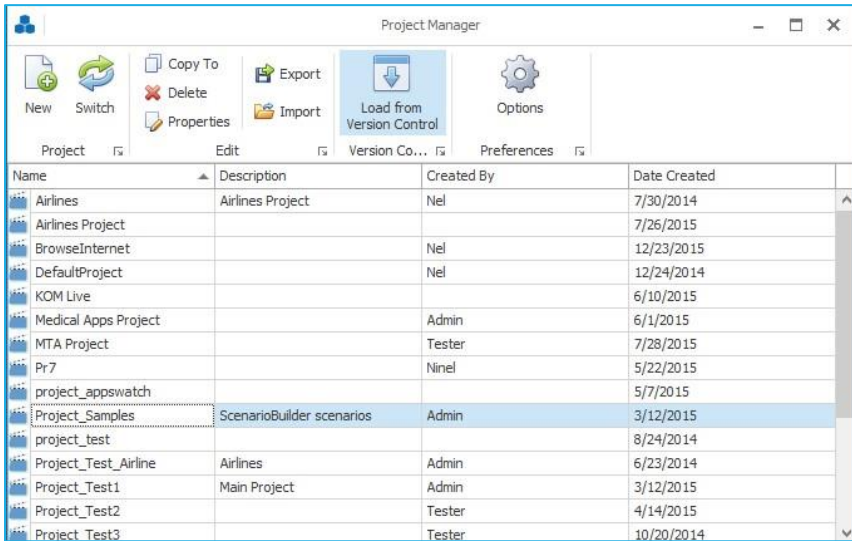
Double-click the desired .zip file. ScenarioBuilder will acknowledge the import;

Imported project will appear in the projects list.

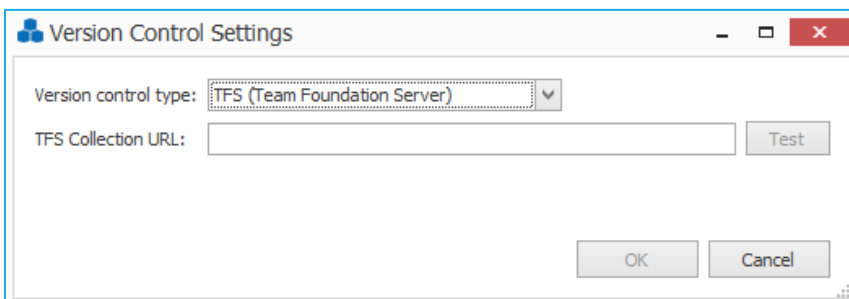


LOAD PROJECT FROM VERSION CONTROL

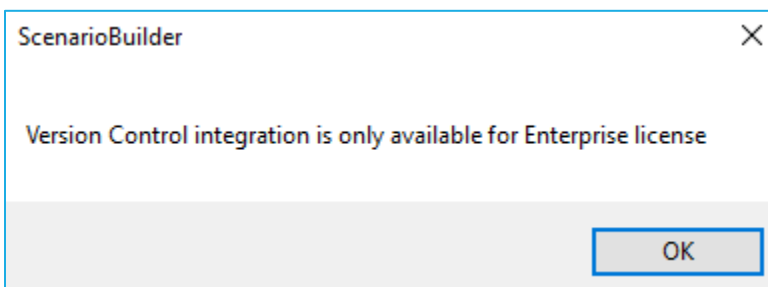
Click on “Load from Version Control” to use “Team Foundation Server” capability.



Select the type of version control product you are connecting to, and specify the appropriate URL.

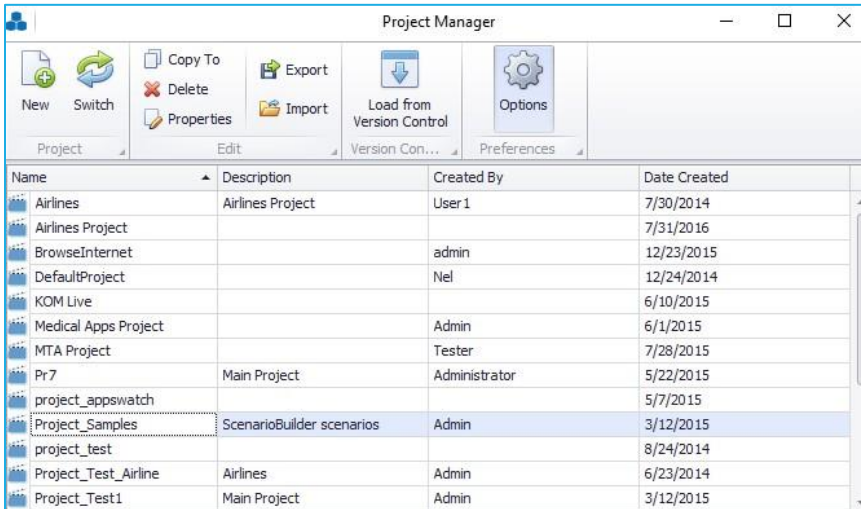


This option works based on the license type and is only available for Enterprise licenses.




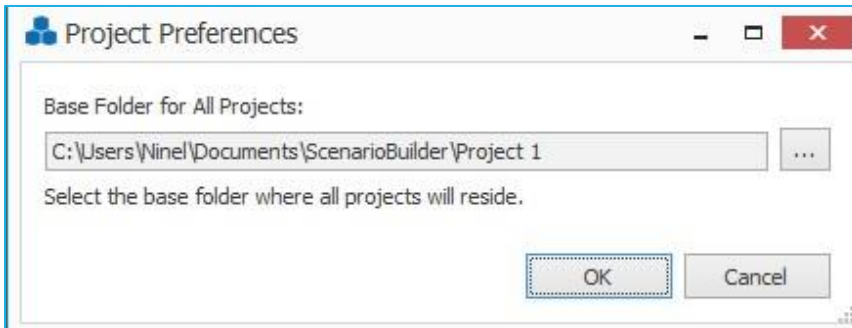
PROJECT OPTIONS

Click on “Options” to view the project preferences.





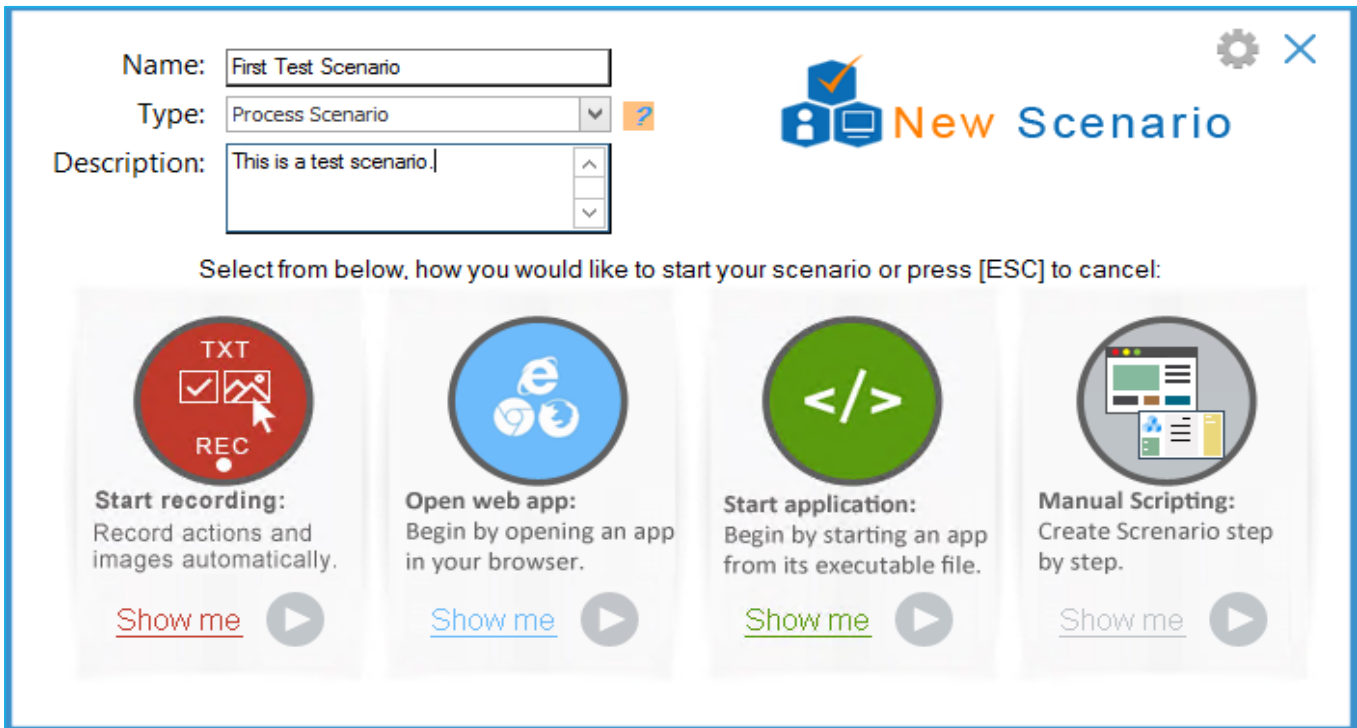
Here you can change the Base folder.

Click the ellipses  button in the “Project Preferences” pane to change the Project Base Folder;



New scenario

When you click the “New Scenario”  [New Scenario](#) link from the Splash Screen, click the “New Scenario”  icon from the ScenarioBuilder tool bar, or select “New” from the ScenarioBuilder “File” menu, the “New Scenario Properties” form displays. The form will get you started with your new Scenario;



Name: Enter a meaningful name for the scenario.

Type: Choose scenario type (allows you to put scenarios into categories).

- **Process Scenario:** Perform a business process. e.g. create invoice.
- **Global Scenario:** Perform actions that are the same across a project.
- **Flow Scenario:** Master script that call other process scenarios to simulate a business flow. e.g. create customer, create invoice, receive payment.

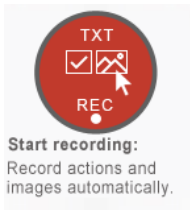
Description: Add an optional description to help identify the scenario.

Select one of the four ways for your scenario to begin.

Click “Show Me” to view a short video explaining how that method works.



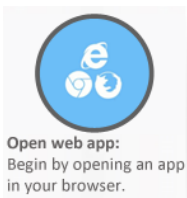
START RECORDING



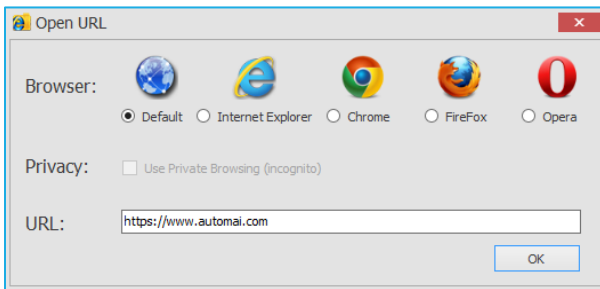
To begin recording mouse movements and clicks. This choice opens ScenarioBuilder in Record mode. Click on any icon on your desktop or toolbar to launch the application you wish to test.

Scenario Recorder will launch.

OPEN WEB APP



To launch a web application with your default browser or a specified browser. This choice opens ScenarioBuilder in Record mode. Begin by entering information when prompted:



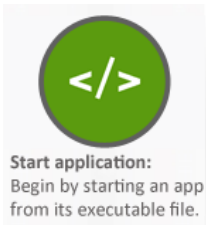
Privacy: If checked, the web browser will be opened in private-browsing mode.

URL: Provide the complete URL address of the web application (e.g. <http://www.automai.com/>). Click "OK". The web application will open.

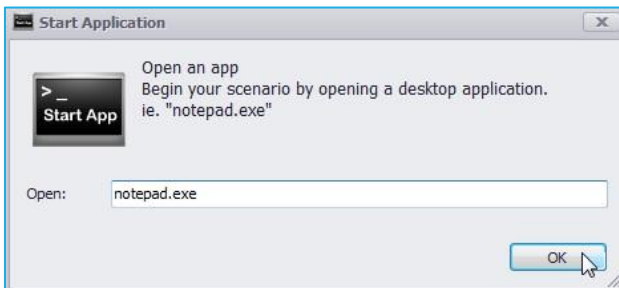
Scenario Recorder will launch.



START APPLICATION



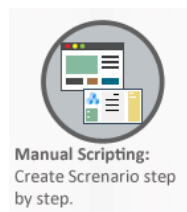
To start your company's application or any installed application such as iexplorer.exe. This choice opens ScenarioBuilder in Record mode. Begin by entering information when prompted:



Open: Enter the name and extension of the executable file. Provide the full path unless the file resides in the Windows path. Click “OK”. The application will open.

Scenario Recorder will launch.

MANUAL SCRIPTING



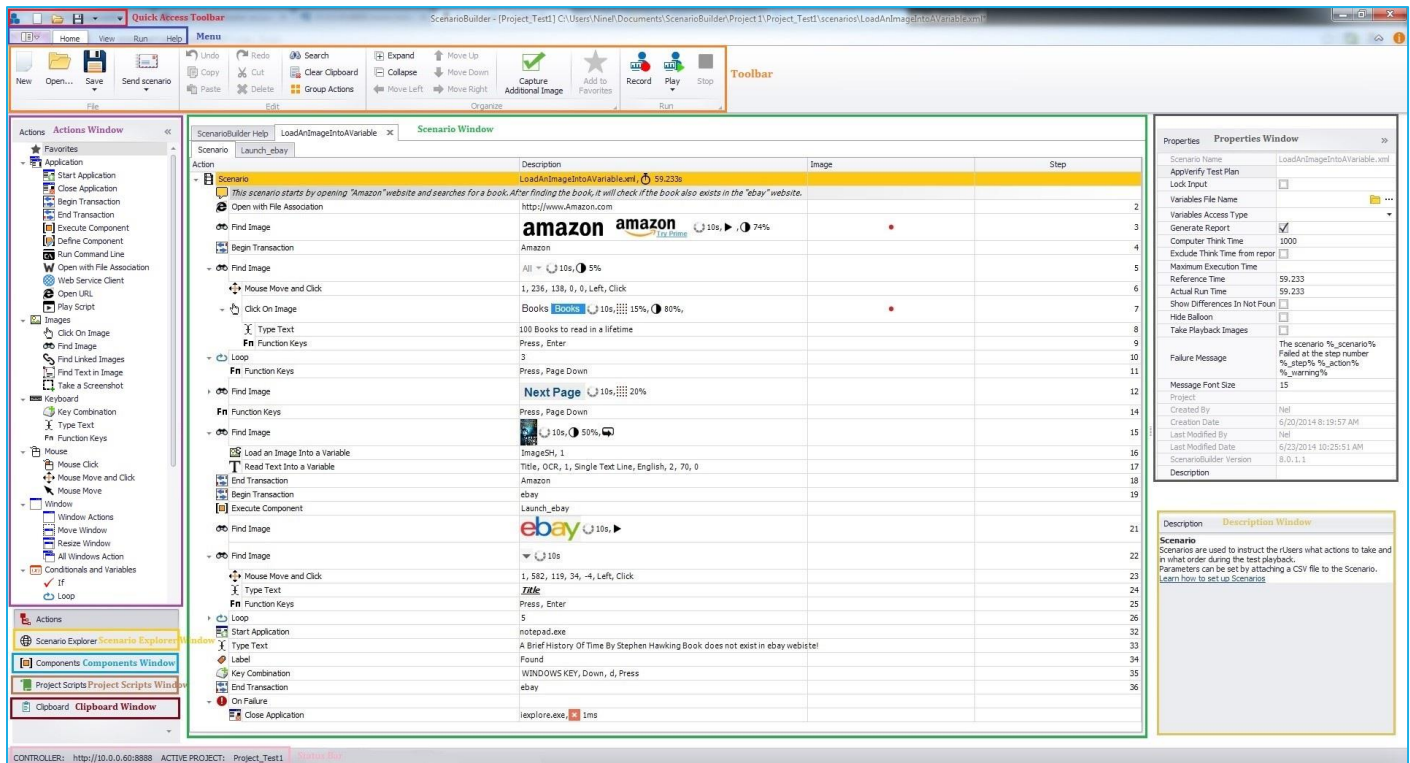
This choice simply starts an empty scenario with no Actions. ScenarioBuilder opens in the edit view. Now you can create actions for your scenario manually.




User interface

ScenarioBuilder's user interface can be adjusted to meet your personal preferences. The following screen shows the default layout. Keep this default layout, or move/resize the panes however you like. If you prefer that the Properties pane is right next to the Actions pane (to reduce cursor movement when updating properties), simply drag and drop the Properties pane across the screen.

DEFAULT LAYOUT



Quick Access Toolbar: Quick Access Toolbar includes commonly used commands and is always accessible. It contains *New*, *Open* and *Save* buttons.

Menu: ScenarioBuilder's menu options are grouped under the  *Home*, *View*, *Run*, *Help* menus. Switching between the *Home*, *View*, *Run* and *Help* tabs will change the Toolbar.

Toolbar: ScenarioBuilder provides an assortment of commonly used menu options in the Toolbar for quick access.

Actions Window: Actions are the building blocks of scenarios. An Action is a step in a scenario that represents a single event (such as a mouse click or an application launch). ScenarioBuilder includes a comprehensive array of





end-user Actions to facilitate the creation of virtually any scenario. This pane contains all of these Actions. Use the '▾' symbol to expand or collapse any of the Action groups.

Components Window: User can access to all components of the current project from this window.

Project Scripts Window: User can access to all scripts of the current project from this window.

Clipboard Window: Copy an Action or a series of Actions to the Clipboard to be recalled later in the current scenario. Actions copied to the Clipboard retain their Property settings. Use the Clipboard when Actions with the same properties are repeated in different places within the scenario. Highlight the Action(s) and right-click. Choose "Copy" from the menu and the item(s) will be added to the Clipboard. Then double-click on the item in the Clipboard to add it to the scenario. Note that the Clipboard is cleared whenever ScenarioBuilder is closed.

Scenario Explorer Window: Scenario Explorer window includes scenario's script, components and CSV files.

Scenario Window: The Scenario Window is the heart of a scenario. It contains the step by step series of Actions in the order in which they will be executed. Add Actions to the scenario by either dragging them into the Scenario Window or simply double-clicking on them (add Component and Component Templates to the scenario by double-clicking on them). Actions can be relocated within the Scenario Window by clicking and dragging them, or by highlighting them and clicking the "Move Up"  Move Up or "Move Down"  Move Down icons in the tool bar. For each Component or Component Template you add to your scenario, a tab for that Component is automatically added to the Scenario Window. The Scenario Window shows, by default, "Action Name" and "Content", however the view can be customized to include "Step" and "Execution Time".


As you open Components or additional scenarios, tabs are added to the Scenario Window. Click on the appropriate tab to display the desired Component or scenario.

Properties Window: Every Scenario, Component and Action has an associated set of properties. These properties are displayed in the Properties pane when the item is highlighted in the Scenario Window. Property values can be edited in the Properties pane.

Description Window: In the Description pane, ScenarioBuilder offers a brief description and/or tip for the highlighted Action or property field.

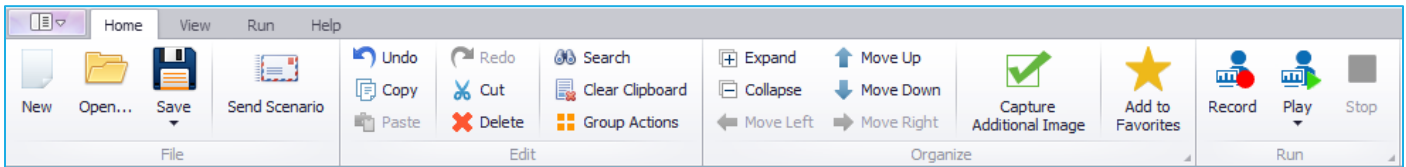
Status Bar: At the bottom of the window, the Status Bar contains useful information about the Controller/Base with its IP address:Port number and also it shows the active project.






 You can launch Controller/Base in the default browser by double clicking on the "CONTROLLER" address on the status bar. Also you can open project folder in the file system by double clicking on the ACTIVE PROJECT name on the status bar.

HOME TOOLBAR












ScenarioBuilder provides an assortment of commonly used menu options in the toolbar for quick access;



Below is a directory and brief description of the toolbar options;

<u>Icon</u>	<u>Name</u>	<u>Description</u>
	New Scenario	Create a new scenario.
	Open Scenario	Browse the current Project to open an existing scenario.
 Save Ctrl+S Save As... Save All Save Workflow As	Save Scenario	Save the scenario and Components.



 Send Scenario	Send Scenario	Send the scenario to either the Controller/Base (command will not execute if “Reference Time” is 0.)
 Undo	Undo Last Action	Reverse the last change made to the Scenario, restoring it to an earlier state. Undo can be used repeatedly to erase a series of changes.
 Redo	Redo Last Action	Reverse the effects of the “Undo” command. Redo can be used for each “Undo” command performed.
 Copy	Copy Selection to Clipboard	Copy the highlighted step(s) to the clipboard.
 Cut	Cut Selection	Cut (copy to clipboard, and remove) the highlighted step(s).
 Paste	Paste Selection from Clipboard	Paste the latest “copied” item from the clipboard to the scenario.
 Delete	Delete Selection	Delete the highlighted step(s).
 Search	Search	Search for the text that user has typed.
 Clear Clipboard	Clear Clipboard	Clear all the actions that were copied
 Group Actions	Group Actions	Group redundant Actions under a loop, for organizational purposes.
 Expand	Expand Actions	Expands the Child Actions of a selected action

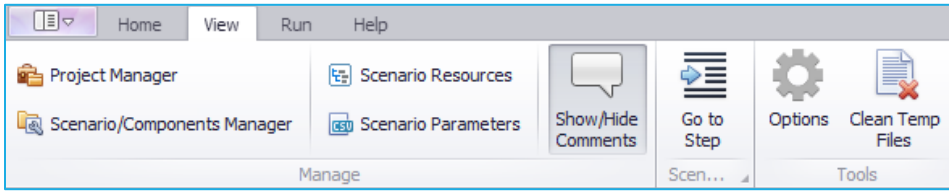


 Collapse	Collapse Actions	Collapses the Child Actions of a selected action
 Move Up	Move Selection Up	Move the highlighted Action up in the scenario.
 Move Down	Move Selection Down	Move the highlighted Action down in the scenario.
 Move Left	Move Selection to the Left	Move a child Action left, toward the parent level.
 Move Right	Move Selection to the Right	Move an Action under the preceding Action, making it a child of the preceding. The child Action will only execute if the parent Action is successful.
	Capture Additional Image	Capture Additional Image for different purposes such as AppVerify validation images, etc.
	Add to Favorites	Add an Action to the Favorites folder to be called later in the current scenario or in a future scenario (from the same Project). Actions saved to Favorites include child Actions, and they retain their Property settings.
	Scenario Recorder	Launch Scenario Recorder.
	Play Scenario from Start	Play the entire scenario.
	Stop Scenario	Stop scenario that was played







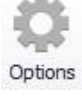


VIEW TOOLBAR

Selecting the *View* tab from the Menu will display this toolbar:



Below is a directory and brief description of the toolbar options;

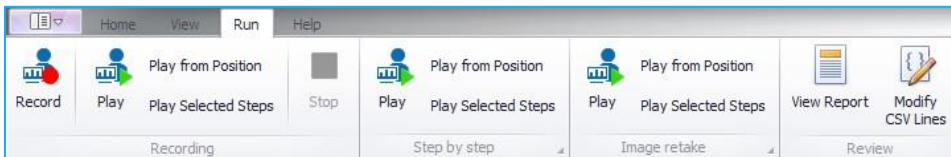
<u>Icon</u>	<u>Name</u>	<u>Description</u>
 Project manager	Project Manager	Opens Project Manager window
 Scenario/Components manager	Scenario/Components Manager	Opens the Scenario/Components Manager
 Scenario Resources	Scenario Resources	Shows the Scenario's Resources such as images, variables, CSV files and the layout.
 Scenario Parameters	Scenario Parameters	Scenario and components parameters
 Show/Hide Comments	Show/Hide Comments	Click to show or hide comments in the Scenario Window
 Go to step	Go toStep	Go to Scenario step
 Options	Options	Opens the 'Options' Menu




 <p>Clean Temp Files</p>	<p>Clean Temp Files</p>	<p>Opens the 'Clean Temp Files' window</p>
---	--------------------------------	--

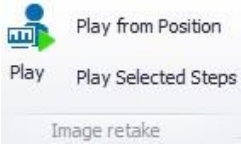


RUN TOOLBAR

Selecting the Run Tab will open a toolbar with different playback options;



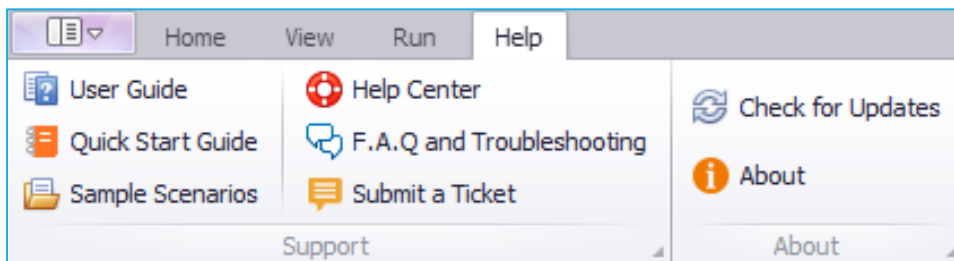
<u>Icon</u>	<u>Name</u>	<u>Description</u>
 <p>Record</p>	<p>Scenario Recorder</p>	<p>Launch Scenario Recorder.</p>
 <p>Play from Position Play Play Selected Steps Recording</p>	<p>Play Scenario from Start</p>	<p>Play the entire scenario.</p>
 <p>Stop</p>	<p>Stop Playback</p>	<p>Stops the current Scenario Playback</p>
 <p>Play from Position Play Play Selected Steps Step by step</p>	<p>Step by Step Playback</p>	<p>Play entire Scenario, selected steps or play from position in Step by Step playback mode. Step by step playback is useful when troubleshooting a Scenario.</p>





 <p>Play from Position Play Play Selected Steps Image retake</p>	<p>Image Retake Playback</p>	<p>Play entire Scenario, selected steps or play from position in Image Retake playback mode. Image Retake playback is useful when updating a Scenario's images.</p>
 <p>View Report</p>	<p>View Runtime Report</p>	<p>Shows the Runtime Report of the last instance the Scenario was run</p>
 <p>Modify CSV Lines</p>	<p>Modify CSV Lines</p>	<p>Allows the CSV line that will be used during playback to be chosen.</p>







HELP TOOLBAR

Selecting the Help Tab will open a tool bar with different help options;



<u>Icon</u>	<u>Name</u>	<u>Description</u>
 User Guide	<p>User Guide</p>	<p>Opens the ScenarioBuilder User Guide</p>
 Quick Start Guide	<p>Quick Start Guide</p>	<p>Opens the ScenarioBuilder Getting Started Guide</p>




 Sample Scenarios	Sample Scenarios	Opens sample Scenarios that can be used as reference when creating your Scenarios.
 Help Center	Help Center	Navigate user to Automai Help Center website
 F.A.Q and Troubleshooting	FAQ and Troubleshooting	Navigate user to ScenarioBuilder FAQ and Troubleshooting on Automai Help Center
 Submit a Ticket	Submit a Ticket	Opens “Submit a support request” form to allow ScenarioBuilder’s users to send requests
 Check for Updates	Check for Updates	Checks for any ScenarioBuilder updates
 About	About	View information about ScenarioBuilder version and license

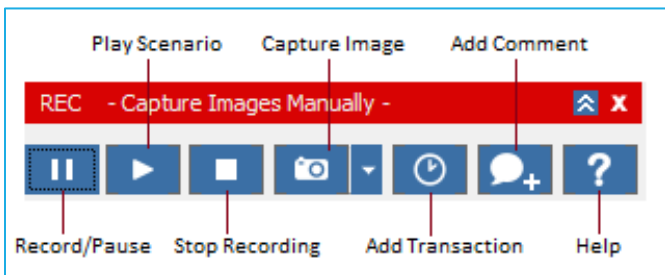



Scenario recorder

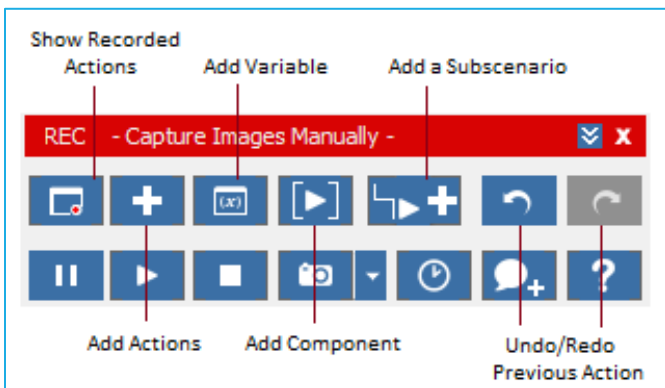
Scenario Recorder makes it easy to create your test scenarios. In Record mode, just use your application the way a user would and the recorder will capture your keystrokes and mouse clicks as they occur. The Scenario Recorder menu provides quick access to many of the commonly used actions in ScenarioBuilder, while remaining in record mode.



To begin, create a new scenario in any mode other than "Manual Scripting", or click the Scenario Recorder

button . ScenarioBuilder will hide and the Scenario Recorder menu will open in the lower right corner of the desktop:






Expand  the Scenario Recorder menu to display additional buttons:





You are now ready to create a test script by clicking on buttons and links and entering text as a user would. The recorder will capture images, keystrokes, and mouse clicks for you. Click "Show Recorded Actions"  to see each action as you are recording it. When you are done recording, simply click Stop  and the Recorder will close, sending you back to ScenarioBuilder with all your scenario steps listed. That's it! Now you can edit your

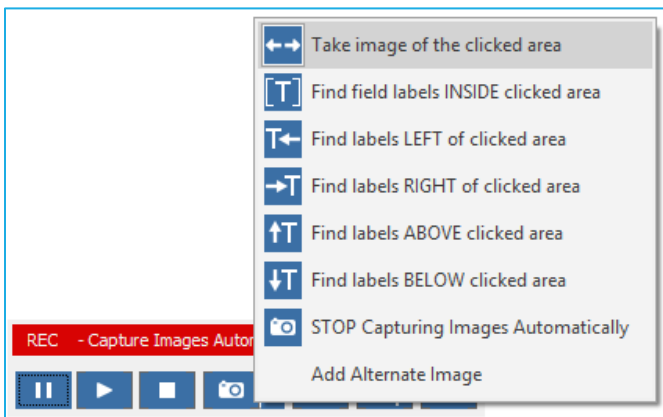


scenario if needed and play it back by clicking Play  on the menu. During recording, you can also click Pause  and then Play  on the Recorder to play back your recorded actions thus far.

At any time during the recording of a scenario, you have a variety of options that can be performed by clicking the Recorder buttons. Some commonly used options are described next.


CAPTURING IMAGES

Images can be captured manually or automatically. If you are in Manual Image Capture mode, you will see the button . Click the down-arrow  next to it to get a list of Image Capture options:



Choosing a new option will change the icon in the Recorder and put you into that Image Capture mode:



The option " Take image of the clicked area" is the most common mode for automatic capture. This tells the Recorder to take an image of the screen wherever you click. This is useful for clicking on icons, buttons, or links in your application.



Data Entry Forms

If you are testing a screen that uses data entry forms, it can be helpful for the Recorder to capture labels to better identify the fields you click on. You can choose the Image Capture option that describes the direction labels are positioned in your application relative to entry boxes. For example, you would choose "Find labels LEFT of clicked area" if the screen you are testing has labels that look something like this:

Patient:	<input type="text"/>
Reason:	<input type="text"/>


Screens with variations

If your clicked area might look different during playback, you can use Manual Image Capture to choose an image that you know is consistent in all variations of playback. These situations may occur when a different user logs in, or when variations in data entry cause changes in color, icons, or wording near the place you want to click. For these situations, change Image Capture mode to "Stop Capturing Images Automatically". Then click on the Capture Image Manually button. Your cursor will change to a screen capture tool. Click the spot where the capture box is positioned over the image you wish to capture, or resize the capture box by pressing the [Ctrl] key twice and then dragging the cursor around the image you wish to take. Choose a name for the image when prompted and then click Accept.

Press [Ctrl] to change mode.
Use mouse or arrow keys with [Shift] to size image.
x:45 y:170 w:41 h:20

Patient	Address	Patient No	Entry Date	Medic
ADAM, J	123456789 LA, 91206	7	17-Oct-2016	FULLER, Andrew (Dr.)
DAVOLI, J	49 642-62-9741 507 - 20th Ave. E.Apt. 2A, Seattle, USA, 98122	1	04-Oct-2007	FULLER, Andrew (Dr.)
DODSON, J	40 482-712-0989 7 Houndstooth Rd., London, UK, WG2 7LT	5	02-Dec-2005	FULLER, Andrew (Dr.)
NEWGLASS, J	64 722 Moss Bay Blvd., Kirkland, USA, 98033	9	01-Feb-2017	FULLER, Andrew (Dr.)
PASSWORD123, Janet (nrgcitrixruser003)	48 812-731-9082 4726 - 11th Ave. N.E., Seattle, USA, 98105	2	17-Aug-2004	FULLER, Andrew (Dr.)
PASSWORD123, Laura (nrgcitrixruser001)	42 72-892-9802 4110 Old Redmond Rd., Redmond, USA, 98052	6	14-Feb-2008	FULLER, Andrew (Dr.)
PEACOCK, Margaret (Mrs.)	58 721-90-6429 123 Oak St., Pasadena	3	30-May-2002	FULLER, Andrew (Dr.)
STANDISH, Sarah	16 Coventry HouseMiner Rd., London, USA, EC2 7JR	8	01-Feb-2017	FULLER, Andrew (Dr.)
SUYAMA, Michael (Mr.)	44 901-27-4332	4	05-Jul-2006	FULLER, Andrew (Dr.)









After this image has been captured, your next click will be anchored to this image. After that, you can switch back to one of the Automatic Image Capture modes, such as " Take image of the clicked area" to have Scenario Recorder capture images automatically for your next actions.





[Click here to watch a video demonstration of Scenario Recorder.](#)

SCENARIO RECORDER MENU







This is a full listing of all options on the Scenario Recorder menu.

<u>Icon</u>	<u>Name</u>	<u>Description</u>
	Pause Recording	Pause the recording session, but keep Scenario Recorder open.
	Start Recording	Record keystrokes and mouse clicks.
	Play Scenario	Play the entire scenario from Scenario Recorder.
	Stop Recording	Stop the recording session. Scenario Recorder closes and ScenarioBuilder displays. Recorded Actions appear in the Scenario Window where they can be edited.
	Capture Image Manually	Click this option in the drop-down menu to go into manual image capture mode. Then click this button on the Recorder to capture an image while remaining in Scenario Recorder mode.
	Take image of the clicked area	Take image where the mouse is clicked (do not look for labels)



	Find field labels INSIDE clicked area	Look for a label where the mouse is clicked (can be used when labels are inside text boxes before data is entered)
	Find labels LEFT of clicked area	Used for labels to the left of entry boxes
	Find labels RIGHT of clicked area	Used for labels to the right of entry boxes
	Find labels ABOVE clicked area	Used for labels above entry boxes
	Find labels BELOW clicked area	Used for labels below entry boxes
	Add Alternate Image	Click this option in the drop-down menu to manually add another version of the current image taken.
	Add Transaction	Insert a Transaction while remaining in the Scenario Recorder mode. Click the icon once to “begin” Transaction, and once to “end” transaction.
	Add Comment	Insert a comment into the scenario. Comments have no impact on the scenario, but can be helpful for informational purposes.
	Help	Access ScenarioBuilder help documentation.
	Show Recorded Actions	Click on this icon to make the script, with the recorded actions appear above the Scenario Recorder Menu

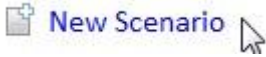


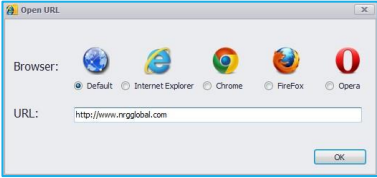
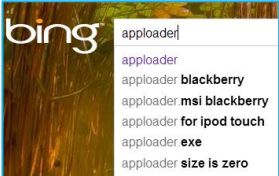



	<p>Scenario Recorder “Actions” Menu</p>	<p>Display the complete list of ScenarioBuilder Actions while remaining in Scenario Recorder mode. Actions can be clicked on to add to scenario.</p>
	<p>Add Variable</p>	<p>Add a variable parameter to the scenario. Variable parameter can be retrieved from an existing CSV file or from the pre-defined set that ScenarioBuilder offers.</p>
	<p>Add Component</p>	<p>Insert a Component while remaining in Scenario Recorder mode. Enter the name of an <u>existing</u> Component to simply insert it. Enter the name of a <u>new</u> component to create it on the fly. If creating a new component, clicking the “Add Component” icon a second time will conclude the Component creation and revert to adding Actions to the scenario.</p>
	<p>Add Sub-scenario</p>	<p>Insert a Sub-scenario while remaining in Scenario Recorder mode. Select the name of an existing Sub-scenario from the list.</p>
	<p>Undo Previous Action</p>	<p>Reverse the last change made to the Scenario, restoring it to an earlier state. Undo can be used repeatedly to erase a series of changes.</p>
	<p>Redo Previous Action</p>	<p>Re-do a change that was previously undone.</p>

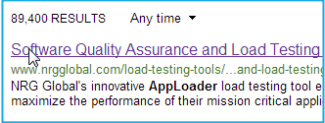

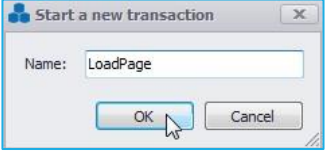


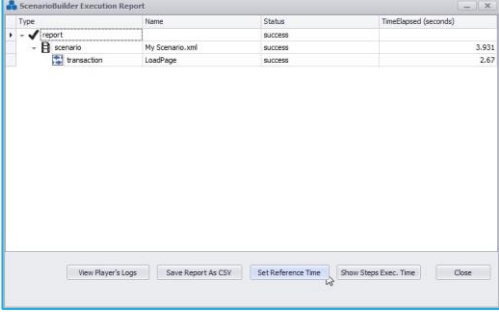



Creating your first scenario with scenario recorder

The following tutorial will help you become familiar with some common Actions used in Scenario Recorder, as well as demonstrate its ease and functionality. This step-by-step walk-through starts with the launching of ScenarioBuilder, takes you through building a scenario that opens Bing, conducts a search, selects a result, and measures the time elapsed for the selected web page to load, then concludes with the playback of the scenario. For a video demonstration of the following walk-through, [click here](#).

Description	Screenshot	Procedure
Create a New Scenario		Launch ScenarioBuilder and click the “New Scenario” link on the “Splash Screen”.
		Note: if ScenarioBuilder is already open, click the “New Scenario” icon in the tool bar.
Launch Bing in your default web browser;		Enter a name for your scenario and select “Open Web App” from the “New Scenario” form.
		In the “Open” field, type <i>http://www.bing.com</i> Press <Enter>. Result: <i>Bing</i> home page opens in default web browser.
Type search criteria “AppLoader” into Bing search field; Press <Enter> on keyboard;		Type <i>AppLoader</i> . Press <Enter>.
		Submit search.



<p>Click on a result;</p>		<p>Click <i>Software Quality Assurance and Load Testing</i> result.</p>
<p>Add a Transaction to measure the time it takes for the web page to load;</p>		<p>Click “Add a Transaction” button on Scenario Recorder menu.</p>
		<p>Name the Transaction and click “OK”.</p>
		<p>Click the “Add a Transaction” button on the Scenario Recorder.</p>
<p>Play Scenario.</p>		<p>Click “Play Scenario” icon on Scenario Recorder menu. Result: Scenario plays from beginning. Upon completion, ScenarioBuilder’s “Execution Report” pops-up.</p>
<p>Set Reference Time;</p>		<p>Click “Set Reference Time” button to establish a performance benchmark;</p>
<p>Stop recording;</p>		<p>Click “Stop Recording” icon on Scenario Recorder menu. <u>Result:</u> Scenario Recorder closes and ScenarioBuilder displays.</p>

Your complete scenario appears in the Scenario Window of ScenarioBuilder.

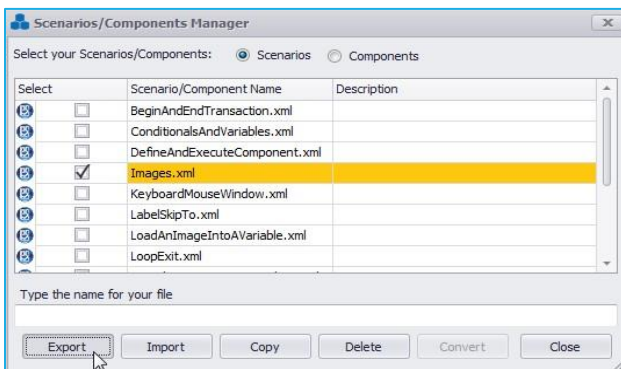


Import and export scenarios

Use “Import” and “Export” to move scenarios in and out of Projects residing in different work spaces. All the resources such as Images, Variables, and Components will accompany imported and exported scenarios.

EXPORT SCENARIO

In the “Scenarios/Components Manager” window click the “Scenarios” radio button. Select the scenario(s) to export and click the “Export” button;

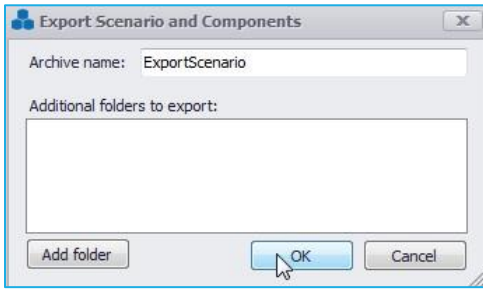


Navigate to an existing directory or create a new folder for your export destination. Note that the export folder is a neutral location from which the Scenario(s) will be imported; it is not intended to be the final location (i.e. the Project folder) for the Scenario(s);

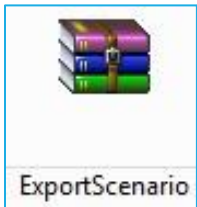



Click “OK”. The “Exporting Grouped Files” window pops up. Enter a name for the exported set of files and click “OK”;






Your exported Scenario(s) will be saved as a .zip folder;

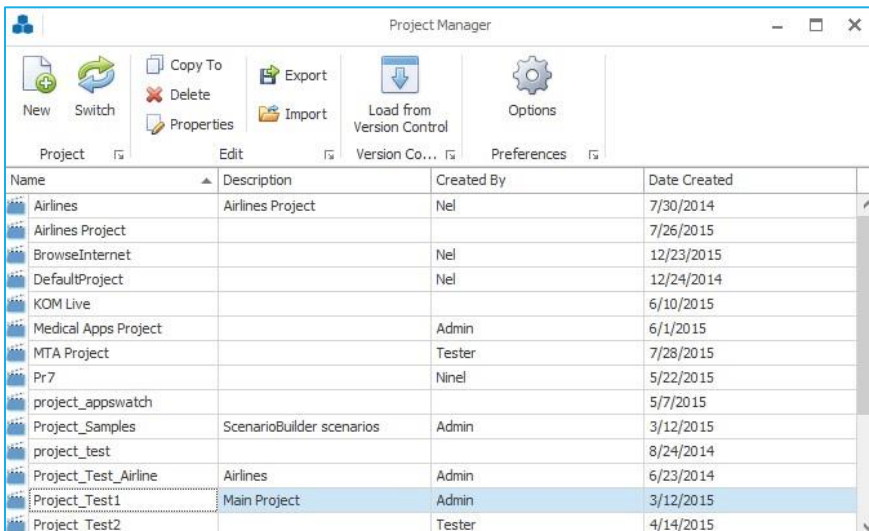


 You can also add additional folders such as “Documents” to your scenario.

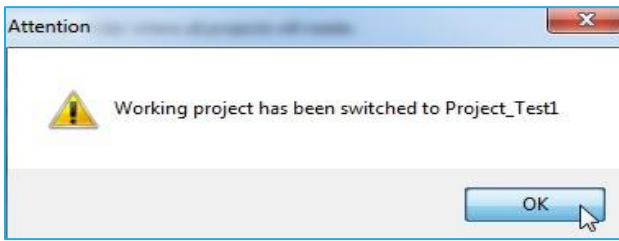
IMPORT SCENARIO

After exporting your Scenario(s), open the existing Project, or create a new Project into which you wish to import the Scenario(s) (if importing to a Project in a different base folder, change the Project base folder from the “Options” menu);

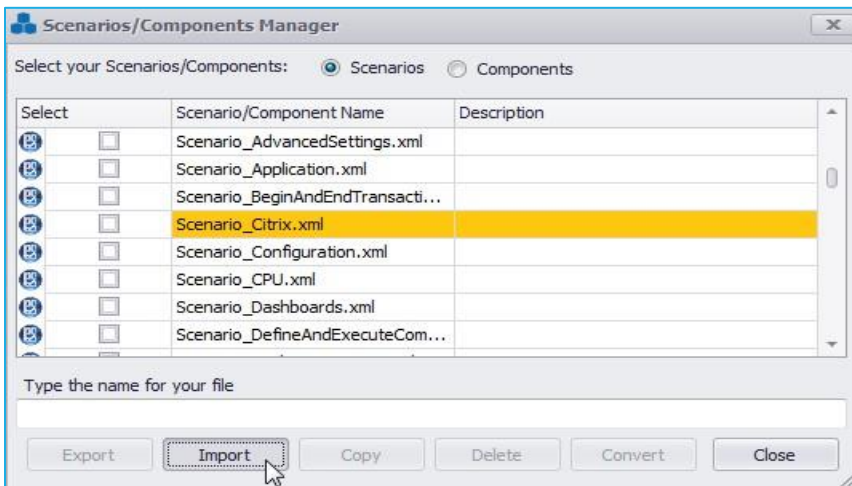
Click the “Project Manager”  option in the “File” menu. The Project Manager opens;



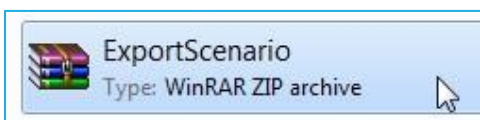
Select the Project to which you want to switch and click “Switch” button. ScenarioBuilder will notify you that the Project folder has been switch;



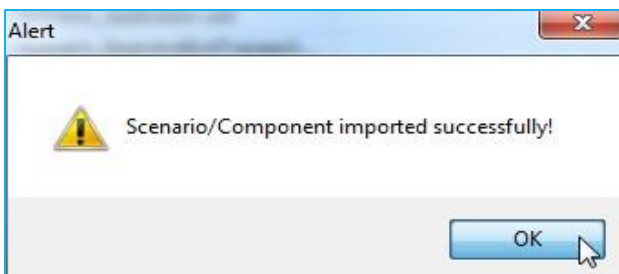
Now that you are working in a new Project folder, you can import the Scenario(s). Open the “Components and Scenarios Manager” and click “Import” button;



Browse to the folder containing the exported Scenario(s);



Double-click the desired .zip file. ScenarioBuilder will acknowledge the import;




Imported Scenario(s) will become available in the current Project.

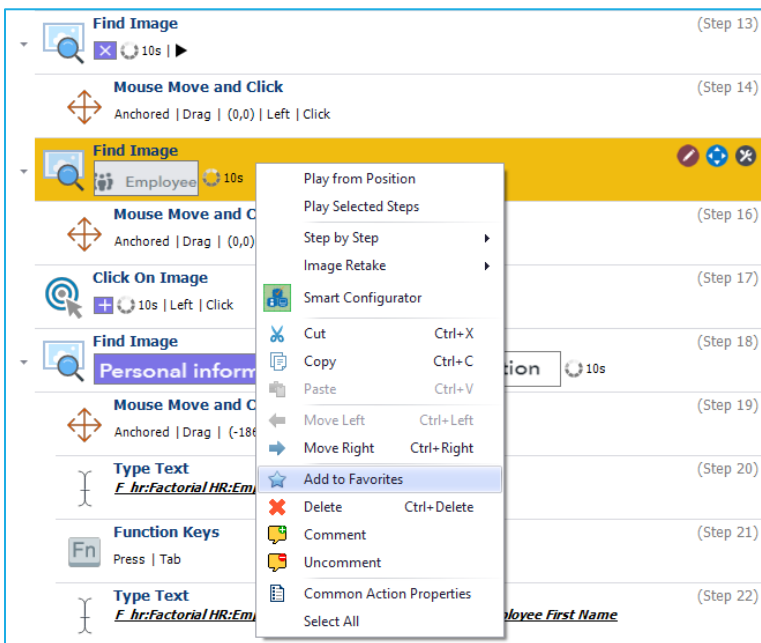


Actions

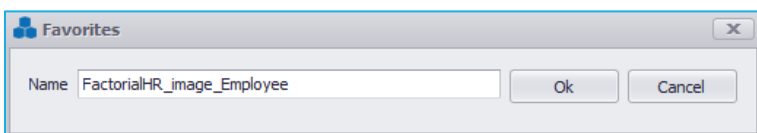
This section defines ScenarioBuilder’s actions and their associated properties.

Favorites

Add an Action to the Favorites folder to be called later, in the current scenario or in a future scenario (from the same Project). Actions saved to Favorites include their child Actions, and they retain their Property settings. Right-click the desired Action (do not highlight the child Actions) and select “Add to favorites” (or highlight the Action and click the “Add to Favorites”  Add to Favorites tool bar icon);

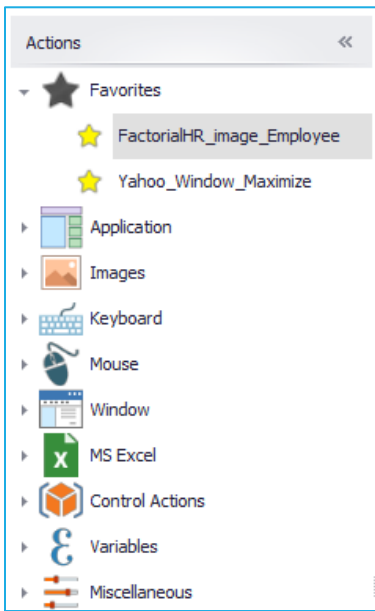


The “Favorites” window opens. Provide a name for the Action and click “Ok”;




Favorites can be found in the “Actions” pane of the “Actions/Components” window;



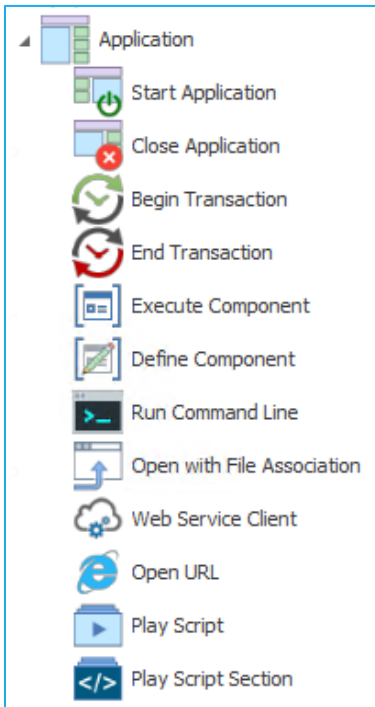


Double-click or drag the “Favorite” item to add it to the Scenario, just as you would with any Action.

Delete an item from the Favorites by right-clicking and choosing “Remove”  Remove from Favorites ;



Application



START APPLICATION

Launch a program via its executable file.

Application Name: To launch an executable that is in the Windows path, simply enter the executable name (e.g. to open the *Notepad* application, simply enter *notepad.exe*). If the executable is *not* in the Windows path, provide the full path of the executable (e.g. *c:\program files\internet explorer\iexplore.exe*). It is acceptable to add parameters to the executable (e.g. *c:\Program Files\Internet Explorer\iexplore.exe http://www.google.com*).

Working Folder (Optional): Provide the application's working folder.

Continue On Failure: If enabled, the execution of the scenario will continue if this step fails.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



CLOSE APPLICATION

Application Name: Enter the name of the open program you wish to close. Do not use the full path to the executable.

Application Started by this Scenario: If enabled, closes only the instances of this application that were launched by the “Start Application” Action from this scenario.

Use Task Kill Command Line: If enabled, ScenarioBuilder uses “TaskKill” command to close application by ending its process (unsaved data is lost).

Close Active Application: Check this parameter if you need to close the currently active application window using Alt-F4 keys. If “Application Name” option is filled, this option will be ignored.

Continue On Failure: If enabled, the execution of the scenario will continue if this step fails.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

BEGIN TRANSACTION

Use Transactions to mark specific steps in a scenario for response time tracking. For example, a Transaction can be used to measure the time it takes for a user to open and log into an application. Add “Begin Transaction” right *before* the step where you would like the measurement to begin.

Transaction Name: Specify a meaningful name for the Transaction. This is how the Transaction will be identified on reports and graphs

Reference Time (seconds): The response time standard to which all subsequent iterations of the Transaction are compared.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



END TRANSACTION

Use Transactions to mark specific steps in a scenario for response time tracking. For example, a Transaction can be used to measure the time it takes for a user to open and log into an application. Add **End Transaction** *after* the last step of the sequence you are measuring.

Transaction Name: Use the drop-down list to retrieve the Transaction that you want to end.

Reference Time (seconds): The response time standard to which all subsequent iterations of the Transaction are compared. Value is set automatically when “Set Reference Time” button is clicked on “Execution Report”.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

For more about Transactions actions, please check out “[Create Transaction](#)” in our [Video Tutorials](#) series.

EXECUTE COMPONENT

Use to call and execute a defined Component. Refer to [Define Component](#) for more information on defining a Component.

Component Name: Use the drop-down menu to specify the block you wish to execute. You should define your Component first.

Working Folder (Optional): Provide the application’s working folder.

Reg Expression: Use a regular expression to read the execution results. For example, for ping www.automai.com, expression #output# will place what the command returns in the variable output, and expression sent = #sent# will place what is between =space and comma in the SENT variable.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: success or fail. For example, name your variable ExecMyCmdStatus.

Continue On Failure: If enabled, the execution of the scenario will continue if this step fails.

Skip If: Skip executing this action when the condition is satisfied.



Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

DEFINE COMPONENT

A Component is a container within a scenario consisting of a single Action or a series of Actions. Components can be called more than once in a scenario or at any point in the scenario by using the Execute Component Action. Components break up your scenario into meaningful segments for better script organization and maintenance.

Name: Specify a meaningful name for the Component.

Variables File Name (Optional): Browse for the CSV file used by this Component. This should be used to attach a user-defined Scenario Parameters file on the Component level (values are only retrievable from within the Component).

Variables Access Type (Optional): If Scenario Parameters are used, select if the variables should be accessed sequentially or randomly from file.

Reference Time (seconds): The response time standard to which all subsequent iterations of the Component are compared. Value is set automatically when “Set Reference Time” button is clicked on “Execution Report” (if “Report” property is enabled).

Report: If enabled, execution time of the Component will be included in reports and graphs.

Use Scenario Think Time: If enabled, this Component will use the same “Think Time” as that which has been established in the scenario properties.

Exclude Think Time from Report: If enabled, the Think Time will not be included in Component’s execution time. (Note: Setting is only relevant if “Report” property is enabled).

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



To remove a Component from your Scenario, simply right-click on the component tab and select **Remove component**.



RUN COMMAND LINE

Use to execute a program/command at the Operating System level. For example, ping <http://www.automai.com/>. This Action has the same functionality as Start -> Run from the Windows menu bar.

Program/Command Name: Enter the command such as iexplore.exe or launch any file such as *C:\Program Files (x86)\Notepad++\notepad++.exe*

Working Folder (Optional): Enter the working folder.

Timeout (seconds): Define the time allotted for the command to return results. If the Action does not complete within this specified time, the step finishes with a FAIL status.

Reg Expression: Use a regular expression to read the execution results. For example, for ping www.automai.com, expression #output# will place what the command returns in the variable output, and expression sent = #sent# will place what is between =space and comma in the SENT variable.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible results: *success* or *failure*.

Continue On Failure: If enabled, the execution of the scenario will continue if this step fails.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

OPEN WITH FILE ASSOCIATION

Use to launch a file or an application with the program associated with it. For example, if you want to launch a URL, this Action will open the URL in the default Internet browser.

Open Associated Name: Type the name of the file/application to be opened. For example, for a website address, type the full path: <http://www.automai.com/>. Make sure to include the http:// and www as needed. The scenario will then attempt to open this address in the default browser.

Working Folder (Optional): Enter the working folder.



Reg Expression: Use a regular expression to read the execution results. For example, for ping www.automai.com, expression #output# will place what the command returns in the variable output, and expression sent = #sent# will place what is between =space and comma in the SENT variable.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible results: *success* or *failure*.

Continue On Failure: If enabled, the execution of the scenario will continue if this step fails.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

WEB SERVICE CLIENT

This action can be used to test and integrate REST and SOAP APIs with your scenario. The action can send HTTP 1.1 messages to servers, and receive responses back. It can then check the status code of the response and make sure that the request was successful. Also, this action can verify whether the response payload is equal to a predefined value that you can specify in its properties. Moreover, you can integrate the response value(s) in your scenario by using Xpath querying on an XML response, or JSON path in JavaScript notation to assign these values to a variable.

Name: Give a meaningful name for your Web Service Client. This name will become a variable that will hold the response payload, and it can become an array for the variables that you want to parse from the response (see variable array property).

URL: Specify the URL to be called with this HTTP Client. i.e. <http://www.automai.com/>

Method: Select the HTTP method you want to use with this Web Service Client.

Format: Select the format of the response payload.

Timeout: Duration in seconds to wait for a response from the server.

Basic Authentication Required (Optional): If Checked the client will use basic HTTP authentication with the server. Note that username and password properties need to be filled as well.



Username (Optional): Specify the username to be used for basic http authentication. Make sure that the Basic Authentication property is checked.

Password (Optional): Specify the password to be used for basic http authentication. Make sure that the Basic Authentication property is checked.

HTTP Header(s) (Optional): HTTP Header(s) to be used with the HTTP request.

HTTP Header#:

Http header value should have the format: HTTP_HEADER_KEY: HTTP_HEADER_VALUE

For example: Content-Type: text/xml; charset=utf-8

Request Payload (Optional): Enter the text that will be sent with the request.

Expected Response Payload (Optional): Enter the expected response payload. After getting the response, the Web Service Client Action will check to see if the response payload is equal to this value (trailing newlines are ignored).

Response Variable(s) (Optional): The variables that are expected in the response. If parsed correctly, the result will be an array with the HTTP Client and the index of the variable. For example, if the HTTP Client name is 'weatherClient', and the first variable is parsed, then you can use %weatherClient[1]% to get the variable value.

Response Variable # (index):

Response Variable index in the response variables array and the path to identify its value.

For XML formatted responses, the path is the XPath for a value i.e. /weather/forecast[temp>20]/temp.

For JSON formatted responses, the JavaScript notation is expected as the path i.e.

["weather"]["forecast"][0]["temp"].

Continue On Failure: If enabled, the execution of the scenario will continue if this step fails.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



OPEN URL

Use to launch any valid URL using a specified browser which can be the default one. For example, if you want to launch a URL, this Action will open the URL in the default Internet browser.

Browser: Select the browser you want to open the URL with. It can be “Chrome” , “Firefox” , “IExplore” , “Opera” or the default one.

URL: Specify the URL to be opened. For example, type the full path: <http://www.automai.com/> or www.automai.com. The scenario will then attempt to open this URL in the default or specified browser.

Browser Path (Optional): Location of the browser executable. i.e. *C:\Program Files (x86)\Google\Chrome\Application\chrome.exe*

Private Browser: If checked, the web browser will be opened in private-browsing mode.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

PLAY SCRIPT

This feature allows you to play a scenario from the current project.

Script Type: The type of the scenario.

Script Name: Browse for a saved scenario and select the scenario file you wish to include (file extension for saved scenarios are .xml).

Continue On Failure: Check this field if you would like your main scenario to continue even if this selected (or nested) scenario returns as a FAIL.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: *success* or *failure*.

Skip If: Skip executing this action when the condition is satisfied.



Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

For more about Play Script action, please check out “[Play Script](#)” in our [Video Tutorials](#) series.

PLAY SCRIPT SECTION

This feature allows you to play a section of a Global scenario from the current project.

Script Type: The type of the scenario.

Script Name: Browse for a saved scenario and select the scenario file you wish to include (file extension for saved scenarios are .xml).

Play Section: The section to execute

Continue On Failure: Check this field if you would like your main scenario to continue even if this selected (or nested) scenario returns as a FAIL.

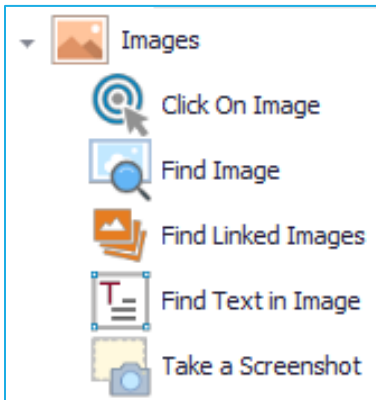
Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: *success* or *failure*.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



Images




CLICK ON IMAGE

With this Action, ScenarioBuilder looks for and then clicks on an image.

Images: If you wish to add more than one image to the Action, simply click on the + icon to the right of Images property field. You then can select your Image #2, #3, and so forth. If you have two or more images, the Action will pass if any one of the specified images is visible on the screen. For example, if you selected bitmap1 and bitmap2, then the scenario will look for both bitmaps and it will proceed to the next Action once it has found the first one.

Find Image #: Right click in this field to determine how you want to select the image:

Capture Image: Select if you need to take a screenshot of an image. Your cursor will change to a screen capture tool. Drag the cursor around the image you wish to take and provide a meaningful name for it.

 You can use arrow keys to move the capture box, and also “shift and arrow keys” combination to change the size of the capture box. This along with a magnified view of the capture image box will allow for easy and accurate capture of small images.

Capture Alternative Image: Select if you want to capture an alternate version of the previous image you captured.

Delayed Capture Image: Select if you want to capture an image on a delay.

Load Image: Select if you have previously captured an image and wish to use it for this Action.



Choose a Variable: If a variable has been created with the “Load an image into a variable” Action, that variable can be selected here.

Contrast Level (can only be used after an image has been “captured” or “loaded”): This will launch the Black and White Contrast Display tool which allows you to test and set a contrast level to save the image in black and white. Use this feature if variations in color are preventing ScenarioBuilder from consistently recognizing and finding an image.

View Image (can only be used after an image has been “captured” or “loaded”): This will open the image in your system’s default viewer.

Edit Image: If you're not satisfied with the way a picture was captured, you can use this option to edit the image to remove useless borders or information for aesthetic reasons or to focus on certain elements in it.

Rename Image: Select if you wish to rename the image.

Remove Image: Select if you wish to remove the image from this Action.

Remove and Delete Image: Select if you wish to remove the image from this Action and delete it from project images.

Timeout (seconds): Define the time allotted for the image to be found. If the image is not found within this specified time, the step finishes with a FAIL status.

On Failure Skip To: Enable this property if you would like your scenario execution to skip to a specific step.

Continue On Failure: Enable this property if you would like your scenario to continue even if the bitmap was not found.

Tolerance: Set the color matching tolerance percentage. The default value (0) matches the exact colors and finds images faster. Note: use this feature especially with Flash and Flex applications as the colors vary with each rendering on a High Color (16 bit) display. Warning: a tolerance higher than 50% may cause false positives.

Black and White Contrast Ratio: A value greater than 0 causes ScenarioBuilder to search for image in black and white. Use the Black and White Contrast Level tool to find an appropriate setting.

Anchor Position X: Specifies where a mouse click will occur horizontally on the image (the “x” coordinate). The default setting is “Center”.



Anchor Position Y: Specifies where a mouse click will occur vertically on the image (the “y” coordinate). The default setting is “Center”.

Mouse Coordinates: Coordinates to where to move the mouse. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %MoveLeft%)

Mouse Button: Select which mouse button will execute the click.

Click Type: Select the desired mouse click function to be executed on the image (*Click; Double-click; Down; Up*).

Timeout after click: If “Repeat” property is greater than 0, the “Timeout after click” value determines how long to wait (in seconds) for the image to disappear, before repeating the mouse click. After the designated number of repeats, if the image has not disappeared, the step will fail.

Repeat: If value is greater than 0, the Action will wait the designated “Timeout after click” length of time for the image to disappear before repeating the mouse click. If value is set to 0, the mouse click will not be repeated and the Action will not depend on the image disappearing.

Sleep Time (milliseconds): This is the amount of time scenario sleeps between image checks. For example, if “Timeout” is set to 10 seconds and “Sleep Time” is set to 500 milliseconds, the scenario will check for image a total of 20 times.

Image Search Area: Click the ‘...’ icon and use the tool to outline an area in which scenario will search for image. Be sure the search area covers the entire image you've taken. Use this feature only when the Image stays on a fixed location on the screen. If specified, the scenario will look for the image in the specified location; otherwise, the scenario will search the entire desktop for the image. In both cases, the scenario searches within milliseconds; however, when the search area is specified, the search is even faster.

Start Search Area X: Image search area coordinates upper left corner X. If needed, you can replace the numerical coordinate value with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftX%)

Start Search Area Y: Image search area coordinates upper left corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftY%)

End Search Area X: Image search area coordinates lower right corner X. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightX%)



End Search Area Y: Image search area coordinates lower right corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightY%)

Click the '✕' to remove the "Image Search Area".

Advanced Image Capture: Some application images are not served in the regular image buffer. This option produces same image quality as Windows Print Screen.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: *success* or *failure*.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

FIND IMAGE


Scenario looks for an image just as an end-user would. This Action can be used to verify that a step has occurred properly (for example, finding a logo, image or banner to confirm that a web page has loaded, before typing text). The "Find Image" Action includes a "Repeat" property that allows you to invoke a keyboard function (like "Page Down") as many times as necessary to search an entire page or form for the image (see "Action Key if no image" property below).

Images: If you wish to add more than one image to the Action, simply click on the + icon to the right of Images property field. You then can select your Image #2, #3, and so forth. If you have two or more images, the Action will pass if any one of the specified images is visible on the screen. For example, if you selected bitmap1 and bitmap2, then the scenario will look for both bitmaps and it will proceed to the next Action once it has found the first one.

Find Image #: Right click in this field to determine how you want to select the image:

Capture Image: Select if you need to take a screenshot of an image. Your cursor will change to a screen capture tool. Drag the cursor around the image you wish to take and provide a meaningful name for it.



 You can use arrow keys to move the capture box, and also “shift and arrow keys” combination to change the size of the capture box. This along with a magnified view of the capture image box will allow for easy and accurate capture of small images.

Capture Alternative Image: Select if you want to capture an alternate version of the previous image you captured.

Delayed Capture Image: Select if you want to capture an image on a delay.

Load Image: Select if you have previously captured an image and wish to use it for this Action.

Choose a Variable: If a variable has been created with the “Load an image into a variable” Action, that variable can be selected here.

Contrast Level (can only be used after an image has been “captured” or “loaded”): This will launch the Black and White Contrast Display tool which allows you to test and set a contrast level to save the image in black and white. Use this feature if variations in color are preventing ScenarioBuilder from consistently recognizing and finding an image.

View Image (can only be used after an image has been “captured” or “loaded”): This will open the image in your system’s default viewer.

Edit Image: If you're not satisfied with the way a picture was captured, you can use this option to edit the image to remove useless borders or information for aesthetic reasons or to focus on certain elements in it.

Rename Image: Select if you wish to rename the image.

Remove Image: Select if you wish to remove the image from this Action.

Remove and Delete Image: Select if you wish to remove the image from this Action and delete it from project images.

Timeout (seconds): Define the time allotted for the image to be found. If the image is not found within this specified time, the step finishes with a FAIL status.

On Failure Skip To: Enable this property if you would like your scenario execution to skip to a specific step.



Continue On Failure: Enable this property if you would like your scenario to continue even if the bitmap was not found.

Tolerance: Set the color matching tolerance percentage. The default value (0) matches the exact colors and finds images faster. Note: use this feature especially with Flash and Flex applications as the colors vary with each rendering on a High Color (16 bit) display. Warning: a tolerance higher than 50% may cause false positives.

Black and White Contrast Ratio: A value greater than 0 causes ScenarioBuilder to search for image in black and white. Use the Black and White Contrast Level tool to find an appropriate setting.

Anchor Position X: Specifies where a mouse click will occur horizontally on the image (the “x” coordinate). The default setting is “Center”.

Anchor Position Y: Specifies where a mouse click will occur vertically on the image (the “y” coordinate). The default setting is “Center”.

Action Key if no image: Select an Action to perform in the event that image is not found. For example, “Page Down” to reveal a lower section of a form or page. Note: this property is ignored if “Repeat” property is set to 0.

Timeout after Key press: This value establishes how long (in seconds) to look for the image after the “Action Key if no image” property is invoked.

Repeat: Establishes how many times to repeat the “Action Key if no image” keystroke. If value is set to 0, the “Action Key if no image” property will be ignored.

Sleep Time (milliseconds): This is the amount of time scenario sleeps between image checks. For example, if “Timeout” is set to 10 seconds and “Sleep Time” is set to 500 milliseconds, the scenario will check for image a total of 20 times.

Hide Progress Feedback window: Select this option if the image might appear in the bottom right area of the screen.

Image Search Area: Click the ‘...’ icon and use the tool to outline an area in which scenario will search for image. Be sure the search area covers the entire image you've taken. Use this feature only when the Image stays on a fixed location on the screen. If specified, the scenario will look for the image in the specified location; otherwise, the scenario will search the entire desktop for the image. In both cases, the scenario searches within milliseconds; however, when the search area is specified, the search is even faster.



Start Search Area X: Image search area coordinates upper left corner X. If needed, you can replace the numerical coordinate value with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftX%)

Start Search Area Y: Image search area coordinates upper left corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftY%)

End Search Area X: Image search area coordinates lower right corner X. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightX%)

End Search Area Y: Image search area coordinates lower right corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightY%)

Click the '✕' to remove the "Image Search Area".

Wait for Image to Disappear: If enabled, Action name (in Scenario Window) changes to "Wait Image Disappear". Step will succeed if image is not present. Usually this Action is used immediately after a "Click on Image" Action that clicks the same image, to ensure that the application has responded to the click (opened the new page; advanced to the link; etc.).

Advanced Image Capture: Some application images are not served in the regular image buffer. This option produces same image quality as Windows Print Screen.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: *success* or *failure*.

Stored Image Coordinates: Put Variable names to store Image Coordinates when the image is found.

Variable to store Image X: This is the name of the variable that will store the X coordinate of the image.

Variable to store Image Y: This is the name of the variable that will store the Y coordinate of the image.

Skip If: Skip executing this action when the condition is satisfied.



Reference Time: Find Bitmap reference time is used to calculate the delay of each execution. This is the required time which within the bitmap search should finish

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

FIND LINKED IMAGES

Find images in a grid to identify and select a specific row or record.

Images: Click on the camera icon at right to capture an image using ScenarioBuilder's tool. Your cursor will change to a screen capture tool. Drag the cursor around the image you wish to take and provide a meaningful name for it. Repeat the process, taking enough images from a row to uniquely identify it.

Timeout (seconds): Define the time allotted for the images to be found. If the images are not found within this specified time, the step finishes with a FAIL status.

Continue On Failure: Enable this property if you would like your scenario to continue even if the bitmap was not found.

Tolerance: Set the color matching tolerance percentage. The default value (0) matches the exact colors and finds images faster. Note: use this feature especially with Flash and Flex applications as the colors vary with each rendering on a High Color (16 bit) display. Warning: a tolerance higher than 50% may cause false positives.

Black and White Contrast Ratio: A value greater than 0 causes ScenarioBuilder to search for image in black and white. Use the Black and White Contrast Level tool to find an appropriate setting.

Anchor Position X: Specifies where a mouse click will occur horizontally on the image (the "x" coordinate). The default setting is "Center".

Anchor Position Y: Specifies where a mouse click will occur vertically on the image (the "y" coordinate). The default setting is "Center".

Sleep Time (milliseconds): This is the amount of time scenario sleeps between image checks. For example, if "Timeout" is set to 10 seconds and "Sleep Time" is set to 500 milliseconds, the scenario will check for image a total of 20 times.



Advanced Image Capture: Some application images are not served in the regular image buffer. This option produces same image quality as Windows Print Screen.

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: *success* or *failure*.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

FIND TEXT IN IMAGE

Scenario searches for specified text within an area of the screen. This action can be used for selecting from a drop down field.

Text to Find: Input the text ScenarioBuilder will search for here.

Regular , Advanced: Regular or Advanced image searching.

Use Anchor: If checked relative coordinates to the parent image or window will be used. This is only valid if the action is an image or window actions' child.

Continue On Failure: If checked the execution of the scenario will continue.

OCR Type: Converts scanned images into ASCII characters and stores it into a variable. Select the type of reading the OCR engine should perform. If ScenarioBuilder is having issues recognizing your language, please contact us and we will provide you with the appropriate setup files. There are 4 types: Auto, Single Block, Single Line and Single Word. The default is value "Auto".

Language: By default ScenarioBuilder uses English Language dictionary to recognize words. This parameter allows to choose a different language if desired. To download data for other languages use:

<https://code.google.com/p/tesseract-ocr/downloads/list>

Scale Factor: The scale factor is used to blow up an image by this factor to properly recognize the image. The Scale Factor is set to 0 by default.



Match Exact Text: If checked the action will succeed if only the text was found in the entire line.

Character Threshold: This is the number of characters that can be off and still make a match. By default, it is set to 0. This means the text to find and the OCR text found must match exactly. If the OCR action is having issues with a character or two you can adjust the number of characters that can be different and still make a match. For example, if you are looking for the text "ID 108764917" and it was having issues correctly recognizing the 9, you could set the Finding Threshold property to 1. This means the text it is looking for and the text it finds can be off by 1 character and still make a successful match. The higher the number, the more characters that can be different and still make a match. Care must be taken when using this property as setting the value too high could result in false positives.

Include Characters: If target contains only alpha characters, all of the same case (upper or lower), this can be indicated to OCR ('A' indicates that all characters are upper case alpha; 'a' indicates that all characters are lower case alpha)

Exclude Characters: Use to eliminate unwanted characters from the text value (such as dashes in a phone number: 213-234-7550 is read as 2132347550)

Text Area: This is the area where ScenarioBuilder will look for the text. Click on the ' ' icon and drag the mouse around the Image.

Start Text Area X: Text search area coordinates upper left corner X. If needed, you can replace the numerical coordinate value with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftX%)

Start Text Area Y: Text search area coordinates upper left corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftY%)

End Text Area X: Text search area coordinates lower right corner X. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightX%)

End Text Area Y: Text search area coordinates lower right corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightY%)

Click the ' × ' to remove the "Text Area".



Mouse Coordinates: Relative coordinates to where to move the mouse

Mouse Button: Select the mouse button on which you need to operate a function

Click Type: Select the mouse function from the drop down list

Execution Status Variable (Optional): Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of this execution. This variable will have two possible values: *success* or *failure*.

Timeout (seconds): Define the time allotted for the images to be found. If the images are not found within this specified time, the step finishes with a FAIL status.

Repeat: Establishes how many times to repeat the “Action Key if no image” keystroke. If value is set to 0, the “Action Key if no image” property will be ignored.

Timeout after Key press: This value establishes how long (in seconds) to look for the image after the “Action Key if no image” property is invoked.

Sleep Time (milliseconds): This is the amount of time scenario sleeps between image checks. For example, if “Timeout” is set to 10 seconds and “Sleep Time” is set to 500 milliseconds, the scenario will check for image a total of 20 times.

Action Key if no image: Select an Action to perform in the event that image is not found. For example, “Page Down” to reveal a lower section of a form or page. Note: this property is ignored if “Repeat” property is set to 0.

Debug OCR: If checked, the log folder will contain the input and output of the OCR.

OCR Engine: Use an integer between 0 and 3 to determine the type of OCR Engine. 0 Legacy engine only. 1 Neural nets LSTM engine only. 2 Legacy + LSTM engines. 3 Default, based on what is available.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



TAKE A SCREENSHOT

Using this action will capture a screenshot of the entire desktop and save it to a defined folder (default: “My Documents”). This action can be used when performing regression tests to verify that the entire screen appears as intended,

File Name: Name that will be given to the screenshot

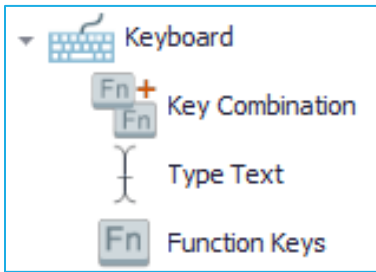
Directory: Type in the full directory where the screenshot will be saved. If left blank, the screenshot will be saved in the “My Documents” folder.

Send to Base: Select to send the screenshot to AppVerify.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



Keyboard



KEY COMBINATION

Use to enter function keys in a congruent order, or when a specified action needs to be taken for each function key in a Key combination.

Keys and Actions:

First Key: The first function key in the combination. Select from the drop down menu. The accompanying action that will be taken with this key is determined by the First Action field.

First Action: This is the action that will be taken using the First Function key.

- 'Down' means that it will hold the key down until instructed to release it.
- 'Press' means that it will only press the key. It will not hold it down.
- 'Up' instructs the specified key to release the key it was holding down.

Second Key: The second function key in the combination. Select from the drop down menu. The accompanying action that will be taken with this key is determined by the Second Action field.

Second Action: This is the action that will be taken using the Second Function key.


Third Key: The third function key in the combination. Select from the drop down menu. The accompanying action that will be taken with this key is determined by the Third Action field.

Third Action: This is the action that will be taken using the Third Function key.

Keyboard Reset: If checked the action will execute Key up functions even if they are not added

Timeout (seconds): Define the time allotted for the image to be found. If the image is not found within this specified time, the step finishes with a FAIL status.



Windows Properties: Specify the window for this Action. To select the window, click the “search”  icon, highlight the desired window with your cursor and hit the CTRL key. After you have selected a window, the **Window Name** and **Window Class Name** fields will automatically be populated. Check these names to make sure you have selected the correct window.

Advanced Find: When checked, the scenario will try to find the best match down to the first 4 letters of the window's name and class.

Skip If: Skip executing this action when the condition is satisfied.


Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

TYPE TEXT

Use Type Text when you need to enter a single character or string of text.

Text: Click the Text field to enter the text you wish to be typed then click “OK”.

Encrypt Text: Check this box if you wish to encrypt the text for security purposes. This is used mainly for passwords.

Windows Properties (Optional): You can specify the window where you wish to type the function key entries. To select the window: click on the  **Search**, highlight the desired window with your cursor and hit the **CTRL** key. After you have selected a window, the **Window Name** and **Window Class Name** fields will automatically be populated. Check these names to make sure you have selected the correct window.

Timeout (seconds): Define the time allotted for the image to be found. If the image is not found within this specified time, the step finishes with a FAIL status.

Advanced Find: When checked, the scenario will try to find the best match down to the first 4 letters of the window's name and class.

Delay Key Strokes (milliseconds): Specify the delay between each keystroke.

Keyboard Reset: If checked the action will execute Key up functions even if they are not added.



Additional Keys And Actions: Select keys and actions to be pressed from the drop down lists. These keys/sctions will happen after Type Text action is executed.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

FUNCTION KEYS


Use Function Keys when you need to press/hold function keys, such as Enter, F6, Tab, etc. Also can be used when a combination of keys must be struck at once.

Action: Select the keyboard action you wish to execute from the drop down box (*Combination, Down, Press, Up*).

Function Keys: Select the function key(s) you wish to type. Use the drop-down box to select each function key. For example, you could set Function Key 1: Ctrl, Function Key 2: Alt, and Function Key 3: Shift.

Timeout (seconds): Define the time allotted for the image to be found. If the image is not found within this specified time, the step finishes with a FAIL status.

Windows Properties (Optional): You can specify the window where you wish to type the function key entries.

To select the window: click on the  **Search**, highlight the desired window with your cursor and hit the **CTRL** key. After you have selected a window, the **Window Name** and **Window Class Name** fields will automatically be populated. Check these names to make sure you have selected the correct window.

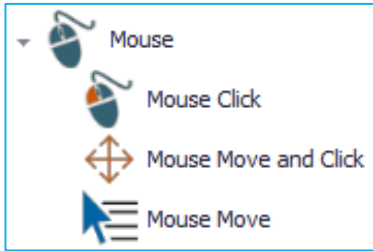
Advanced Find: When checked, the scenario will try to find the best match down to the first 4 letters of the window's name and class.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



Mouse



MOUSE CLICK

Execute a mouse click. Use to select an icon in an application or a product photo on a web page.

Button: Use the pull-down menu to select whether the scenario should click the right, left, or middle button on the mouse.

Click Type: Use the pull-down menu to define the click type (*Click, Double Click, Up, or Down*).

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

MOUSE MOVE AND CLICK

This Action combines the mouse “Move” and “Click” properties into a single Action.

Use Anchor: If enabled, mouse location is relative to the parent *Image* or *Window* Action (property is only relevant if the “Mouse Move” Action is a child of either an Image or Window Action)

Use Drag Motion: If checked the mouse will not jump to the target coordinates. This can be useful in drag and drop actions.

Mouse Coordinates: Use the “Mouse Coordinates” tool to click anywhere on the desktop and automatically set the x and y coordinates. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %MoveLeft%).



Button: Use the pull-down menu to select whether the scenario should click the right, left, or middle button on the mouse.

Click Type: Use the pull-down menu to define the click type (*Click, Double Click, Up, or Down*).

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

MOUSE MOVE

Specify mouse (cursor) location.

Use Anchor: If enabled, mouse location is relative to the parent *Image* or *Window* Action (property is only relevant if the “Mouse Move” Action is a child of either an Image or Window Action)

Use Drag Motion: If checked the mouse will not jump to the target coordinates. This can be useful in drag and drop actions.

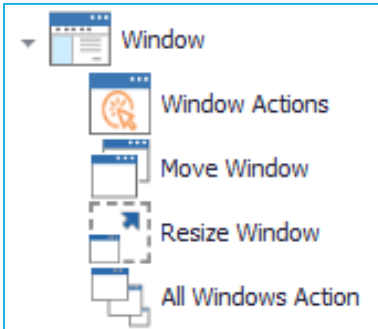
Mouse Coordinates: Use the “Mouse Coordinates” tool to click anywhere on the desktop and automatically set the x and y coordinates. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %MoveLeft%).

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.




Window



WINDOW ACTIONS

Use to manipulate the behavior of windows on the screen.

Window Action Name: Select the desired window Action (*Close, Focus, Maximize, Minimize, Restore, or Waitwindow*).

Windows Properties: Specify the window for this Action. To select the window, click the “search”  icon, highlight the desired window with your cursor and hit the CTRL key. After you have selected a window, the **Window Name** and **Window Class Name** fields will automatically be populated. Check these names to make sure you have selected the correct window.

Match Characters: The scenario player will try to find the best match down to the specified number of characters of the window’s name and/or class.

Advanced Find: When checked, the scenario will try to find the best match down to the first 4 letters of the window’s name and class.

Timeout (seconds): Define the time allotted for the window to be found. If the window is not found within this specified time, the step finishes with a FAIL status.

Sleep Time (milliseconds): This is the amount of time scenario sleeps between window checks. For example, if “Timeout” is set to 10 seconds and “Sleep Time” is set to 500 milliseconds, the scenario will check for the window a total of 20 times.

Wait for Window to Disappear: Enable this property if you want the scenario to wait for the window to disappear from the screen.



Continue On Failure: Enable this property if you would like your scenario to continue even if the window Action is not completed successfully.

Execution Status Variable: Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of its execution. This variable will have two possible results: *success* or *failure*.

Anchor Position X: This is the x coordinate. Specifies where the mouse will move if a move mouse Action follows. The position is Center by default.


Anchor Position Y: This is the y coordinate. Specifies where the mouse will move if a move mouse Action follows. The position is Center by default.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

MOVE WINDOW

Use this Action to move a window to a new location on the screen.

Windows Properties: Specify the window for this Action. To select the window, click the “search”  icon, highlight the desired window with your cursor and hit the CTRL key. After you have selected a window, the **Window Name** and **Window Class Name** fields will automatically be populated. Check these names to make sure you have selected the correct window.

Match Characters: The scenario player will try to find the best match down to the specified number of characters of the window’s name and/or class.

Advanced Find: When checked, the scenario will try to find the best match down to the first 4 letters of the window's name and class.

Timeout (seconds): Define the time allotted for the window to be found. If the window is not found within this specified time, the step finishes with a FAIL status.

Sleep Time (milliseconds): This is the amount of time scenario sleeps between window checks. For example, if “Timeout” is set to 10 seconds and “Sleep Time” is set to 500 milliseconds, the scenario will check for the window a total of 20 times.




Wait for Window to Disappear: Enable this property if you want the scenario to wait for the window to disappear from the screen.

Continue On Failure: Enable this property if you would like your scenario to continue even if the window Action is not completed successfully.

Execution Status Variable: Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of its execution. This variable will have two possible results: *success* or *failure*.

Anchor Position X: This is the x coordinate. Specifies where the mouse will move if a move mouse Action follows. The position is Center by default.

Anchor Position Y: This is the y coordinate. Specifies where the mouse will move if a move mouse Action follows. The position is Center by default.


Coordinates: Enter the desired destination coordinates on the screen. Click on the  tool at right. The mouse cursor will become a '+' sign and you may click a different area while using the captured image as a reference point. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to enclose the variable in percent signs (e.g. %MoveLeft%).

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

RESIZE WINDOW

This Action modifies the window to the desired width and height.

Windows Properties: Specify the window for this Action. To select the window, click the “search”  icon, highlight the desired window with your cursor and hit the CTRL key. After you have selected a window, the **Window Name** and **Window Class Name** fields will automatically be populated. Check these names to make sure you have selected the correct window.

Match Characters: The scenario player will try to find the best match down to the specified number of characters of the window’s name and/or class.



Advanced Find: When checked, the scenario will try to find the best match down to the first 4 letters of the window's name and class.

Timeout (seconds): Define the time allotted for the window to be found. If the window is not found within this specified time, the step finishes with a FAIL status.

Sleep Time (milliseconds): This is the amount of time scenario sleeps between window checks. For example, if "Timeout" is set to 10 seconds and "Sleep Time" is set to 500 milliseconds, the scenario will check for the window a total of 20 times.

Wait for Window to Disappear: Enable this property if you want the scenario to wait for the window to disappear from the screen.

Continue On Failure: Enable this property if you would like your scenario to continue even if the window Action is not completed successfully.

Execution Status Variable: Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of its execution. This variable will have two possible results: *success* or *failure*.

Anchor Position X: This is the x coordinate. Specifies where the mouse will move if a move mouse Action follows. The position is Center by default.

Anchor Position Y: This is the y coordinate. Specifies where the mouse will move if a move mouse Action follows. The position is Center by default.

Size: Enter the desired height and width for the window.

Width: Width of the window

Height: Height of the window

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

ALL WINDOWS ACTION

Use when you want to close or minimize all open windows on the desktop.

Action: Select the window Action to be taken (*close* or *minimize*).



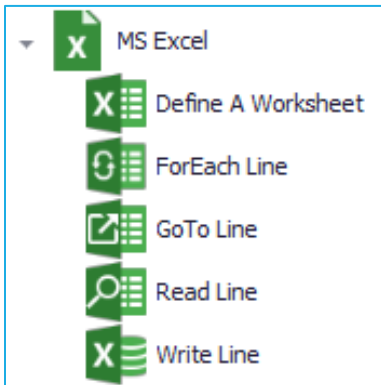
Skip Windows: (Optional) List of windows that you don't want to be closed/minimized with this action

Skip Window #1: Any window that contains this value in its name will be skipped (will not be closed/Minimized)

Scenario Section: This is for Global scenarios only. Specify the section name that can be called from other scenarios.



MS Excel



ScenarioBuilder Microsoft Excel actions let you perform automation steps using Microsoft Excel files. They will allow scripts to read from and write to Excel files.

DEFINE A WORKSHEET

This action should be called for each Excel document/worksheet that will be used in the script.

Dataset Name: Provide a short name that will be used to identify a group of variables in an excel worksheet.

File Name: Click on the folder icon to select your excel file template.

Continue On Failure: If checked, the execution of the scenario will continue even if there is a failure.

Sheet Name: Specify the Excel sheet to be used. If one is not specified, ScenarioBuilder will use the active sheet.

Headers Start Row: Specify the line where your headers start.

Headers End Row: If you have sub-headers, specify the line where your last sub-headers are located

Headers Delimiter: Provide how you want to access your variables from the Dataset by choosing a delimiter value.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



FOREACH LINE

This action repeats the execution of child actions on each line in the Excel handle.

Dataset Name: Name of the already defined Excel Dataset that you want to read from or write to. This action assumes that you already used a Define A Worksheet action before to setup a dataset. Click on the arrow to select a dataset.

On Failure: What to do if any of the loop actions fails. Set to empty to continue to the next line.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

GOTO LINE

This action specifies the line in your dataset to read from or write to.

Dataset Name: Name of the already defined Excel Dataset that you want to read from or write to. This action assumes that you already used a Define A Worksheet action before to setup a dataset. Click on the arrow to select a dataset.

Traverse To Line: Specify where you want to traverse to.

- First: the first line after the headers lines
- Last: the last line in the excel sheet
- Next: the following line in the excel sheet
- Previous: the previous line in the excel sheet

Continue On Failure: If checked, the execution of the scenario will continue even if there is a failure.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



READ LINE

Read the variable data from the Excel worksheet into ScenarioBuilder.

Dataset Name: Name of the already defined Excel Dataset that you want to read from or write to. This action assumes that you already used a Define A Worksheet action before to setup a dataset. Click on the arrow to select a dataset.

Continue On Failure: If checked, the execution of the scenario will continue even if there is a failure.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

WRITE LINE

Writes the value of the dataset to the current line.

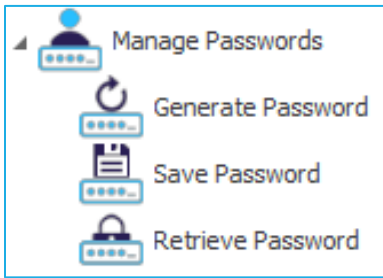
Dataset Name: Name of the already defined Excel Dataset that you want to read from or write to. This action assumes that you already used a Define A Worksheet action before to setup a dataset. Click on the arrow to select a dataset.

Continue On Failure: If checked, the execution of the scenario will continue even if there is a failure.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



Manage Passwords



ScenarioBuilder Manage Passwords actions let you create new passwords, store, and retrieve them when you need to access the application.

Instead of being stored in text files, usernames and passwords can be saved in and retrieved from a vault in MS password manager.

GENERATE PASSWORD

Generate a password and put it in a variable.

Password Length: Specify the length of the generated password.

Include Letters: If Checked the generated password will have letters characters in it.

Include Mixed Case: If Checked the generated password will have upper and lower case letter characters in it.

Include Numbers: If Checked the generated password will have number characters in it.

Include Special Characters: If Checked the generated password will have at least one special character.

Variable Name: The name you want to use to call up this password.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



SAVE PASSWORD

Save a Password in the Password Manager.

Account Name: Create a short name that will be used to save and retrieve associated username/password. Most likely this name will refer to the application being accessed. (for example: Citrix_Access).

User Name Value: Enter a username directly, or enter a variable with % around it (%name%).

Password: To call up a generated password that is stored in MS password manager, enter the variable where you saved it under. (%password%).

To enter a password directly, type into the field directly.

Encrypt Text: Only check this box if you want the password to be encrypted for security reasons. If checked, the password and even the variable name will show as *****.

The original password or variable cannot be viewed again (if unchecked, the password and variable will show as a string of random characters).

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

RETRIEVE PASSWORD

Retrieve a Password from the Password Manager.

Account Name: Use the Account name that you used to save the associated username and password. It most likely has the application name.

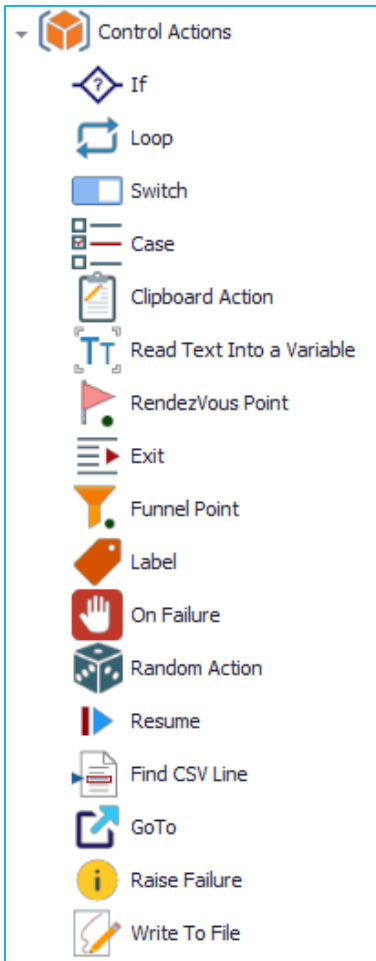
User Name: Provide a variable name where the user name will be saved.

Password: Provide a variable name where the password will be saved.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



Control actions



IF

An “If” Action allows you to set up an “If-Then-Else” condition in your scenario that will execute one of two courses of action, depending on whether the condition is a success (True), a failure (False) or empty.

Expression: The condition to evaluate as being either True or False (e.g. `%Variable%==1; %ReturnedValue%!=Error; %ExecMyImageStatus%==success`).

Use the following comparison operators in your expressions:

Operator	Meaning
==	Equal
!=	Not Equal







>	Greater Than
>=	Greater Than or Equal
<	Less Than
<=	Less Than or Equal
And	And
OR	OR

Below the “If” row in the Scenario Window, add the Action(s) that should occur when the condition is true.

Show Else: Check this box to show the “Else” action. The child Actions under “Else” will occur when the condition is false.

Below the “Else” row in the Scenario Window, add the Actions that should occur when the condition is false. Be sure that the Actions below the “If” and “Else” rows are indented, making them child Actions. Below is an example of an “If” statement:

✓	If	<i>Low Fare</i> == SUCCESS		22
	Find Image	 5s		24
	Mouse Move and Click	Anchored Drag (0,0) Left Click		25
>	Else			26
	Click On Image	 5s		27
	Find Image	 5s		28
	Mouse Move and Click	Anchored Drag (0,0) Left Click		29
	Find Image	 5s		30
	Mouse Move and Click	Anchored Drag (0,0) Left Click		31

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

 For more about working with variables in ScenarioBuilder, please refer to [Scenario Parameters](#) in this guide.



LOOP

Use to repeat an Action or series of Actions a specified number of times.

Iterations: Specify the number of times the Actions in the Loop should be repeated.

Drag Actions into the Scenario Window under the Loop Action. Be sure they are indented, making them child Actions of the Loop.

Variables File Name (Optional): Adds a Variable file to the Loop. Provide full path of the CSV file.

Variable Access Type: Determines how ScenarioBuilder will implement the variables. Sequential or Random.

On Failure: Specify the action if any of the loop actions fails. Set to empty to continue to the next iteration.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

SWITCH

Use Switch when you have different values for a variable and you need to apply different Actions based on the value retrieved. Switch/Case is very similar to the IF/THEN/ELSE. It is normally used when you want to find an equal result, and not a greater or less than result.

Variable Name: After adding **Switch** to your Scenario, enter a Variable Name. Be sure to surround the variable by percentage symbols (e.g. %variable1%).

Next, you will need to add a Case to your Switch sequence. See the description of Case below.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



CASE

Used with the Switch Action.

Once you have added **Switch** to your scenario and defined the variable, then you can drag the **Case** Action over to your Switch sequence.

Variable Value: Define the variable value that will make the case succeed and execute its Action.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

CLIPBOARD ACTION

This action allows you to:

1. Read text from Clipboard into a variable in ScenarioBuilder
2. Write text to Clipboard into a variable in ScenarioBuilder



The clipboard action only lets you to read/write texts from/into the system clipboard using a ScenarioBuilder's variable. This action cannot be used to process files.

Read/Write: Select Read or Write from the drop down menu to read from or write to the system clipboard.

Variable Name: Provide a name for the variable you want to read/write from/to.

Reset Clipboard: Select if you want to clear the clipboard value.

Continue On Failure: If checked, the execution of the scenario will continue even if there is a failure.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



READ TEXT INTO A VARIABLE

This Action copies selected formatted text, or translates selected characters from an image (using optical character recognition (OCR) technology) into text, then reads the value into a variable.

Variable Name: Give a meaningful name to your variable (i.e. a name which represents the characters within the image).

Reg Expression: Use a regular expression to read the execution results. For example, for ping www.automai.com, expression #output# will place what the command returns in the variable output, and expression sent = #sent# will place what is between =space and comma in the SENT variable.

Copy/OCR: Select “Copy” when capturing formatted text; select OCR when capturing an image of text.

Use Anchor: If checked, the relative coordinates to the parent image or window will be used.


Continue On Failure: If checked, the execution of the scenario will continue even if there is a failure.


OCR Type: Converts scanned images into ASCII characters and stores it into a variable. Select the type of reading the OCR engine should perform. If ScenarioBuilder is having issues recognizing your language, please contact us and we will provide you with the appropriate setup files.

Language: Select the language in which the text is written to assist OCR in converting to actual text.

Scale Factor: The scale factor is used to blow up an image by this factor to properly recognize the image. The Scale Factor is set to 0 by default.

Include Characters: If text is all uppercase, enter “A” in this field; if all lowercase, enter “a” in this field. If mixed, leave this field empty.

Exclude Characters: Use when certain characters should be omitted from the captured text. For example, if capturing a phone number that is displayed *(888)624-4447*, excluding parenthesis and dashes will result in the value being read as: *8886244447*. Note: if excluding more than one character, list in this property field *without* any separators, e.g.  .

Text Area: Click the ellipses  to use your mouse to select the area from which to capture text. Capture only the desired text. Capturing extraneous data, even pixels from surrounding area, can cause OCR to fail. Note: the “Start” and “End” Text Area “X” and “Y” property fields will automatically populate after you make your selection.



Start Text Area X: Text search area coordinates upper left corner X. If needed, you can replace the numerical coordinate value with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftX%)

Start Text Area Y: Text search area coordinates upper left corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftY%)

End Text Area X: Text search area coordinates lower right corner X. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightX%)

End Text Area Y: Text search area coordinates lower right corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightY%)

Click the '✕' to remove the "Text Area".


Execution Status Variable: Assign a property variable to this Action if you will later need to take other Actions depending on the success or failure of its execution. This variable will have two possible results: *success* or *failure*.

Debug OCR: If checked, the log folder will contain the input and output of the OCR.

OCR Engine: Use an integer between 0 and 3 to determine the type of OCR Engine. 0 Legacy engine only. 1 Neural nets LSTM engine only. 2 Legacy + LSTM engines. 3 Default, based on what is available.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

 For more about Read Text Into A Variable action, please check out "[Read Text Into Variable](#)" in our [Video Tutorials](#) series.



RENDEZVOUS POINT

Rendezvous points instruct multiple rUsers to wait during the test execution in order to simultaneously perform a task. For example, to emulate peak load on a Citrix server, you can insert a rendezvous point instructing 100 rUsers to click on the update button at the same time.

Put the Rendezvous Point right before the scenario step where you want peak load (all users). Give the Rendezvous Point a meaningful name in the properties pane.

Rendezvous Point: In a Load Test Performance a Rendezvous point instructs vUsers to wait until a defined number of vUsers come to this point to proceed together.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



Rendezvous Point is only applicable to AppLoader; this Action will be ignored in AppsWatch.

EXIT

Exit stops the execution of a Loop, Component, or scenario.

Exit: From the drop-down menu, select which section you want to stop: *Loop, Component, or scenario*.

Exit With Failure: Enable if you want the “Exit” Action to trigger a Failure message.

Exit Message: Compose the Failure message that will be generated upon exit and displayed in reports.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

FUNNEL POINT

In performance and load testing, a Funnel Point instructs Vusers to wait until a defined number of Vusers (Users Limit) have resumed at a set time limit. This is ignored in AppsWatch.



Funnel Name: Give a meaningful name to your Funnel Point

Users Limit: Number of users that can resume execution at a Time Limit Span.

Time Limit: Time (in seconds) that is required to let more users resume execution.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

LABEL

Normally ScenarioBuilder runs the Actions sequentially, but if you wish to skip some Actions based on conditions you can use a Label to do so. A Label is just a name that separates the different Actions. Everything following the Label is considered under that Label.

The scenario is able to jump to labeled Actions by calling their Label name using “Go To” [label name] Action. The only limitation is that you can not skip to a label which had been defined in previous steps. The label must be defined afterwards, since “skip to” just looks forward to find the label.

Label: Give a meaningful name to your label.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

ON FAILURE

On Failure lets you determine what Action(s) to take when a scenario or Component fails. For example, if your Component fails, you might choose to close the application or web page. After adding On Failure to your Scenario, you can drag any Action (i.e. Exit or Close Window) from the Actions panel. These will be the steps taken upon failure.

Raise Failure: Enable the “Raise Failure” property to report the failure in your test results. If this property is not enabled, the scenario will return a “success” status when “On Failure” steps are invoked.




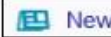





Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



RANDOM ACTION

This will choose one of its child actions to be executed when the scenario runs.

Name Name that will be given to the Random action.

Random Action	Images
<ul style="list-style-type: none"> Find Image <ul style="list-style-type: none"> Mouse Move and Click 	 Mail  10s  60%
<ul style="list-style-type: none"> Find Image <ul style="list-style-type: none"> Mouse Move and Click 	 News  10s  56%
<ul style="list-style-type: none"> Find Image <ul style="list-style-type: none"> Mouse Move and Click 	 Sports  10s  66%

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

RESUME

Resume can be used only under an “On Failure” Action. The “Resume” Action allows the scenario to try and continue running from the point where the failure occurred. Use “Resume” to handle failures caused by pop-ups, for example. Follow an Action that closes the pop-up window with a “Resume” Action to pick up where the scenario left off before the pop-up disrupted the execution.

Retries: This property determines how many times the scenario will attempt to resume after a failure.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

FIND CSV LINE

Find a line in a CSV file based on a condition about a variable in it.

Variables File Name: CSV file name/Variables files name

Expression: The expression to search the CSV file lines for.



Continue On Failure: If checked the expression of the scenario will continue even if there is a failure.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

GO TO

This Action lets the scenario skip steps and jump to a certain point based on the label.

Label: Select the label name that you want the scenario to skip to.

Skip If: Skip executing this action when the condition is satisfied.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

RAISE FAILURE

The Raise Failure Action reports the failure in your test results. If Raise Failure is omitted, the scenario will still return a success status.

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.

WRITE TO FILE

Writes defined text into a log file.

File Name: Name of the log. If you wish to write to a log multiple times within the Script, input the same File name with each "Write to File" action. To add to the log file, select "Write Type: Append". To rewrite a log file select "Write Type: Truncate".

Directory: Select the full directory where the log will be saved. If left blank, the log will be saved in the "My Documents" folder.

Write Type: This determines the type of entry. Append will add text to the log. Truncate will rewrite the log.



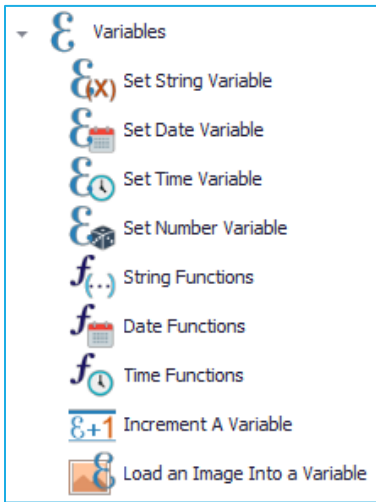
Text: Text that will be typed into the log.

Add New Line: If checked, it will add a new line before writing the text. i.e. 'Enter key'

Scenario Section: This is for Global Scenarios only. Specify the section name that can be called from other scenarios.



Variables



Variables in ScenarioBuilder can store various data types: strings, numbers, dates and times. They can also get their values from external data sources, such as Excel and CSV files. Variables can change their values using the appropriate functions.

With variables, data input can be varied to emulate real world application usage.

SET STRING VARIABLE

Add a string variable to your scenario.

Variable Name - Provide a meaningful name to the variable.

Variable Value - Simply type in your value/text entry or use the pull-down menu to select your pre-defined variable.



The value of a variable is set when the set variable action is executed.

Variable Scope - Specify the level at which you want to access your variable. To learn more about variable scope, please refer to [this article](#).

Encrypted - Checking this box will encrypt the variable value for security purposes. This is useful if your variables contain passwords or other sensitive information.



Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

SET DATE VARIABLE

Add a date variable to your scenario.

Variable Name - Provide a meaningful name to the variable.

Variable Scope - Specify the level at which you want to access your variable. To learn more about variable scope, please refer to [this article](#).

Date Format - Select a Date format.

Below lists the syntaxes associated with Date format:

Syntax	Description
dd	Numeric representation of the day of the month, with leading zeros
ddd	Three-letter abbreviation for the day of the week
dddd	Full name of the day of the week
MM	Numeric representation of the month, with leading zeros
MMM	Three-letter abbreviation for the name of the month
MMMM	Full name of the month
yy	Year represented by only the last two digits
yyyy	Year represented by all four digits

Variable Value - Provide a value for the variable based on the Date format or your own format using the syntaxes associated with it.

The default value is always based on current date.





The value of a variable is set when the set variable action is executed.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

SET TIME VARIABLE

Add a time variable to your scenario.

Variable Name - Provide a meaningful name to the variable.

Variable Scope - Specify the level at which you want to access your variable. To learn more about variable scope, please refer to [this article](#).

Time Format - Select a Time format.

Below lists the syntaxes associated with Time format:

Syntax	Description
hh	Hours with leading zeros (12-hour clock)
HH	Hours with leading zeros (24-hour clock)
mm	Minutes with leading zeros
ss	Seconds with leading zeros
tt	Multicharacter time-marker string (for example, "AM" and "PM")

Variable Value - Provide a value for the variable based on the Time format or your own format using the syntaxes associated with it.

The default value is always based on current time.



The value of a variable is set when the set variable action is executed.



Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

SET NUMBER VARIABLE

Add a number variable to your scenario.

Variable Name - Provide a meaningful name to the variable.

Variable Scope - Specify the level at which you want to access your variable. To learn more about variable scope, please refer to [this article](#).

Variable Value - Enter your variable value or use the pull-down menu to select a random number range [Low-High].



The value of a variable is set when the set variable action is executed.

Number of Decimals - Specify a number of decimal places that you want to use.

Decimals Separator - Enter a character to separate the integer part from the fractional part of a number written in decimal form. (comma, period, space, etc.)

By default, ScenarioBuilder uses period (.) character.

Thousands Separator - Enter a character to separate the integer part from the fractional part of a number written in decimal form. (comma, period, space, etc.)




Currency Symbol - Enter a symbol if you want to display numbers as monetary values. The currency symbol appears right next to the first digit. (\$)

Negative Symbol - Enter a symbol you want to use for negative numbers. The number can be displayed with parentheses, braces, brackets around it.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

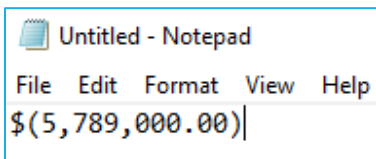


Example:

	Set Number Variable Total Number Script Number -5789000 2	(Step 1)
	Start Application notepad.exe	(Step 2)
	Type Text <i>Total</i>	(Step 3)

Properties <input checked="" type="checkbox"/> Pin This	
Variable Name	Total
Variable Scope	Script <input type="button" value="v"/>
Variable Value	-5789000 <input type="button" value="v"/>
Number Of Decimals	2
Decimals Separator	.
Thousands Separator	,
Currency Symbol	\$
Negative Symbols	0
Scenario Section	

Result:



STRING FUCTIONS

ScenarioBuilder has strong text handling capabilities. Read below to understand the string functions that are available for manipulating text strings.

Variable Name - Enter the name of the variable you wish to manipulate (the "Set String Variable" action must precede this step, establishing the variable to be used).

Function: Select a function.

FirstCapital Function: Returns text with the first letter of each word capitalized.







Split Function: Splits text into a list of values by separating the text at specified delimiter.

You can get the values out of the list by below syntax:

variable name[index]

The list values' indices start at 1 and increment by 1, so the first value's index is 1, the second value's index is 2, etc.

Example:

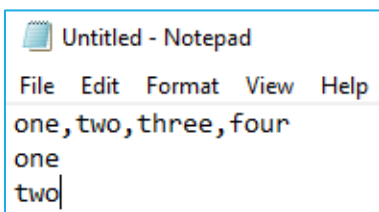
 Set String Variable (Step 1) Text one,two,three,four Script
 String Functions (Step 2) Text Split ,
 Start Application (Step 3) notepad.exe
 Type Text (Step 4) <u>Text</u> <u>Text[1]</u> <u>Text[2]</u>

Properties Pin This

Text	%Text%
------	--------

%Text%
%Text[1]%
%Text[2]%

Result:



Untitled - Notepad

File Edit Format View Help

one,two,three,four
one
two|





ToLowerCase Function: Returns an expression converted to all lowercase (non-capital) letters.

ToUpperCase Function: Return an expression converted to all uppercase (capital) letters.

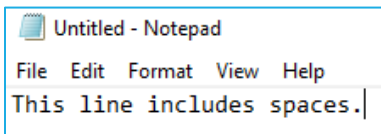


Trim Function: Removes whitespace or tab character from the beginning and end of the source text value.

Example:

	Set String Variable Text This line includes spaces. Script	(Step 1)
	String Functions Text Trim	(Step 2)
	Start Application notepad.exe	(Step 3)
	Type Text <i><u>Text</u></i>	(Step 4)

Result:



Split Delimiter - Enter an arbitrary delimiter (comma, period, space, pipe, etc.)

Reg Expression - Use a regular expression to read the execution results. For example, for ping www.automai.com, expression #output# will place what the command returns in the variable output, and expression sent = #sent# will place what is between =space and comma in the SENT variable.

Continue On Failure: If checked the expression of the scenario will continue even if there is a failure.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.



DATE FUCTIONS

Read below to understand how to use and manipulate values representing dates.

Variable Name - Enter the name of the variable you wish to manipulate (the "Set Date Variable" action must precede this step, establishing the variable to be used).

Function: Select a function.

AddDays Function: To add a number of days to a date.

AddMonths Function: To add a number of months to a date.

AddYears Function: To add a number of years to a date.

Format Function: Returns a formatted value as defined by Date format.

LastDayOfTheMonth Function: Returns the last day of a month, considering leap years too.

Example:

	Set Date Variable Date Script Date dd/MM/yy 13/02/18	(Step 1)
	Start Application notepad.exe	(Step 2)
	Type Text <u>Date</u> , Enter, Press	(Step 3)
	Date Functions Date LastDayOfTheMonth	(Step 4)
	Type Text <u>Date</u> , Enter, Press	(Step 5)

Result:

Untitled - Notepad				
File	Edit	Format	View	Help
13/02/18				
28/02/18				



SubtractDays Function: To subtract a number of days from a date.

SubtractMonths Function: To subtract a number of months from a date.

SubtractYears Function: To subtract a number of years from a date.

Add/Subtract Value - Enter a value you want to add to or subtract from the variable value.

Date Format - Select a Date format.

Continue On Failure: If checked the expression of the scenario will continue even if there is a failure.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

TIME FUCTIONS

Read below to understand how to use and manipulate values representing times.

Variable Name - Enter the name of the variable you wish to manipulate (the "Set Time Variable" action must precede this step, establishing the variable to be used).

Function: Select a function.

AddHours Function: To add a number of hours to a time.

AddMinutes Function: To add a number of minutes to a time.

AddSeconds Function: To add a number of seconds to a time.

Format Function: Returns a formatted value as defined by Time format.

SubtractHours Function: To subtract a number of hours from a time.

SubtractMinutes Function: To subtract a number of minutes from a time.

SubtractSeconds Function: To subtract a number of seconds from a time.



Add/Subtract Value - Enter a value you want to add to or subtract from the variable value.

Time Format - Select a Time format.

Continue On Failure: If checked the expression of the scenario will continue even if there is a failure.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

INCREMENT A VARIABLE

Use the Increment A Variable Action when you want to increment a set variable.

Variable Name: Enter the name of the variable you wish to increment (the “Set A Variable” Action must precede this step, establishing the variable to be incremented).

Increment Value: Enter the value that you wish to increment the entered variable by.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

LOAD AN IMAGE INTO A VARIABLE

This action is used when setting a previously image within an area as a variable.

Variable Name: This is the variable where the text is stored.

Use Anchor: If enabled, mouse location is relative to the parent *Image* or *Window* Action (property is only relevant if the “Mouse Move” Action is a child of either an Image or Window Action)

Image Area: Click the ellipses ... to use your mouse to select the area from which to capture image. Capture only the desired image. Capturing extraneous data, even pixels from surrounding area, can cause OCR to fail. Note: the “Start” and “End” Image Area “X” and “Y” property fields will automatically populate after you make your selection.



Start Image Area X: Image search area coordinates upper left corner X. If needed, you can replace the numerical coordinate value with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftX%)


Start Image Area Y: Image search area coordinates upper left corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %UpperLeftY%)

End Image Area X: Image search area coordinates lower right corner X. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightX%)

End Image Area Y: Image search area coordinates lower right corner Y. If needed, you can replace the numerical coordinate value(s) with a variable. Remember to surround the variable with percent signs (e.g. %LowerRightY%)

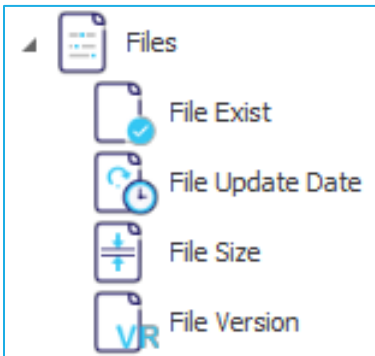
Click the '✕' to remove the "Image Area".

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

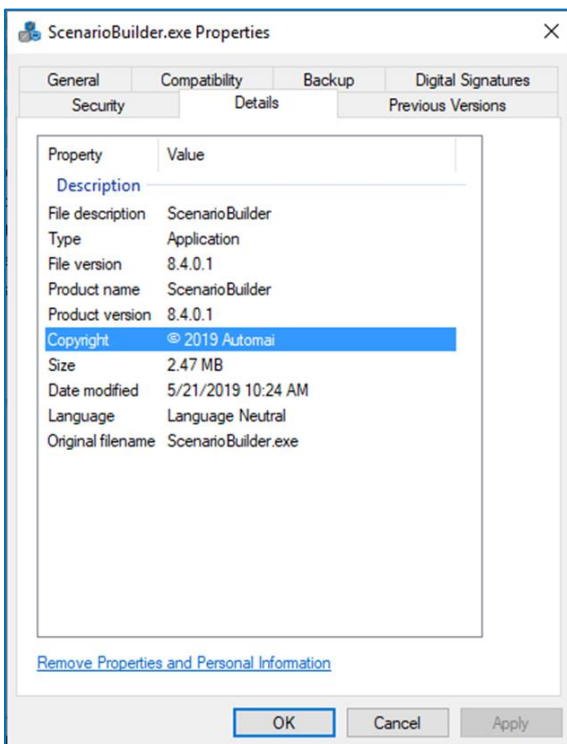
 For more about Load An Image Into A Variable action, please check out "[Load Image As Variable](#)" in our [Video Tutorials](#) series.



Files



The Update Date, Size, and Version actions all read the properties available for a file. You can see these properties by right-clicking on the file in File Explorer and then clicking Properties from the pop-up menu.



FILE EXISTS

This action will check for the existence of a file. An example of how to use this action is to make sure a file exists before the script tries to use it. If not, then you can have it fail rather than continue.

File Path - Provide the full path and file name of the file you want to check if it exists or not. (e.g. "c:\program files (x86)\automai\scenariobuilder\license\sbllicense.lic")



Output Variable - Provide a variable name to hold the output. Output values are SUCCESS or FAILURE.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

FILE UPDATE DATE

If the file exists, this action will retrieve the Date Modified property for the file. This information is stored in a variable. An example of how to use this action is with a process that updates a file. You can use this action to obtain the update date/time for the file and then verify it against current date/time to verify it was updated by the script.

File Path - Provide the full path and file name of the file you are checking. (e.g. "c:\program files (x86)\automai\scenariobuilder\license\sblicense.lic")

Output Variable - Provide a variable name to hold the output. If successful, it will hold the date/time of update. If the file doesn't exist it will report FAILURE.

Note that the date and time are in a string format, not a date and time format.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

FILE SIZE

This action will retrieve the size of a file (in bytes). An example of how to use this action is to verify the file isn't an empty (0 byte) file.

File Path - Provide the full path and file name of the file you are checking. (e.g. "c:\program files (x86)\automai\scenariobuilder\license\sblicense.lic")

Output Variable - Provide a variable name to hold the output. If successful, it will hold the size of the file in bytes. If the file doesn't exist it will report FAILURE.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.



FILE VERSION

This action is only valid for executable (EXE) files. This action will obtain the File Version property. An example of how to use this action is to verify you have the correct version of an application before completing upgrade testing. It has three properties.

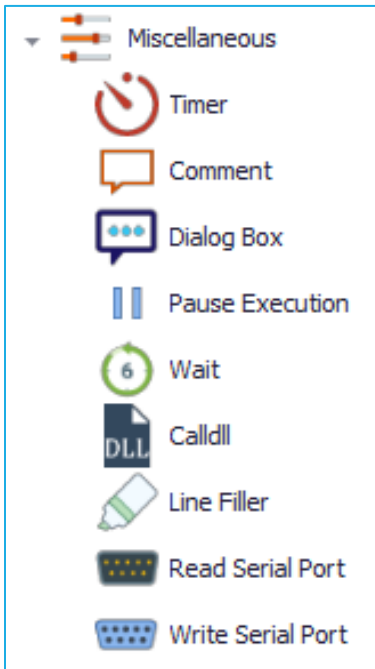
File Path - Provide the full path and file name of the file you are checking. (e.g. "c:\program files (x86)\automai\scenariobuilder\ScenarioBuilder.exe")

Output Variable - Provide a variable name to hold the output. If successful, it will hold the file version. If the file doesn't exist or the File Version property is empty it will report EMPTY.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.



Miscellaneous



TIMER

This feature allows you to stop and resume the execution timer. It can be very useful when you want to perform certain Actions (such as cleaning up an environment), which you do not wish to be reported in the overall execution time.

Use the pull-down menu in the Properties pane to select “Suspend” or “Resume” Timer. The time spent between the Suspend Timer and Resume Timer steps will not be included in the reported execution time.

Variable Value: Choose the action you want to perform on the timer. The time spent executing the actions between Suspend Timer and Resume Timer will not be included in the reported execution time.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.



COMMENT

Use this Action to add a descriptive comment as needed. The Comment Action has no effect on the scenario itself; rather, it can be used as a helpful note or explanation you can add for yourself or anyone who might be working with your scenario.

Comment: Enter descriptive comment.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

DIALOG BOX

In the case of attended automation, where a person is involved with providing feedback in the automation process, the Dialogue Box is the window of interaction. It gives feedback via popups with info, warning or error messages. It requests information from the user in order to decide how to proceed.

The Dialog Box action can be used to:

- Present information or warn the user of errors and to prompt for directions on how to proceed after an error has occurred.
- Display messages and get input from the user.
- Ask the user for directions when the robot doesn't know how to continue the process.
- Insert info in your test for specific actions (passwords or info that was not saved in ScenarioBuilder).

Dialog Box Title - Provide a meaningful name to the dialog box. By default, it is named as "Automai - Dialog Box".

Dialog Box Icon - Use the pull-down menu to select a type for the icon to show in the dialog box. The types can be Error, Info or Warning.

Dialog Box Message 1 - Specify the first message that you want to show in the dialog box.

Set Variable - [Optional] Enter the name of the variable you want to set its value. This can be an existing or a new variable.

Dialog Box Message 2 - [Optional] Specify the second message that you want to show in the dialog box.



Timeout - Duration in seconds to wait for user input, or close the dialog box. If you want the dialog box to wait for user input indefinitely, put 0 value.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

PAUSE EXECUTION

This action will pause the execution in ScenarioBuilder.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

WAIT

Wait periods allow you to add a step in your scenario to give the system enough time to execute the Action, or to load the window/program properly. Use the Wait Action when you need to add more time to execute a line of Action. After adding Wait to your Scenario, enter the wait time (in seconds) in the right pane. The default wait time is 10 seconds.



Make sure the wait time plus the rest of the scenario time does not exceed the playback time.

Time in Seconds: Number of seconds to wait.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

CALL DLL

Use CallDll to call a dll function.

Dllname: Enter the name of the desired dll; for example, user32.dll. It is important to include the .dll extension.



Function: Add the function of the dll.

Parameter: Enter the parameter. You can add as many parameters as needed using the + icon.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

LINE FILLER

This Action adds a colored line within your scenario. It has no impact on the scenario playback, but is simply used to help you visually organize your scenario. You can modify the line's color by editing the R, G, B values in the Properties window.

READ SERIAL PORT

This Action allows you to read data from ScenarioStation's serial port. After adding Read Serial Port to your Scenario, you will need to configure the settings in the right pane. Select the appropriate port number (i.e. 1 or 2), and then configure the port settings (baud rate, data bits) as needed.

In addition to the port settings, **Reg Expression** uses a regular expression to read the execution results. For example, for ping www.AppsWatch.com, expression #output# will place what the Action returns in the variable output, and expression sent = #sent# will place the contents between =space and comma in the SENT variable.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.

WRITE SERIAL PORT

This Action allows you to write data to a ScenarioStation's serial port.

Select the appropriate port number (i.e. 1 or 2), and then configure the port settings (baud rate, data bits) as needed.



In addition to the port settings, **Reg Expression** uses a regular expression to read the execution results. For example, for ping www.AppsWatch.com, expression #output# will place what the Action returns in the variable output, and expression sent = #sent# will place the contents between =space and comma in the SENT variable.

Scenario Section - This is for Global scenarios only. Specify the section name that can be called from other scenarios.




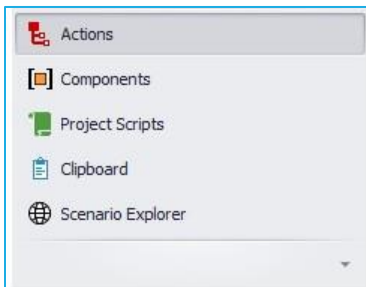
Components

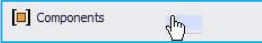
Not all Actions operate on the application under test. Some Actions operate on the scenario itself. For example, as demonstrated in the preceding tutorial, a “Transaction” directs ScenarioBuilder to measure the steps contained between its start and end points. And a Component, while technically an Action, is really more of a structural element in the scenario.

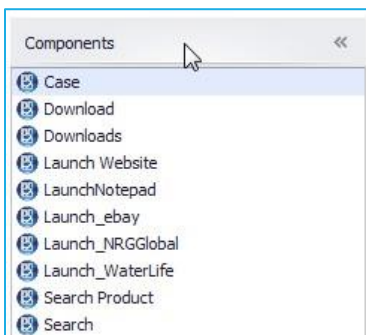
A Component is a user-defined shell within a scenario that contains a single Action or a series of Actions. Think of Components as containers for Actions related to a specific task in the overall process. Breaking up scenarios into Components makes for easier scenario maintenance, and can be helpful in conceptualizing a scenario design.

Components are self-contained units that can be called from any scenario within a Project. Additionally, Components can be exported and imported to and from Projects. If a modification is made to a Component that is being used by multiple scenarios, the Component is updated for all scenarios.

By default, the Components pane is collapsed on the left side of ScenarioBuilder inside the Actions/Components/Project Scripts/Clipboard/Scenario Explorer window. To access it click on  in the bottom of the Actions/Components/Project Scripts/Clipboard/Scenario Explorer Window.




Click the down arrow icon  to expand the pane;

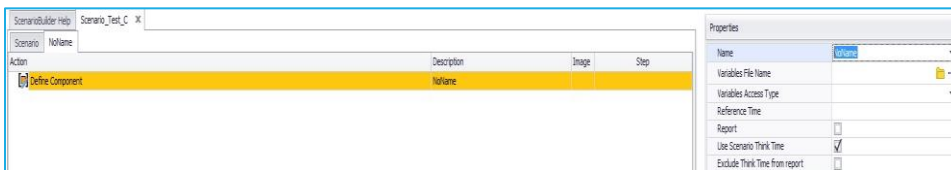


The pane contains the Components in the active Project, which you’ve created or imported. These Components can be shared among different scenarios throughout the Project, and exported for use in other Projects. All scenarios in the active Project that share a Component in the active Project will inherit any changes made to that Component.

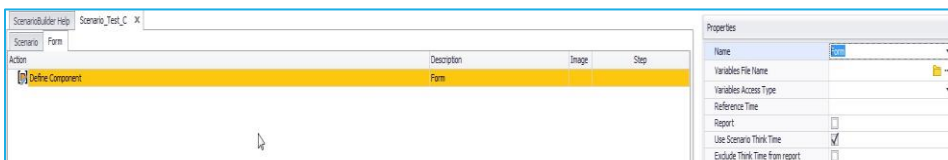
Double-click a Component to add it to your scenario.

Define components

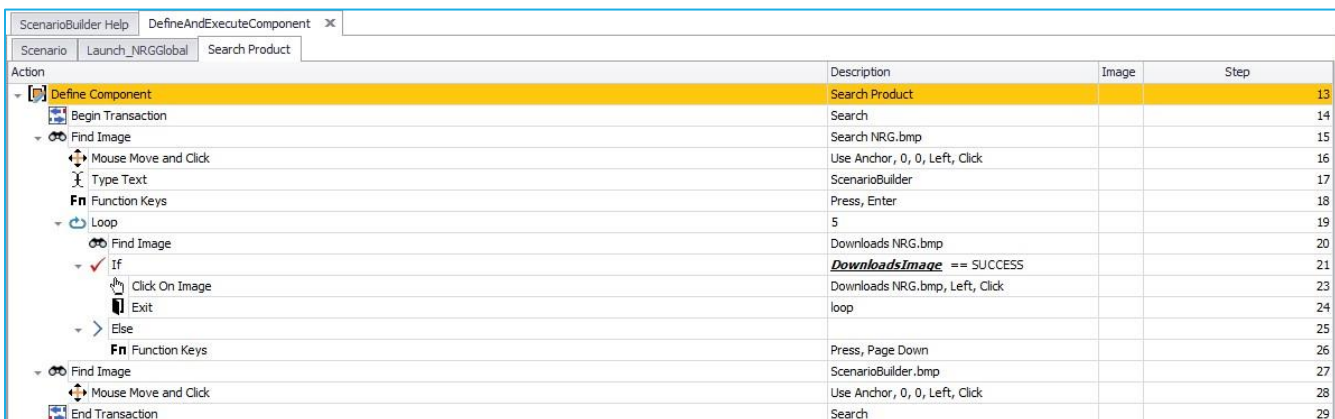
To create a new Component, double-click or drag the “Define Component”  Define Component Action from the “Applications” section of the Actions pane into the Scenario Window. ScenarioBuilder adds a tab called “NoName” to the Scenario Window;



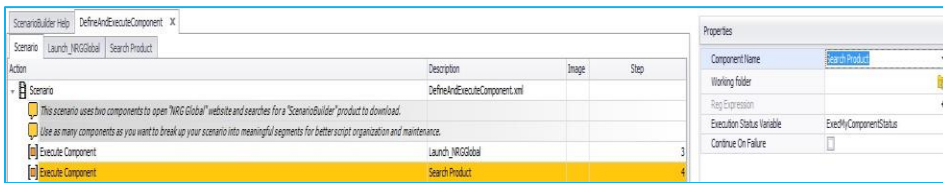
Update the Component’s “Name” property with a meaningful name (notice the tab name changes);




With the Component tab selected, add Actions by double-clicking or dragging into the Scenario Window;

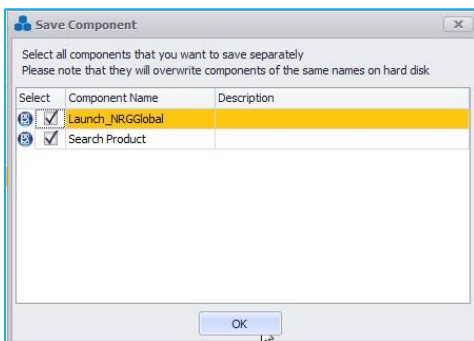


Click the “Scenario” tab. Add an “Execute Component” Action to the scenario. Select the Component from the “Component Name” drop-down box;



Save components

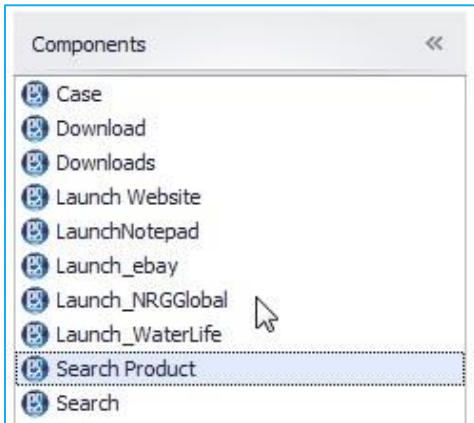
ScenarioBuilder will prompt you to save Components that have been added or modified as part of the Scenario saving process. Click the “Save Scenario”  toolbar icon, or select “Save Scenario” from the “File” menu. If a new Component has been added to the Scenario, or a pre-existing Component has been modified, the “Save Component” window will open;



Select the Components you wish to save and click “OK”. Note that when you save a Component that is shared by other Scenarios in a Project, you are updating it globally, meaning any changes are effective for all Scenarios in which it is shared. If this is not your intention, close the “Save Component” window, re-name the Component, and repeat the “Save” process.

After saving the Scenario, the Components pane is updated with any new additions;

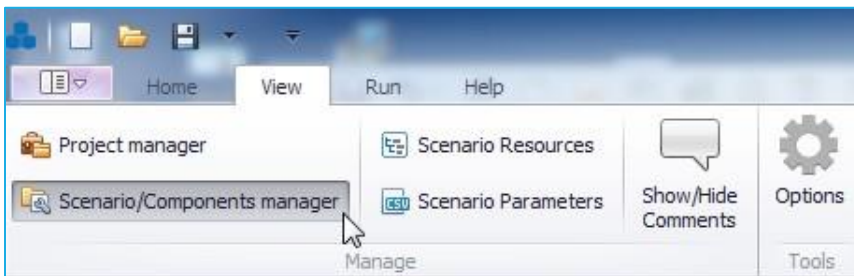




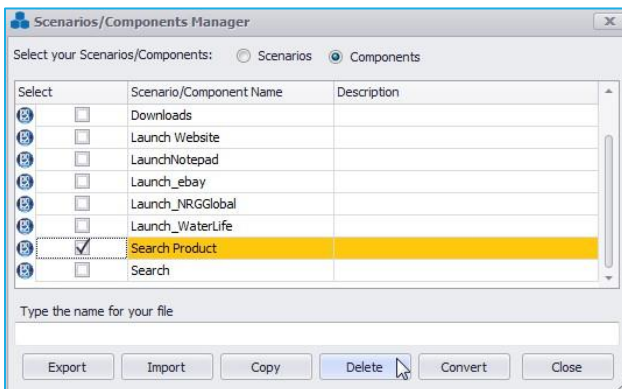
Components will remain in the Components pane for the current Project, unless deleted.

Delete a component

To delete a Component, select “Scenarios and Components Manager” from the *View* tool bar.

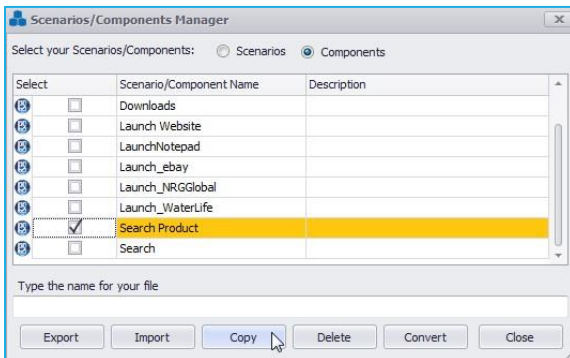


In the “Scenarios/Components Manager” window click the “Components” radio button. Select the Component to delete and click the “Delete” button;

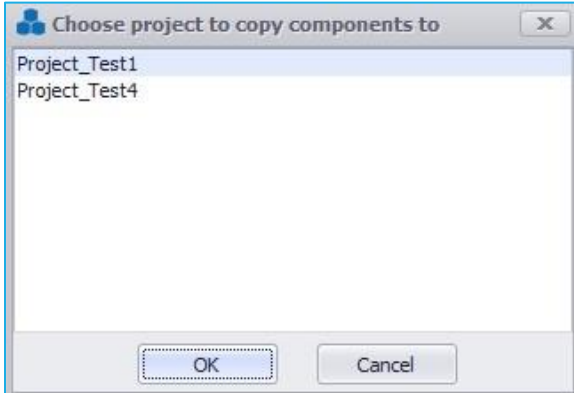


Copy a component to another project

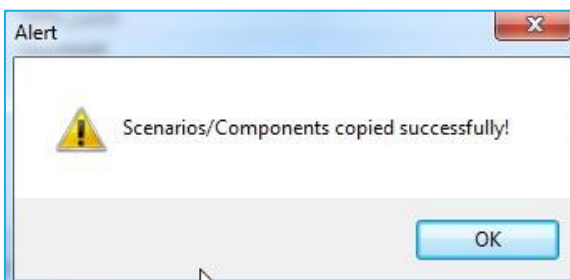
You can directly copy a Component from the current Project to another Project (provided the Projects reside in the same “Base folder for All Projects” which is established in the “Options” menu). Select “Scenarios and Components Manager” from the File menu. In the “Scenarios/Components Manager” window click the “Components” radio button. Select the Component to copy and click the “Copy” button;



The “Choose project to copy Components to” window pops up. All of the Projects in the current base folder are displayed;



Select the Project to which you would like to copy the Component and click “OK”.

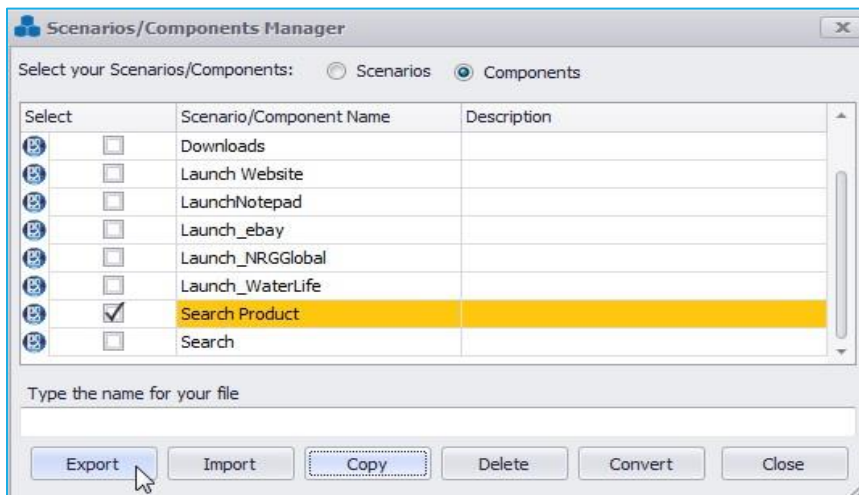


Export and import components

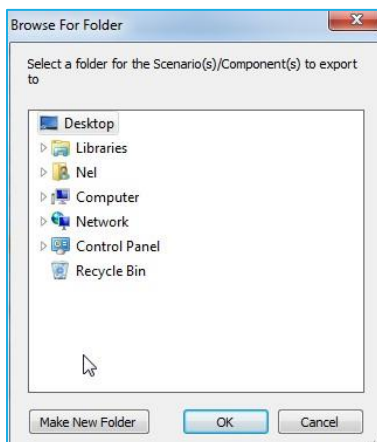
Use “Import” and “Export” to move Components in and out of Projects residing in different work spaces.

EXPORT COMPONENT

In the “Scenarios/Components Manager” window click the “Components” radio button. Select the Component(s) to export and click the “Export” button;

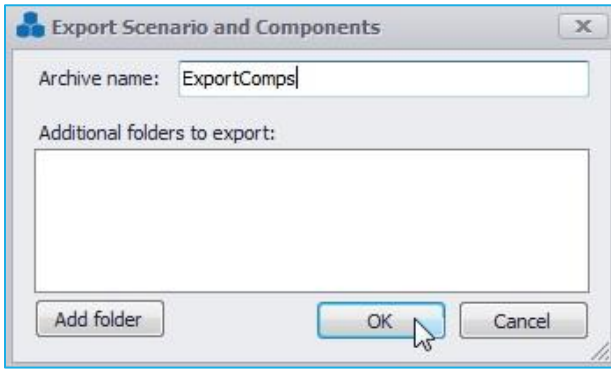


Navigate to an existing folder or create a new folder into which to export. Note that the export folder is a neutral location from which the Component(s) will be imported; it is not intended to be the final location (i.e. the Project folder) for the Component(s). Click “OK”;




The “Exporting Grouped Files” window pops up. Enter a name for the exported set of files and click “OK”;





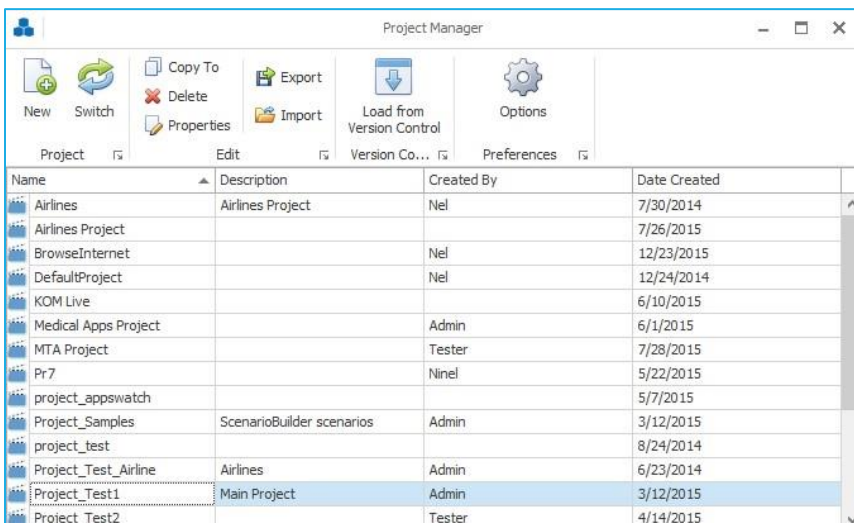
Your exported Component(s) will be saved as a .zip or .rar folder in the folder you selected;




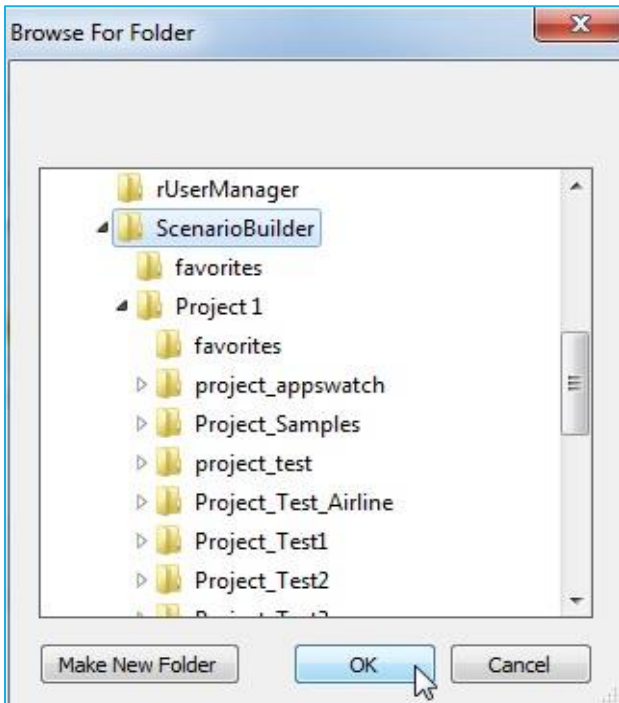
 You can also add additional folders such as “Documents” to your scenario.

IMPORT COMPONENT

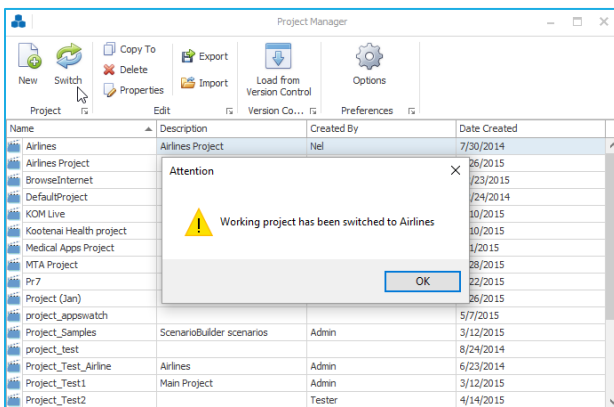
After exporting your Component(s), open the existing Project, or create a new Project into which you wish to import the Component(s). If importing to a Project in a different base folder, open the “Project Manager” from the File menu;



Click on “Options” button, then click the ellipses  button in the “Project Preferences” pane to navigate to the desired Project base folder;

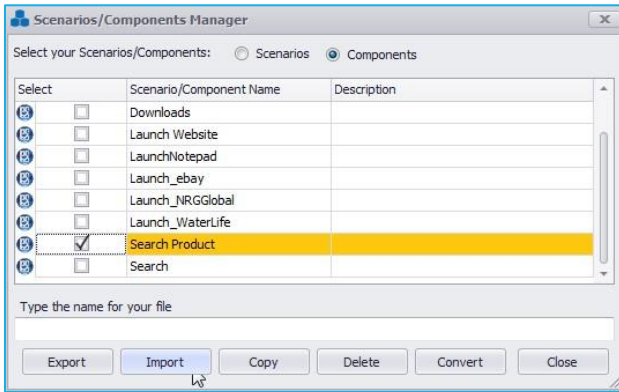


Select the folder and click “OK”. Next, select the Project to which you want to switch and click “Switch Project” button. ScenarioBuilder will notify you that the Project folder has been switched;

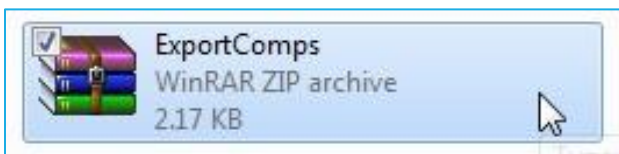


Now that you are working in a new Project folder, you can import the Component(s). Open the “Components and Scenarios Manager” and click “Import” button;

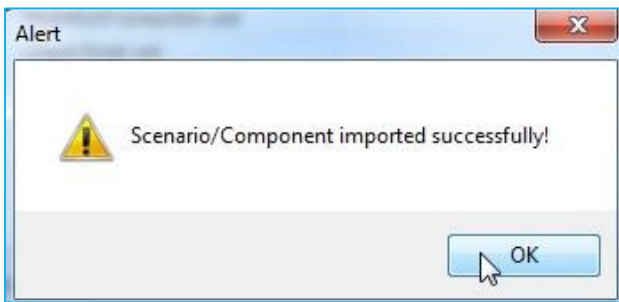




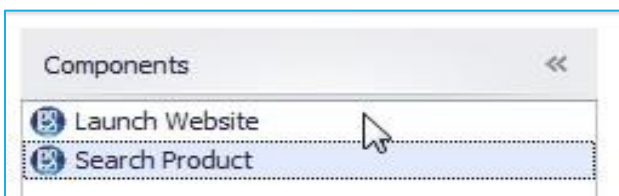
Browse to the folder containing the exported Component(s);



Double-click the desired .zip file. ScenarioBuilder will acknowledge the import;

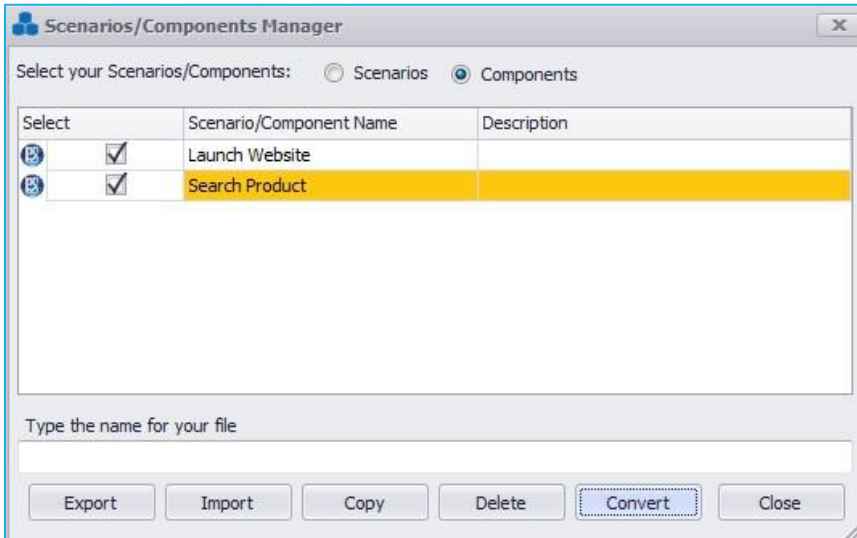


Imported Component(s) will appear in the Project's "Components" pane;

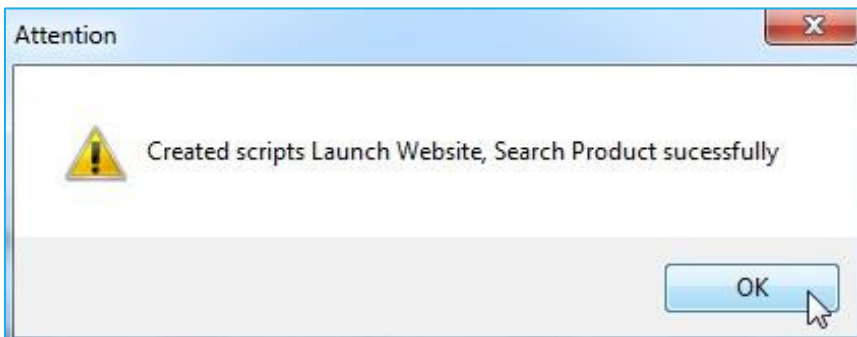


Convert components to scenarios

In the Scenarios/Components Manager, click the radio button next to Components and click the check box next to the Components to convert;




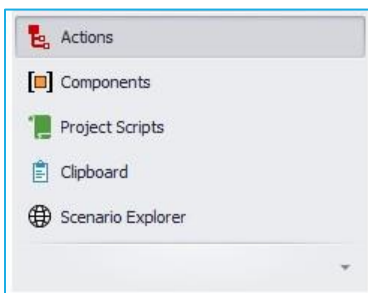
Click "Convert". ScenarioBuilder converts the selected Components into scenarios (scripts). Note that the Component remains intact as well.




Project scripts

A scenario or script is an XML file created with ScenarioBuilder. A scenario contains Actions that emulate the activity of a user interacting with an application. Scenarios can be called in other scenarios (known as [nesting](#)) with the “Play Scenario” Action.

By default, the Project Scripts pane is collapsed on the left side of ScenarioBuilder inside the Actions/Components/Project Scripts/Clipboard/Scenario Explorer window. To access it click on  in the bottom of the Actions/Components/Project Scripts/Clipboard/Scenario Explorer Window.



Click the down arrow icon  to expand the pane;

Play script

Scenarios can be called by other scenarios with the “Play Script” Action. This technique, (also known as “nesting”), can save your time when developing a set of test scenarios that call the same steps. For example, say you want to create the following set of scenarios to test your Email application:

A scenario that logs onto the application can be created for example “email_log_in” below





Action	Description	Image	Step
Scenario	email_log_in.xml		
Find Image	email_login_Img4.bmp		1
Mouse Move and Click	Use Anchor, -67, 97, Left, Click		2
TXT Type Text	<u>email</u>		3
Fn Function Keys	Press, Tab		4
TXT Type Text	<u>password</u>		5
Fn Function Keys	Press, Enter		6
Find Image	email_login_Img2.bmp		7

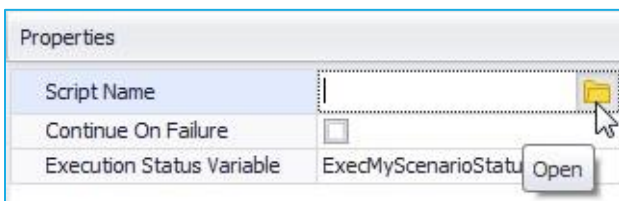
This scenario can then be called by other scenarios that require the log in to email process. For example the scenario below has the “Play Script” action in step 9 that calls the previously created “email_log_in” scenario.

Action	Description	Image	Step
Scenario	email_log_in_comp.xml		
Open with File Association	http://www.yahoo.com		1
Maximize Window	Maximize, Yahoo - Mozilla Firefox		2
Begin Transaction	App_Launch		3
Find Image	email_login_Img1.bmp		4
Mouse Move and Click	Use Anchor, -78, -72, Left, Click		5
TXT Type Text	outlook.com		6
Fn Function Keys	Press, Enter		7
End Transaction	App_Launch		8
Play Script	email_log_in.xml		9

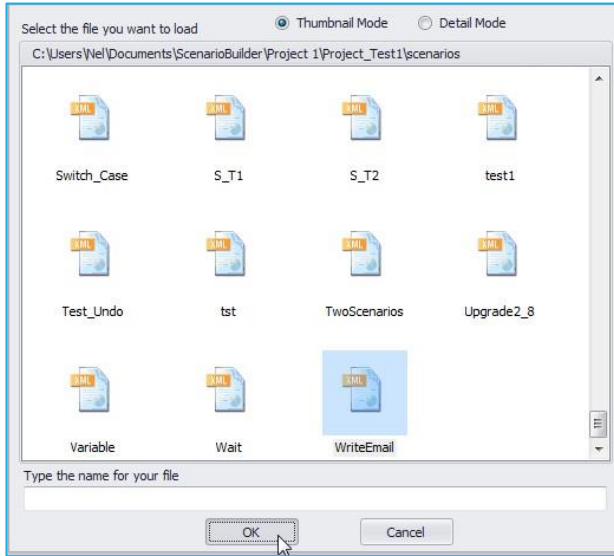
Active scenarios can be edited by simply clicking their tab to display their steps in the Scenario Window.

To add a scenario using “Play Script”, first add the “Play Script” action  Play Script

Then in the Properties section of the “Play Script” action click on the  icon in the Script Name field.



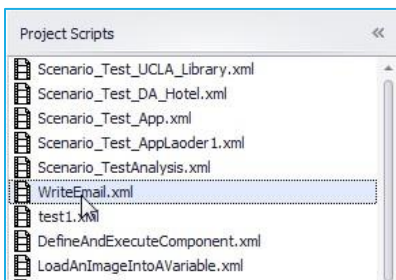
Select the script that will be called on in the “Play Script” action



Select the desired scenario from the pop-up window to complete the Action;



The “Play Script” Action can also be invoked by selecting from the “Projects Scripts” pane. On the left side of ScenarioBuilder, expand the “Projects Scripts” pane (below the “Actions” and “Components” panes) to display all the scenarios from the current project;



Double-click the desired scenario. ScenarioBuilder automatically adds it, via the “Play Script” Action, to the current scenario;



Workflows

Nesting scenarios can also be used to create “Workflows” for Load Tests. A Workflow is simply a Scenario, serving as a scheduler, which plays a set of scenarios in a certain order. A Workflow does not generate an overall report, but it does generate a report for each of the individual scenarios it plays. Workflows can be useful in load testing with AppLoader, as they allow you to create a test plan within the test plan, providing a means to ensure that the same users proceed through a series of processes. For example, if testing the following processes:

- Log into email
- Write and send email
- Log off of email

Create a workflow like this;

Action	Description	Image	Step
Scenario	EmailWorkFlow.xml		
Play Script	email_log_in.xml		1
Play Script	WriteEmail.xml		2
Play Script	email_log_off.xml		

Create multiple workflows to replicate how real users would impact an application.

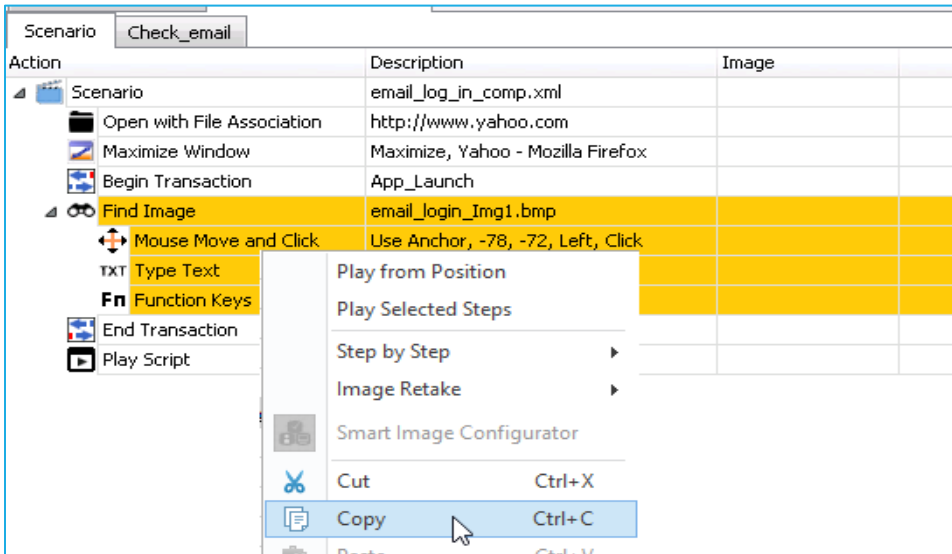


Click here for more about [Using Scenarios with AppLoader](#).

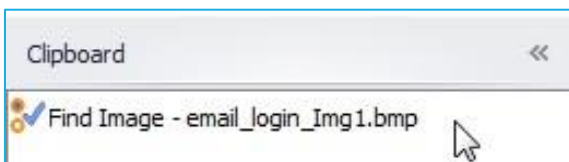


Clipboard

Copy an Action or a series of Actions to the Clipboard to be recalled later in the current scenario. Actions copied to the Clipboard retain their Property settings. Use the Clipboard when Actions with the same properties are repeated in different places within the scenario. Highlight the Action(s) and right-click. Choose “Copy” from the menu;

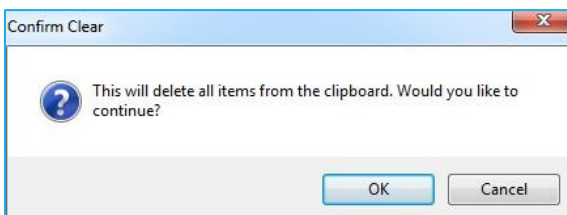


The item(s) will be added to the Clipboard (under the name of the first row of selected Actions);



Double-click or drag the “Clipboard” item to add it to the Scenario, just as you would with any Action.


To clear the Clipboard, select “Clear Clipboard”  from the Home tool bar menu;

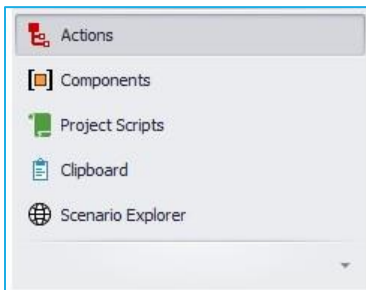


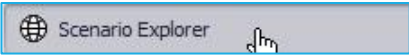
Click “OK” to confirm clear. Note that the Clipboard is cleared whenever ScenarioBuilder is closed.

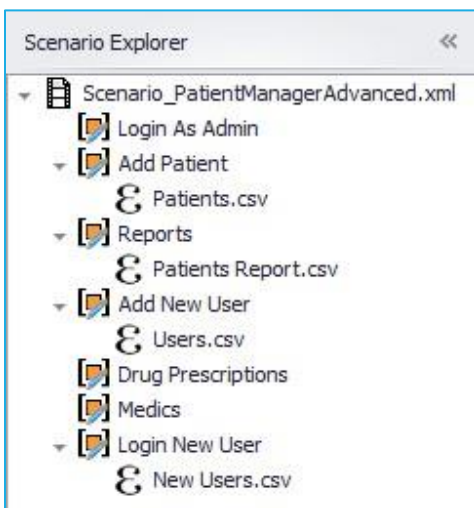


Scenario explorer

By default, the Scenario Explorer pane is collapsed on the left side of ScenarioBuilder inside the Actions/Components/Project Scripts/Clipboard/Scenario Explorer window. To access it click on  in the bottom of the Actions/Components/Project Scripts/Clipboard/Scenario Explorer Window.



Click the down arrow icon  to expand the pane;



The pane contains the current scenario's script(s), components and CSV files.

Once you double-click on a component or CSV file, they will be opened, so you can view and edit them.



Working with images

Image Actions are ScenarioBuilder’s most versatile Actions. Images can be used to start an application, confirm that an application or web page has loaded, or hold until an application has closed.

An image can be any kind of bitmap – a company logo on a client application; a photo on a web page; or a data entry field on an application form.

As an example, if testing the <http://www.automai.com> website, we could verify that the home page has successfully loaded by asking the scenario to check for the Automai logo. A “Find Image” Action would search for the logo until its time out period is exhausted. Subsequent Actions would hold until the outcome of the “Find Image” Action is determined.

CONTACT US FREE TRIAL

automai™
FORMERLY NRG GLOBAL

PRODUCTS ▾ SOLUTIONS ▾ RESOURCES ▾ COMPANY ▾

Why NRG Global is changing to AUTOMAI.
To understand better, let's tell you our story from the beginning, back in 2000: READ THE ANNOUNCEMENT

Get on to Bigger Things

Accelerate your business with Robotic Automation.
Now get the job done without sweating the small (and repetitive) stuff.

DOWNLOAD A FREE TRIAL

Free POC in your environment for every product.
See for yourself how easy it can be.

Automai Robotic Automation Platform



Think like a real user

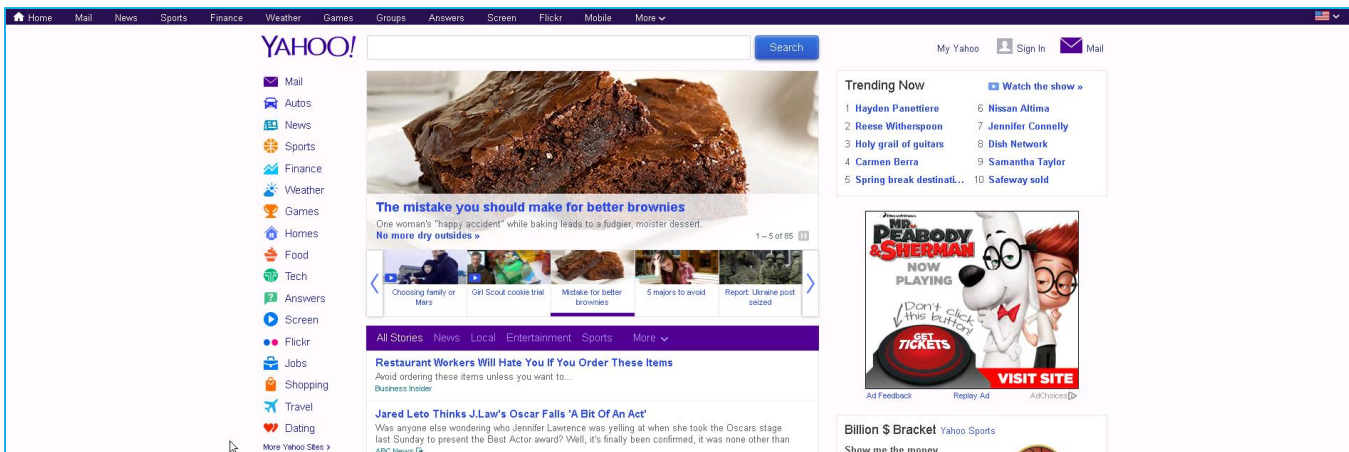
When building a Scenario, think in terms of the end user. How will they interact with the application? What will they do? For example, if the application involves a sign-up or data entry form, a user will not start entering data until the field has properly loaded on the screen. In ScenarioBuilder, that synchronization can be achieved with Image Actions.

IMAGE ACTIONS KEEPS SCENARIOS SYNCHRONIZED

Check out the following example which demonstrates a simple scenario that opens a company’s online job application form, and enters text into the data fields. First the website has to be opened. ScenarioBuilder uses the “Open with File Association” Action to open the URL;

Action	Description	Image	Step
scenario	email_log_in_comp.xml		
Open with File Association	http://www.yahoo.com		1

The yahoo website opens in the default browser;



The next step in the scenario demonstrates a fundamental concept in ScenarioBuilder.

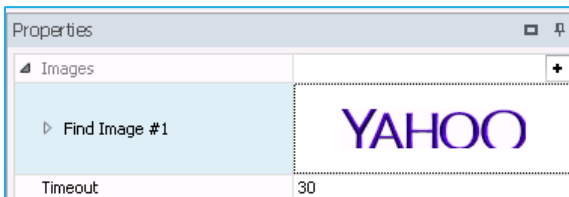
Find Image	email_login_Img1.bmp		4
------------	----------------------	--	---

With a “Find Image” Action, ScenarioBuilder merges real user intelligence with automation. An image of the Yahoo image will be searched as used as an anchor





First, the scenario will look for the field, as a real user would (notice that enough of the field heading is included in the image captured to ensure that it is uniquely identifiable by ScenarioBuilder). ScenarioBuilder will delay subsequent Actions until this image is located. Set the “Time Out” property to provide adequate time for the form to load (factor in a longer response time for multiple users interacting concurrently in your test environment);



Next, the child actions will only play if the parent “Find Image” actions is successful. The “Mouse Move and Click” child Action places the cursor into the url field, notice that this action is not a blind click. The coordinates of the click are based on the location of the image in the “Find Image” action. The “Type Text”, action that comes after will then enter the url address. A real user knows to click onto the url address bar before typing. The last child action is to press the function key enter to submit the url address. This is how ScenarioBuilder achieves proper mouse focus;

Action	Description	Image	Step
scenario	email_log_in_comp.xml		
Open with File Association	http://www.yahoo.com		1
Begin Transaction	App_Launch		3
Find Image	email_login_Img1.bmp		4
Mouse Move and Click	Use Anchor, -78, -72, Left, Click		5
TXN Type Text	outlook.com		6
Fn Function Keys	Press, Enter		7

COMBINE IMAGE AND LOOP ACTIONS TO PAGE DOWN UNTIL IMAGE IS FOUND

Sometimes an image you are looking for on a web page may appear “below the fold”. It might require one, two or many “page downs” before it appears. A real user would quickly scan the screen for the desired image, and page



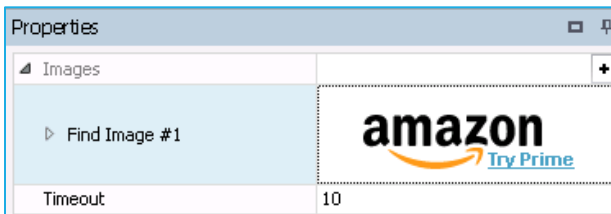
down until the image appears. In ScenarioBuilder that same intelligence can be achieved with Image and Loop Actions. The following example opens Amazon.com;

Action	Description	Image	Step
scenario	Amazon		
Open with File Association	http://www.amazon.com		

Next, the scenario uses an image of the Amazon logo to ensure that the web page has loaded;

Find Image	Amazon_logo.bmp		
------------	-----------------	--	--

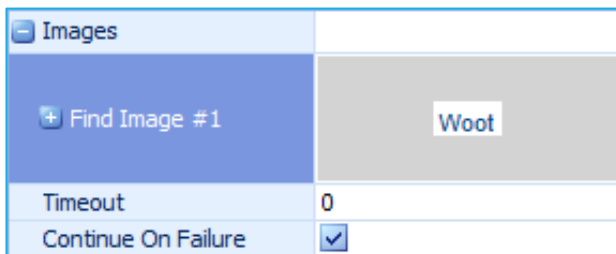
The logo image has a generous “Timeout” value, ensuring that there is ample time for the page to load;



Now the scenario searches for the “Woot” icon located at the bottom of the Amazon home page;



The “Timeout” value for the “Woot” image is minimal;



The scenario knows that the page has already loaded (previous step found the logo image), and within milliseconds knows whether or not the “Woot” image is present. As soon as it scans the screen and realizes the image is not present, it moves on to the next step (note the image’s “Continue On Failure” property is enabled,



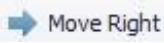

otherwise the scenario would fail and stop after the first attempt to find the image). The complete scenario looks like this;

Action	Description	Image	Step
scenario	Amazon		
Open with File Association	http://www.amazon.com		1
Find Image	Amazon_logo.bmp		2
Loop	5		3
Click On Image	woot.bmp, Left, Click		4
Fn Function Keys	Press, Page Down		6

It's important to note the use of indentation to create a **hierarchy** in the example. Indented Actions are “children” of the Actions under which they're indented. Child Actions only execute if their parent Action has succeeded. The relationship of the steps in a loop (parent to child) must be set up properly or the loop won't function as expected.

Tip For more about How to properly structure a hierarchy, please check out “[Hierarchy Structure](#)” in our [Video Tutorials](#) series.

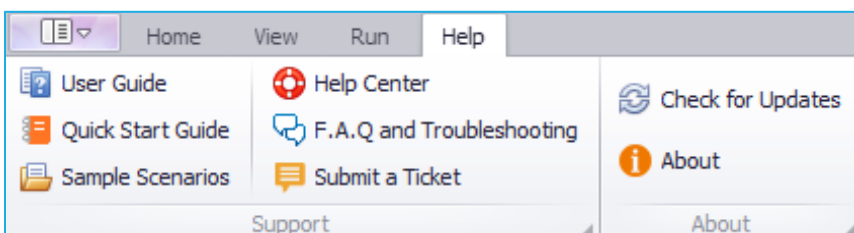
‘Click on Image’ is the first step inside the loop. It's considered inside the loop because it is indented, and therefore a child of the Loop Action.

Use the “Move Right”  and “Move Left”  tool bar icons to shift steps from parent to child or vice versa.

If step 4 succeeds, it will advance through its child Action - step 5- culminating in an *exit* from the loop. If step 4 fails, it will advance (only because “Continue On Failure” is enabled), bypassing its child Action, to step 6. Step 6 executes a “Page Down” function and returns the scenario to step 3 for the next iteration through the loop;

Tip For more about Loops actions, please check out “[Conditional Statements](#)” in our [Video Tutorials](#) series.

Also, see the “Loop/Exit” example in the “Sample Scenarios” section of ScenarioBuilder's “Help” menu;



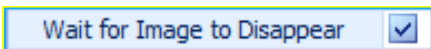
WAIT FOR IMAGE TO DISAPPEAR BEFORE ADVANCING

The “Wait for Image to Disappear” property on a “Find Image” Action can be used to keep a scenario synchronized. Say your web application does not always respond when icons are clicked. Real users know to re-click and wait for the page to load before advancing to the next step, but how do automated users know how to react?

The following scenario opens Yahoo.com and then uses a “Find Image” Action to click on the link to the “Autos” page;

Scenario			
Action	Description	Image	Step
scenario	Doc_open		
Open with File Association	http://www.yahoo.com		
Find Image	Doc_open_Img4.bmp		

Next the scenario uses a “Find Image” Action with the same image of the “Autos” link, but with the “Wait for Image to Disappear” property enabled;



Notice the Action changes to the inverse “Wait Image Disappear” Action;

Scenario			
Action	Description	Image	Step
scenario	Doc_open		
Open with File Association	http://www.yahoo.com		
Wait Image Disappear	Doc_open_Img4.bmp		

The last piece is to add an “On Failure” section which kicks in only if the image does *not* disappear. Two Actions are in the “On Failure” section, a “Mouse Move and Click” Action to click that same “Autos” link followed by a “Resume” Action which instructs ScenarioBuilder to go back to the point of the failure and try again. The “Resume” has a “Retries” property which determines the number of times to loop through the “On Failure” section before giving up. In this example it’s set to 3 times.

The complete scenario looks like this;



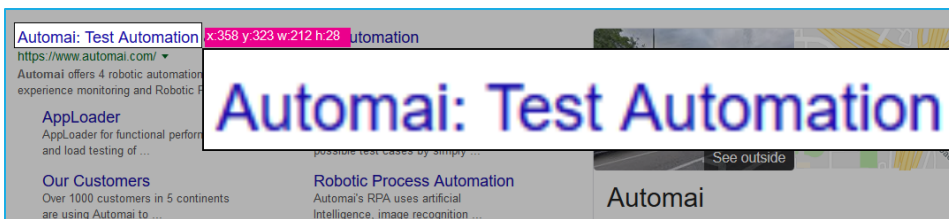
Action	Description	Image	Step
scenario	Doc_open		
Open with File Association	http://www.yahoo.com		
Find Image	yahooLogo.bmp		
Mouse Move and Click	Use Anchor, 89, -26, Left, Click		
Wait Image Disappear	yahooLogoD.bmp		
On Failure			
Mouse Move and Click	288, 276, Left, Click		
Resume	3		

Image variations

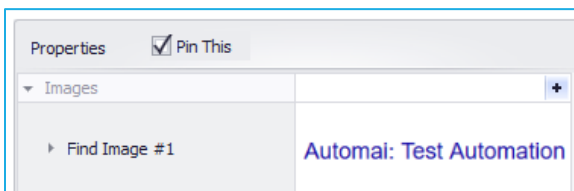
A real user is not thrown off by a slight variation in an icon or the color of a button. However, when automating test scenarios, it is important to take into account possible variations and make allowances for them. ScenarioBuilder provides various means to ensure that images are found under the widest variety of circumstances.

USE MULTIPLE IMAGES FOR AN IMAGE ACTION

A hyperlink will often change colors once it has been followed during a browser session. This image variation can be handled in ScenarioBuilder by first capturing the image in its “un-followed” state;



Once captured, go to the Image Action’s properties and click the “+” in the “Images row;

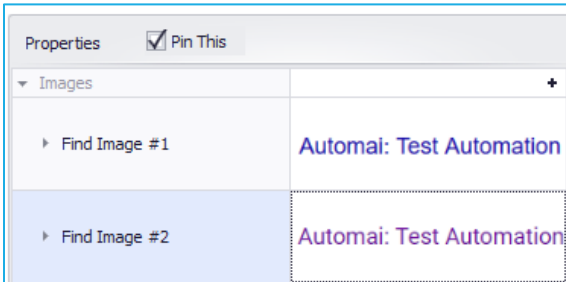


Notice a new property field called “Find Image #2” is added to the pane. Right click on the “No image-Right click for menu” field and capture another image of the hyperlink, this time in its “followed” state;






Now the properties pane includes two images for the Action to seek;

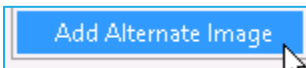


Finding *either* image will make the Action a success.

ADDING ALTERNATE IMAGES THROUGH THE SCENARIO RECORDER

Alternate images can be added to a scenario as it is being recorded. After capturing an Image click on the drop down menu button  in the “Capture Image” icon.

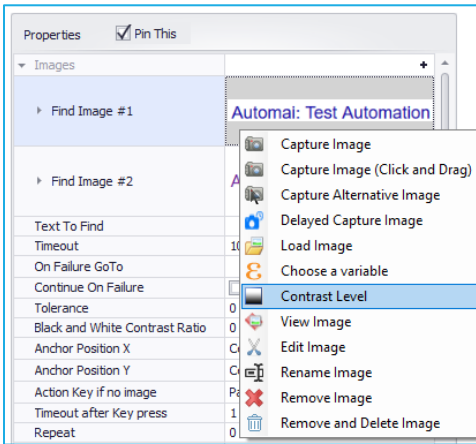
Click on “Add Alternate Image”



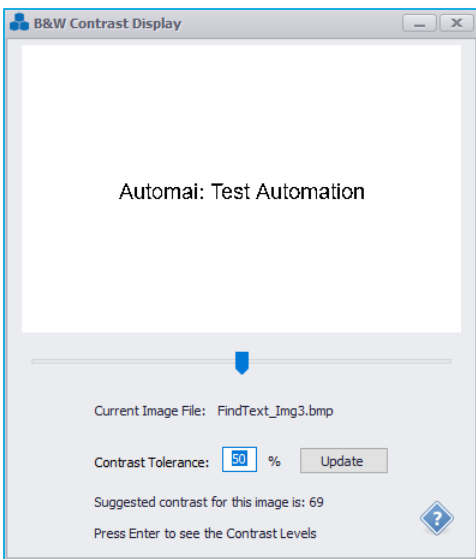
SEARCH FOR IMAGE IN BLACK AND WHITE

An alternate method to handle the problem of a color-changing hyperlink is to search for the image in black and white. ScenarioBuilder provides a tool to set a black and white contrast ratio for an image, which factors out color and searches for the image in grey scale. Right click the “No image-Right click for menu” field in the properties window and select “Contrast Level”;

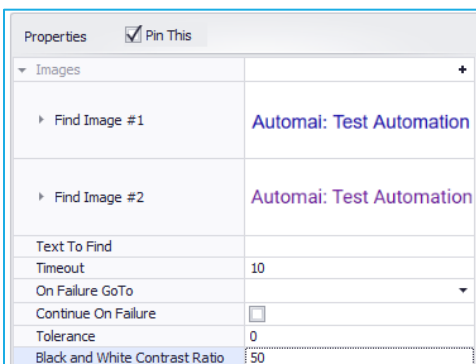




The “B&W Contrast Display” tool opens. Use the sliding scale to establish a black and white “Contrast Tolerance” percentage for the image;

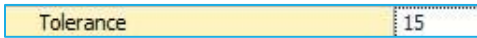


When you settle on a “Contrast Tolerance” that displays the image accurately without under or over saturation, click “Update” to send the setting to the Image properties;



If the “Black and White Contrast Ratio” property is set to any value greater than zero, ScenarioBuilder will look for the image in black and white. To return the Image Action to color, simply reset the value to zero.

For subtle irregularities in image renderings, ScenarioBuilder provides a “Tolerance” property which will increase the image matching error margin;



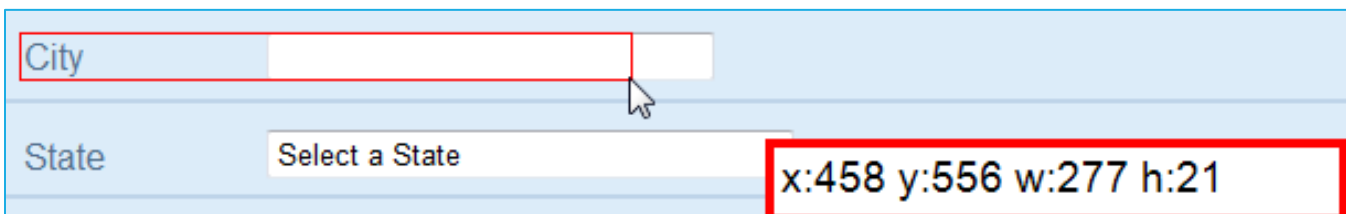
Use this property to establish the maximum percentage of differences allowed between the stored image and the desktop rendering (the higher the percentage, the greater the margin for error). Be aware that a high tolerance (certainly anything over 50%) can lead to false positives.

IMAGE SEARCH AREA

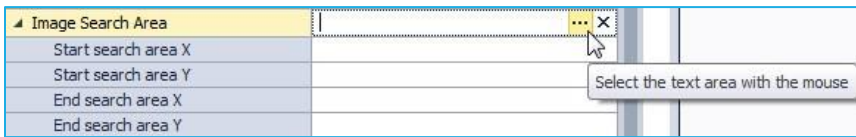
Sometimes a form will have two identically labeled fields. The real user knows which one to click into, but how does ScenarioBuilder know? With the “Search Area” property you can instruct the scenario to search for an image in a specific area of the screen. The following form has duplicate “address” fields with identical field labels;

A screenshot of a web form with two identical sections. The top section is titled "Enter your old address." and the bottom section is titled "Enter your new address." Each section contains fields for Street, City, State (a dropdown menu), and ZIP Code. The Street field has a placeholder text: "Include your apartment, suite number, OR PO Box, if applicable". The ZIP Code field has a button labeled "Auto-fill your ZIP Code". The bottom section also includes a checkbox labeled "This residence was built in the last 6 months." with a link to "More Info".

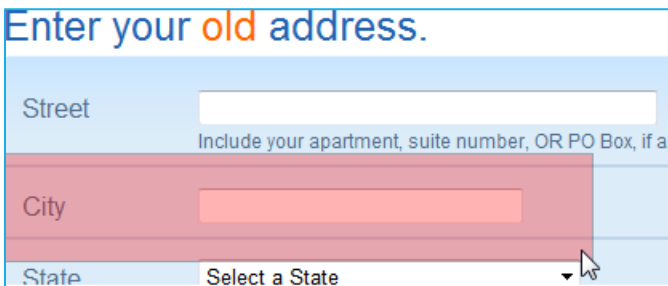
First, with a “Click on Image” Action capture an image of the label and field;



Click the “...” (elipses) icon in the Image Action’s “Image Search Area” property field to activate the “Search Area” tool;



Use the tool to highlight the area of the desktop where you want the search to take place;



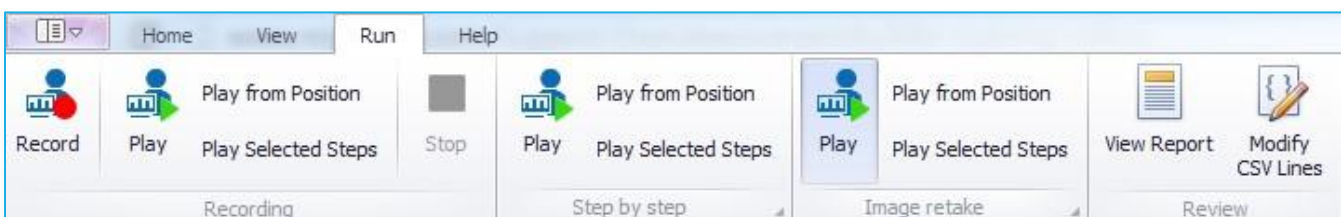
ScenarioBuilder will enter the absolute coordinates of the search area into the Image Action properties and select the “City” field based on these coordinates;

Image Search Area	
Start search area X	1080
Start search area Y	597
End search area X	1080
End search area Y	597

PLAY WITH IMAGE RETAKE

Use this feature to replace an image, or add multiple images to your image Actions *during scenario playback*. Recommended when scenario is being tested on a desktop other than that on which it was created. Differences in OS, color depth, resolution, browsers, etc. can be alleviated with this feature.

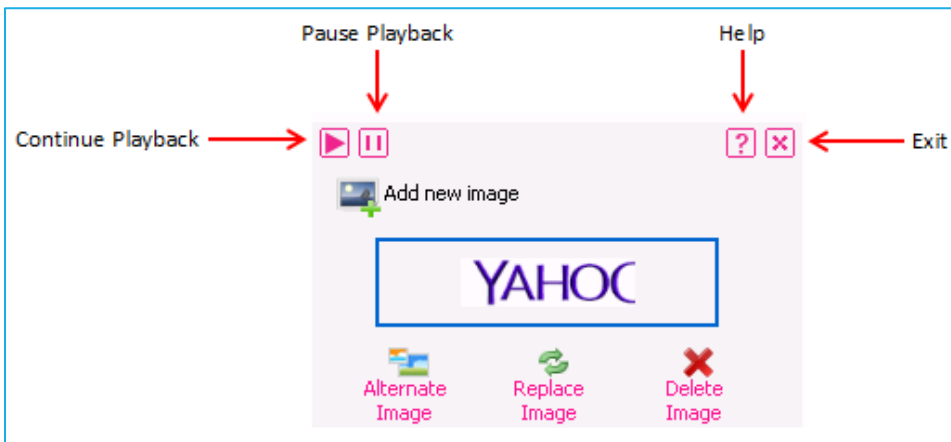
Open the “Run” menu, under the “Image Retake” section and click on “Play”  Play






ScenarioBuilder will begin playing back the scenario. When any image Action does not find its associated image, scenario playback pauses and a warning window pops-up:



The "Retake Image" dialog box pops up simultaneously in the lower right hand corner of the desktop;



From the pop-up dialog box, the user can perform the following operations:

- Click the image thumbnail to **replace the image** currently associated with the Action.
- Click "Add an Alternate Image"  to keep the current image, but **add an additional** image to the Action.
- Click "Continue"  on the pop-up menu to **resume scenario playback**. (Note: if you resume scenario playback without resolving the failed image Action, scenario playback will halt and report a failed execution).
- "Exit" the dialog box  to halt scenario playback and return to ScenarioBuilder.

 For more about Images actions, please check out "[Images Actions](#)" and "[Anchor Images](#)" in our [Video Tutorials](#) series.



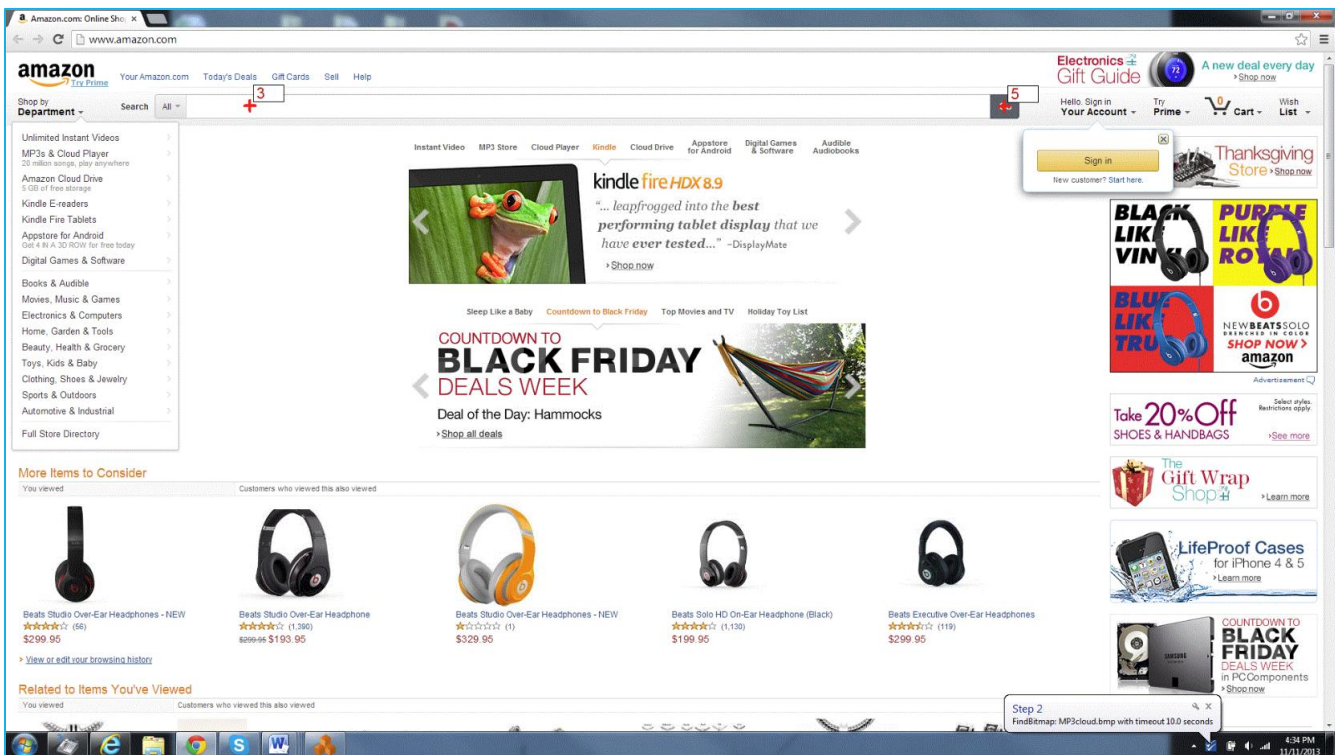
PLAYBACK SCREENSHOT

This feature captures a screenshot of the desktop during each step and displays the location of the mouse clicks in the image. If the mouse click actions are child actions, one screenshot at the parent step will contain all the child mouse click actions.

For example, a script that has a “Mouse Move and Click” child actions at the 3rd and 5th steps will appear like this in the Scenario Window,

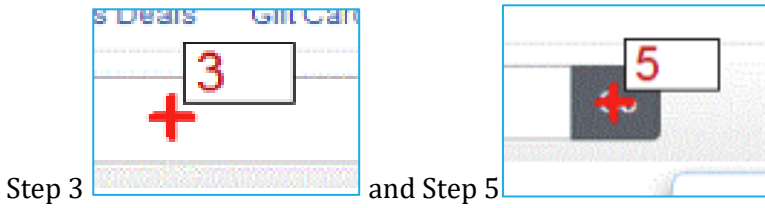
Action	Description	Image	Step
scenario	Check Out		
Open with File Associat...	http://www.amazon.com		1
Find Image	MP3cloud.bmp		2
Mouse Move and Click	Use Anchor, 282, -67, Left, Click		3
Type Text	Books		4
Mouse Move and Click	Use Anchor, 1372, -66, Left, Click		5

Double-clicking on the icon in the “Image” column will show the Playback Screenshot



Notice that the location and the step number of the mouse click actions is displayed on the screenshot:



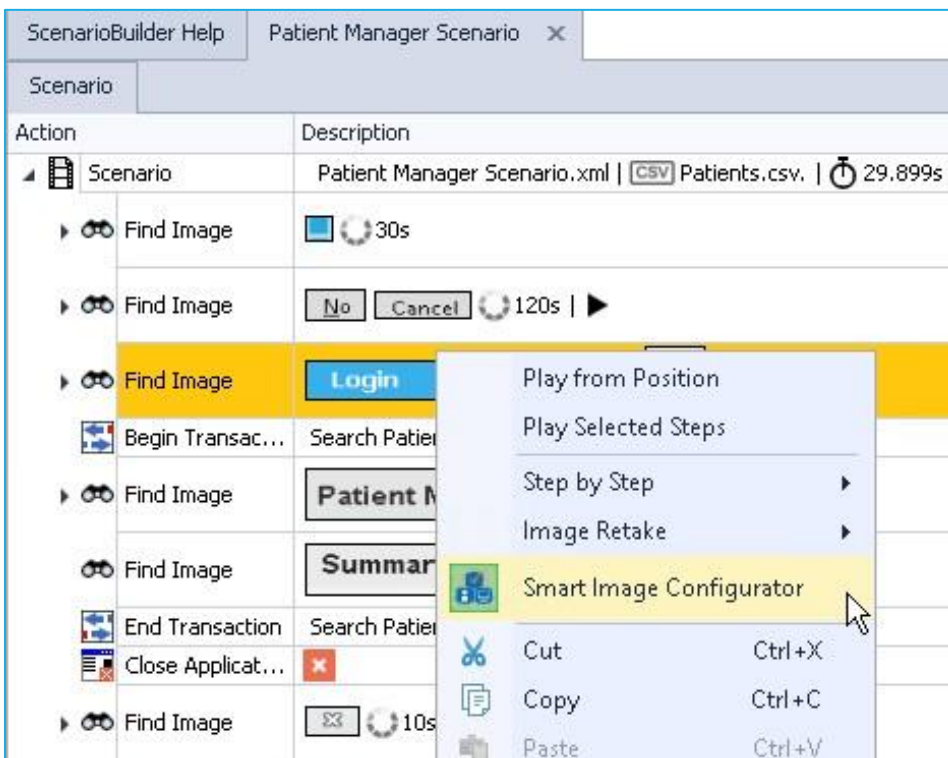


SMART CONFIGURATOR

The “Smart Configurator” is ScenarioBuilder’s built-in configurator for various images that are included in Test Scenarios. You can view, search and modify these images based on different failed screenshots. The screenshots that appear in the Smart Image Configurator are captured automatically by executing scenario(s) in ScenarioBuilder or Automai’s testing tools.







To open the configurator :

Right click the desired Image action that has failure screenshot(s) in the Scenario Window and choose Smart Configurator from the Action menu.

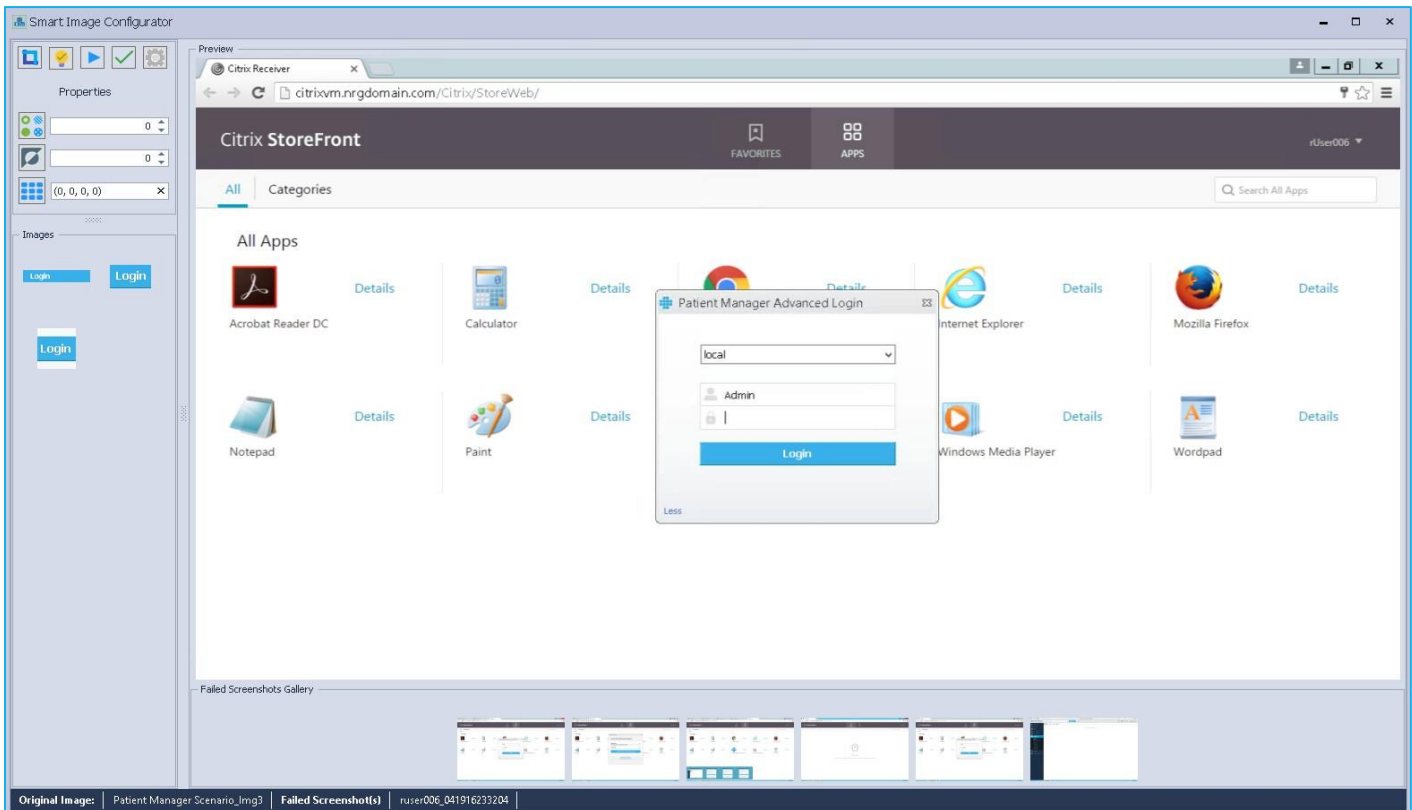


Or double click the desired Image action in the Scenario Window.








	Begin Transac...	Log On		
	Find Image	 30s 5%		
	Mouse Mo...	Anchored Drag (0,0) Left Click		
	End Transaction	Log On		


Here is a sample view of the Smart Image Configurator:





On the top left side of the window, you can see the Image Toolbar:

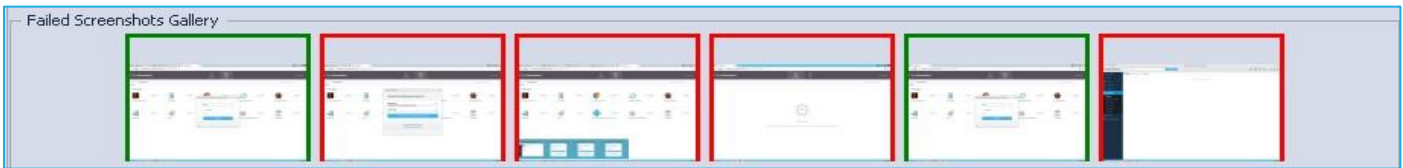
-  Opens the image capture tool to allow new images to be captured within the failure screenshot.
-  Suggests the best values for Tolerance and Image Area parameters if no image is found.
-  Specifies a tolerance percent for a successful image search.
-  Sets the amount of contrast to be applied to the image.
-  Clicking on this icon will allow an image area to be captured for recognition of the image action.





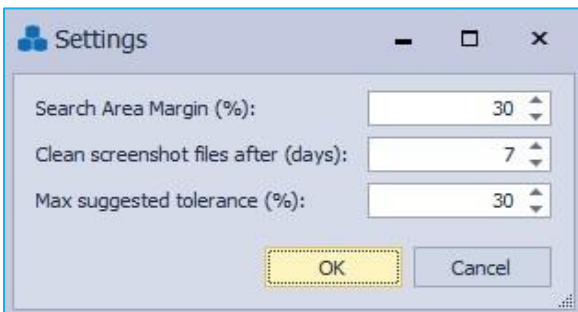
 Note that changing the search area and setting high tolerance may slow down your test execution.

-  Replays the image search action against the failure screenshot(s) using the captured image and based on the suggested recognition properties.

 A green rectangle (which shows the image is found in the screenshot) and a red rectangle (which shows the image is found in the screenshot) appear over the screenshot(s). Double-clicking on any of these screenshots, display the image and the location where the image was found as well as the image search area.



-  Applies changes to image action properties.
-  Allows modification of settings such as search area margin, maximum suggested tolerance, etc.



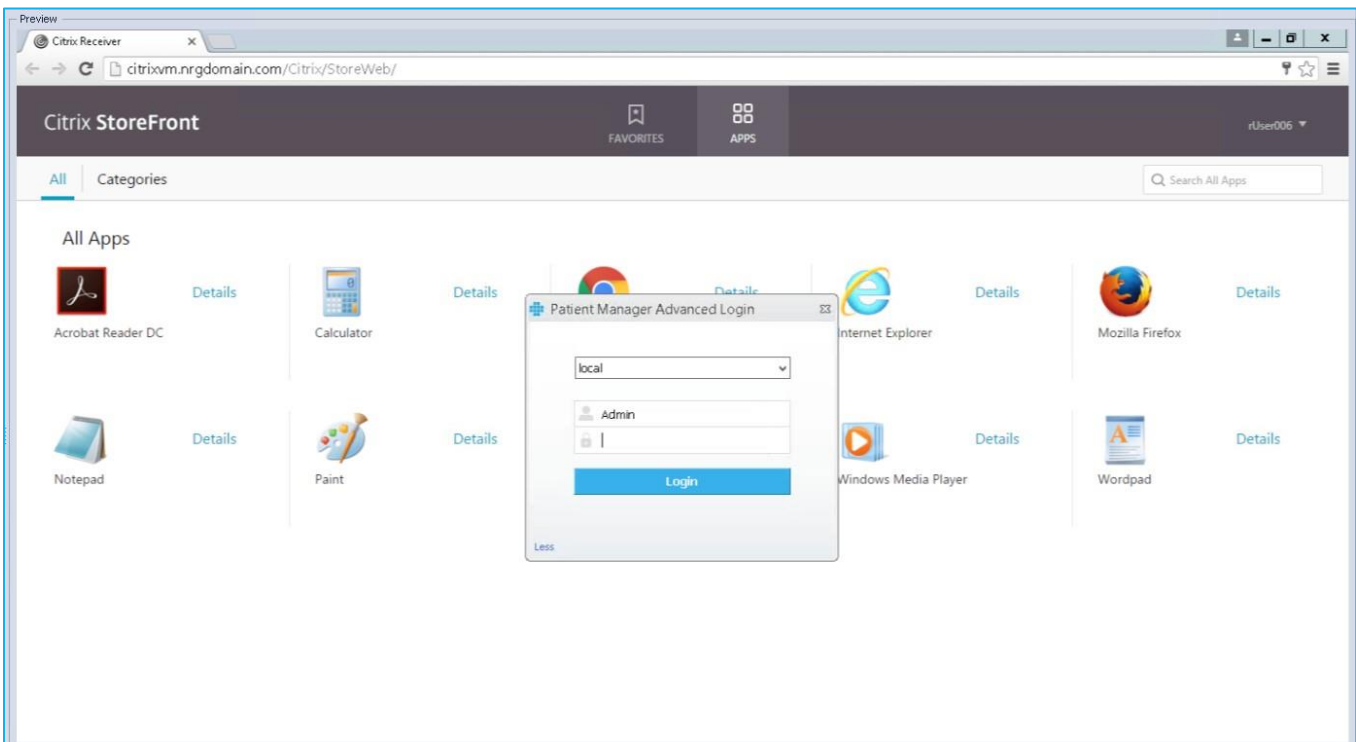
Each image action can have one or more images associated with it. These images are displayed in the "Images" panel.



You can crop, add or remove images by right clicking the image. The “Smart Image Configurator” only supports BMP format.




The contents of the Preview box depend on the screenshot that is currently selected in the “Failed Screenshots Gallery”.



The Failed Screenshots Gallery contains a collection of failed screenshots of the actual image action. This collection helps you create images from your tests.



 The failed screenshots of all images actions are stored in the “ImageStore” folder in *NRGGlobal* or *Automai* folder. This collection includes any failures from ScenarioBuilder playback, as well as the Injector playback (if using AppLoader) or the ScenarioStation (if using AppsWatch).

[Click here to watch a video demonstration of Smart Image Configurator](#)



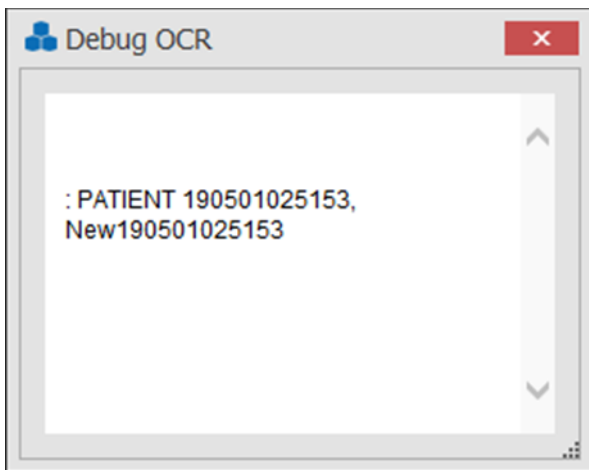
OCR Configurator

"Find Text in Image" and "Read Text Into a Variable" actions rely on optical character recognition (OCR) technology to recognize text on the screen. While this technology has improved a lot over the year, it may still need a little help in recognizing the text correctly.

If your OCR action is having issues finding the right text, the first thing to do is to check to see what text it is finding. To do this, click the Open OCR output text button that appears on the step.

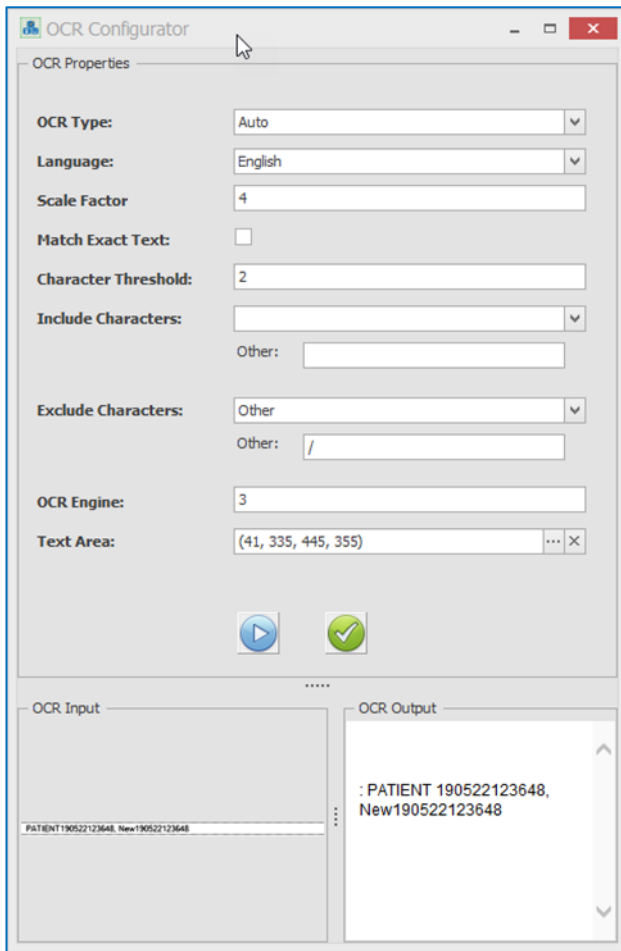


The Debug OCR window appears.



You can tell how successful your settings are by the correctness of the text in this window to the actual on screen text. If it isn't correct, you can adjust the properties using the Smart OCR Configurator. To open the Smart OCR Configurator, right-click on the step and select **Smart Configurator** from the menu.





At the bottom of the window you see the image it took of the text area. On the right is the text that it found. From here you can make configuration changes and try them out. Just be sure your application is open with the text you are trying to read displayed on the screen.

Here are the properties that can be changed in the OCR Configurator:

OCR type – Auto, Single block, Single Line, Single word. The default is Auto and that should work in most cases for a drop-down list with multiple values.

Language – A list of various languages currently available for use with the OCR technology. If you need to use a language other than English, it may require an additional download.

Scale Factor – The system will automatically calculate a scale factor to use for reading the text in the image. If it isn't finding text correctly, then you can override the calculated setting with a scale factor between 1 and 10. This defaults to 0 but can be set anywhere from 0-10. Where possible, the value should be between 1 and 6. If the text is small, use a scale factor of 1 to start.



Match Exact Text – Selecting this property indicates the search text must match all of the text found in the image. Note that this setting is only available for the Find Text in Image action.

Character Threshold – This is the number of characters that can be off and still make a match. By default, it is set to 0. This means the text to find and the OCR text found must match exactly. If the OCR action is having issues with a character or two you can adjust the number of characters that can be different and still make a match. For example, if you are looking for the text "ID 108764917" and it was having issues correctly recognizing the 9, you could set the Finding Threshold property to 1. This means the text it is looking for and the text it finds can be off by 1 character and still make a successful match. The higher the number, the more characters that can be different and still make a match. Care must be taken when using this property as setting the value too high could result in false positives. Note that this setting is only available for the Find Text in Image action.

Include Characters – Small Letters, Capital Letters, Numbers, Special Characters, Other. You can select the characters you want it to find. If you select Other, then enter those characters in the Other text box.

Exclude Characters – Small Letters, Capital Letters, Numbers, Special Characters, Other. These are the same choices as Include Characters. However, selecting items here will cause the OCR technology to ignore those characters instead of specifically look for them. For example, if the OCR is reading a slash and you want it to ignore that slash, add the slash (/) character to the Other field under Exclude Characters.

OCR Engine – By default, this is set to 3 which should work best in most cases. Other possible values are 0 (legacy engine only), 1 (neural nets LSTM engine only) and 2 (legacy and LSTM engines).

Text Area – This is the search area where it searches for the text. If the image in the lower left isn't the area it should be using to look for the text, click the ellipse button and select the correct area.

You can adjust the properties and then click the Play button to see the impact of the changes on recognizing the text.

If you altered the settings and it is reading the text correctly, click the Apply configurations to action button to save the changes and close the OCR Properties window.



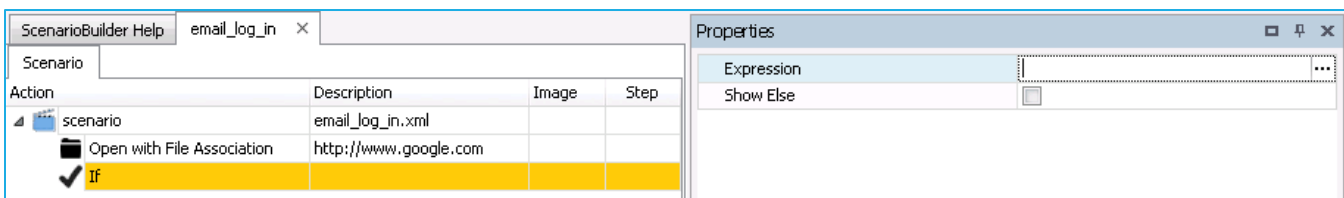
Conditional statements

ScenarioBuilder provides Actions that can evaluate conditions and execute user-defined reactions.

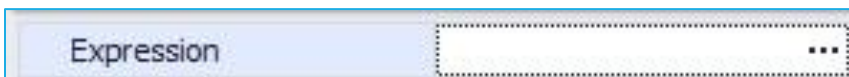
“If” action

Set up an IF statement to dictate a consequent series, or an alternate series of steps, depending on the status of its expression. When the conditions of the expression are met, the IF Action’s child steps will be executed.

Double-click or drag the “IF”  Action into your scenario;



In the properties pane, enter an expression to be evaluated.



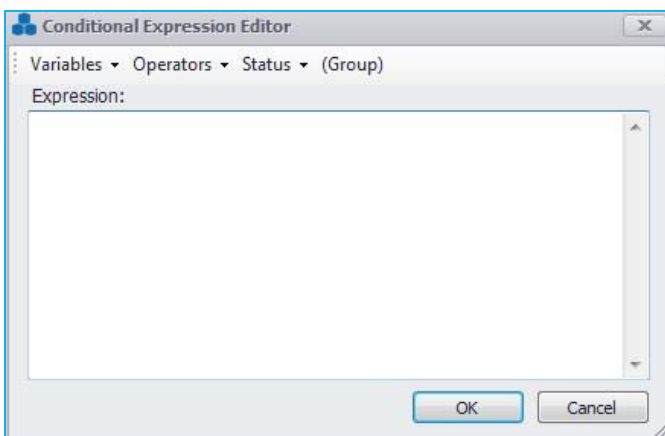
Here are some examples of expressions;

`%Variable%==1`

`%ReturnedValue%!=Error`

`%ExecMyImageStatus%==success`

Click the ellipses in the “Expression” properties field to open the “Conditional Expression Editor”;



From the editor, you can build your expression with quick access to:

- Variables – parameter values contained in a CSV file or MS Excel file attached to the scenario, pre-defined variables set by the user, execution status variables.
- Operators – greater than; less than; equal; not equal; AND; OR; greater or equal; less or equal.
- Status – select SUCCESS or FAILURE or EMPTY
- Group – highlight any section of your expression and click to encase in parenthesis

Show Else: Enable to add ELSE step below IF step, allowing an alternate series of steps to be added to the Action.

If ELSE is enabled, ELSE child actions will be executed when the conditions of the expression are not met.

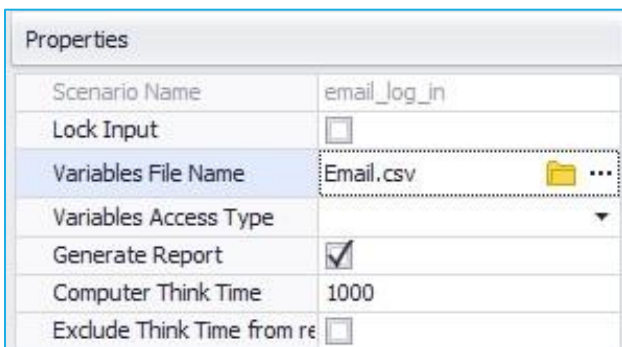
An “IF” Action lets you set up a fork in the scenario with two branches. The scenario evaluates a result or a value, and advances down the branch determined by that result or value. The following example evaluates a user account to see if it is a “superuser”, and advances.


“ IF “ ACTION WITH CSV FILE

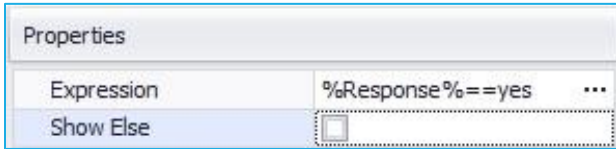
In this example, there is a CSV containing a list of (yes or no) values specifying whether the user account is a “Administrator”;

username	Administrator
jsmith	yes
gpear	no
lmoor	yes

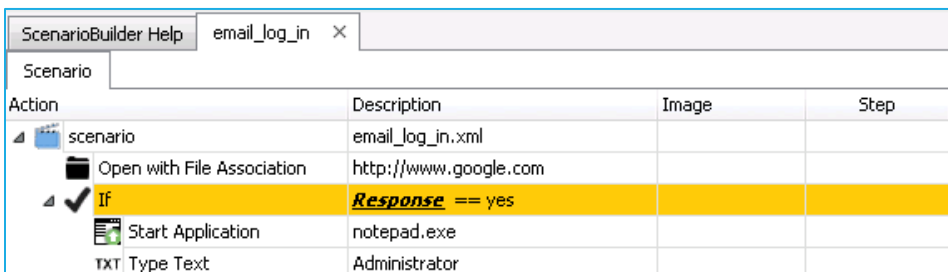
The CSV file is attached to the Scenario;



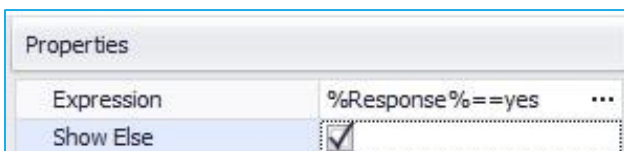
Double-click the  If Action or drag it into the scenario pane. The scenario will use the “Response” status as the condition that determines the path down which it advances. In the “If” Action’s Properties pane, in “Expression” field, type `%Response%==yes;`



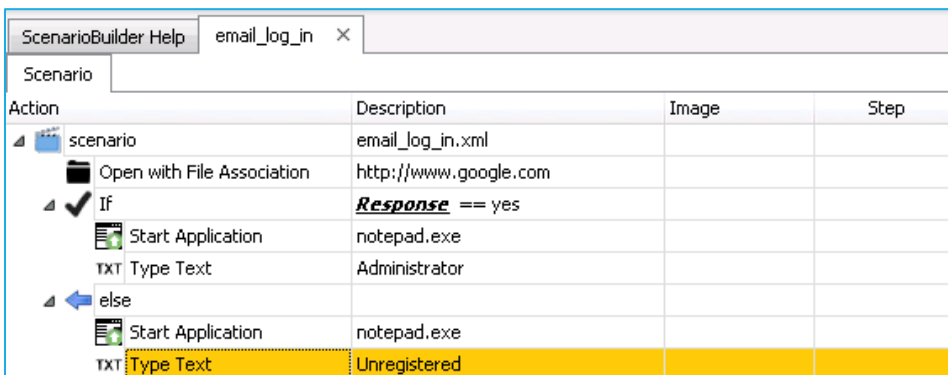
Then add actions to take if the criteria are met.



Click on the Show Else box to set actions to take if the criteria are not met.

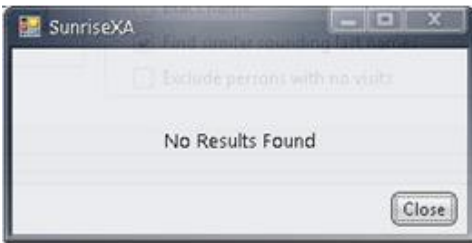


Add the actions to take if the criteria are not met as Children for “else”

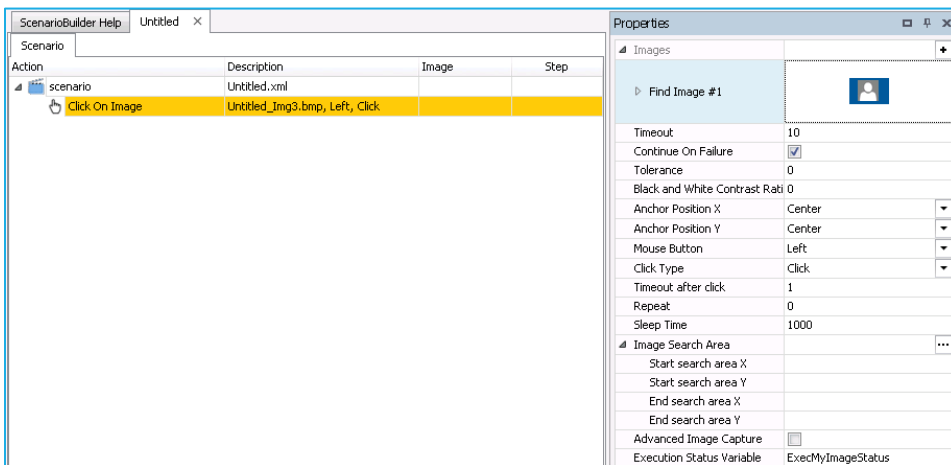



“ IF ” ACTION WITH EXECUTION STATUS VARIABLE

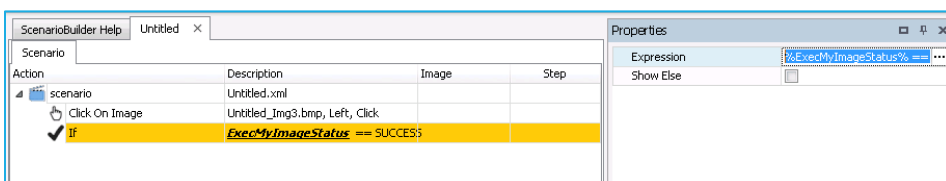
You may want to validate the success or failure of an Action and proceed according to the result. For example, you need to check for a result after executing the search function on a page, the search function might return a result or might return a popup window stating “no results found”. To test the result, take an image of the “no results found” pop-up window;



Enable the “Continue On Failure” property and note the name of the “Execution Status Variable” property (feel free to overwrite this value);



Double click on the  If Action or drag it to the scenario pane. In the “If” Action’s “Expression” property, enter a status expression for the “Execution Status Variable”, for example `%ExecMyImageStatus%==success`.



Now, drag and drop the appropriate Actions under the “If” Action’s “Success” and “Failure” branches.



A Component can be validated for success or failure in that same way as an Image Action. Enable the Component's "Continue On Failure" property and note the name of its "Execution Status Variable" property. Then create an "If" statement based on the success and failure of the Component.



For more about If/else actions, please check out "[Conditional Statements](#)" in our [Video Tutorials](#) series.


Case/Switch

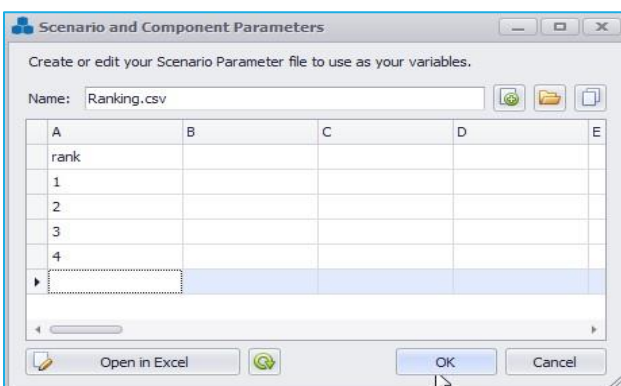
The Case and Switch Actions allow you to build complex scenarios that will respond in a variety of ways, depending on the result of a value. Where the "If" Action can evaluate a condition as being either true or false, and can provide a branch for either possibility, the Case/Switch Actions allow for virtually unlimited result possibilities, and can provide appropriate courses of action for all possibilities.

For example, a web form in which students rate their teacher has 4 possible responses. In order to test the form as real users would, you need a scenario that randomly selects one of the 4 responses;

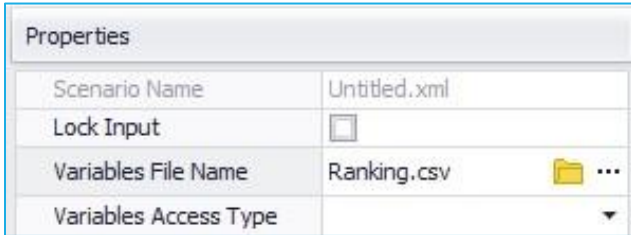
2. Please rate your overall satisfaction with the Professor/Course Instructor:

- Very Dissatisfied
- Dissatisfied
- Satisfied
- Very Satisfied

Start by creating a Scenario Parameters file in ScenarioBuilder. Click the "Scenario Parameters"  icon in the "View" section of the toolbar to open the "Scenario Parameters" window. Provide a name for the file as well as a column "rank" and the variables for it. Click "OK";

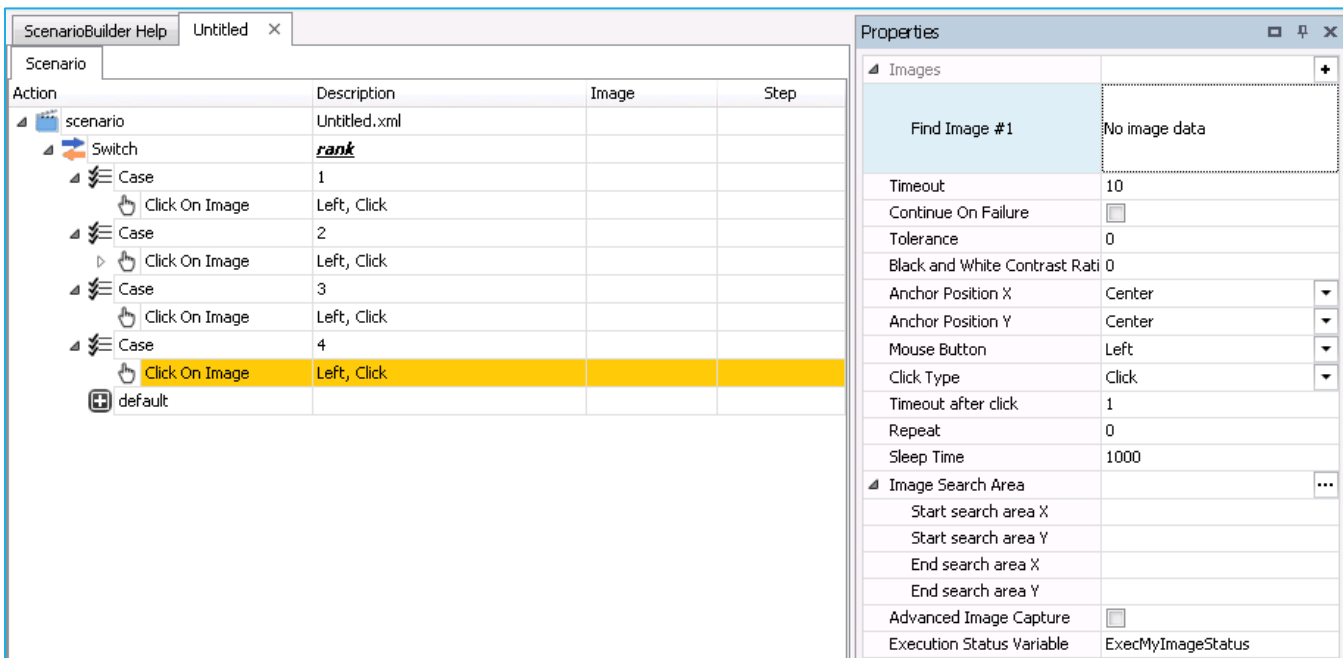


Save the file and attach it to the scenario. Set the “Variable Access Type” property to “Random”;




Add a “Switch” Action to the scenario and set the “Variable Name” property to “%rank%”. Be sure to include the percentage signs framing the variable name. This is the value that will determine the Cases and the subsequent Actions associated with each Case. Notice a “Case” child Action came into the Scenario Window with the “Switch” Action. Click on the “Case” and set the “Variable Value” property to 1. Add a “Click on Image” Action under the “Case” row and capture an image of the “Very Dissatisfied” choice from the web form.

Add 3 more “Case” Actions below the Switch, using values 2-4, and associate each with one of the choices on the form. Ignore the “default” row, which can be used as a “catch all” for values that don’t correspond to a Case in the scenario. Here is what the final scenario looks like;



The resulting scenario makes a random choice, clicking on one of the four evaluation rankings, depending on the value retrieved from the Scenario Parameters file.

 For more about Case/Switch actions, please check out “[Conditional Statements](#)” in our [Video Tutorials](#) series.

Working with scenario parameters

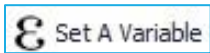
With Scenario Parameters, data input can be varied to emulate real world application usage. Scenario Parameters can be:

- Pre-defined values that follow specific formats, such as a date, time, or number.
- Execution Status Variables determined by the success or failure of an Action.
- User-defined sets of values contained in CSV files, such as names, addresses, birth dates, etc., which can be accessed sequentially or randomly by your scenario.

Adding pre-defined variables to a scenario

SET A VARIABLE

The “Set A Variable” Action offers a selection of pre-defined formats from which to choose. Double-click “Set A Variable” in the Actions pane;



Highlight the Action in the Scenario Window and its associated properties can be viewed;

Properties	
Variable Name	<input type="text"/>
Variable Value	<input type="text"/>
Encrypted	<input type="checkbox"/>

Variable Name: Provide a meaningful name to the variable (e.g. *loginnames* or *date*). Variable Names are case sensitive and should not include any special characters or spaces.

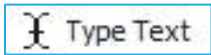
Variable Value: Select one of the pre-defined formats from the drop-down list. Options include random numbers, dates, times, and various environmental system values. For date and time pre-defined variable values, the default values are always based on current date and time. However, values can be manipulated by adding integers to, or subtracting integers from individual month, day, year, hour, minute or second parameters (e.g. *one month from today: [M+1/d/yy]*; *yesterday: [MM/dd-1/yy]*; *one hour ago: [h-1:mm:ss tt]*). Only one modification (addition or subtraction) is allowed per variable value (e.g. *[M+1/d+1/yy]* is not accepted). Do not remove brackets from the pre-defined values.



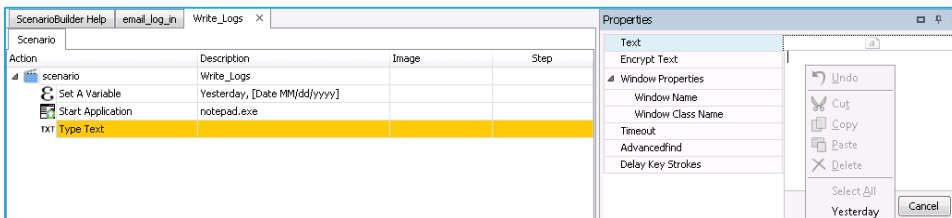
Encrypted: Check this box if you wish to encrypt the text for security purposes. This is useful if your variables contain passwords or other sensitive information.

Call variables into the scenario in the following manner:

From the “Keyboard” section in the Actions pane, double-click “Type Text”;

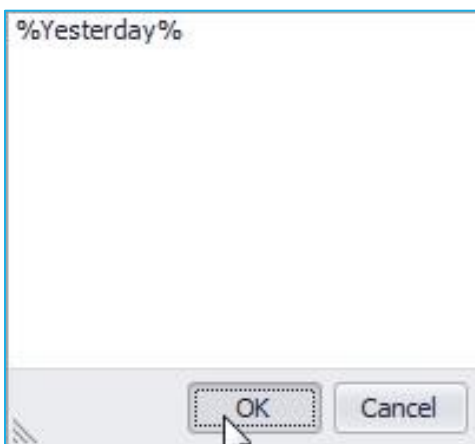


With the “Type Text” Action highlighted in Scenario Window, right click in the text field;

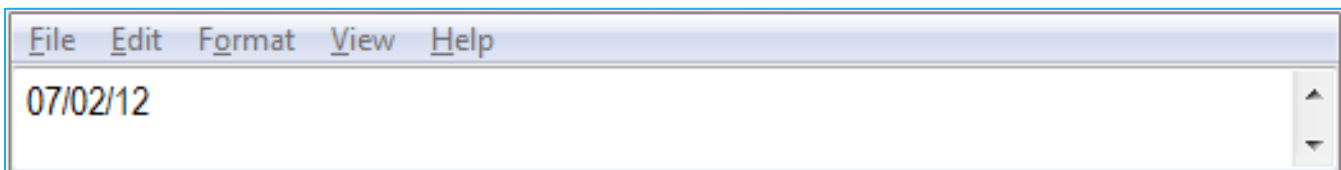


Select your variable from the bottom of the menu (or simply type the name of your variable, framed with percentage (%) signs, into the text field).

Click OK to complete your “Type Text” Action;



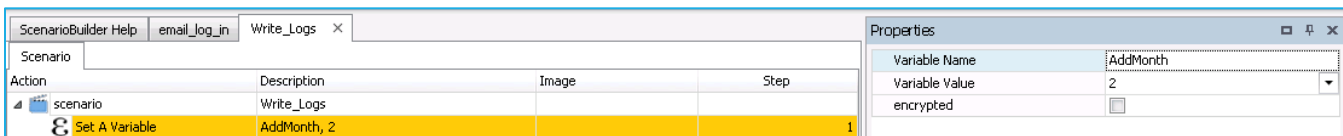
The above example opens Notepad and types yesterday’s date;



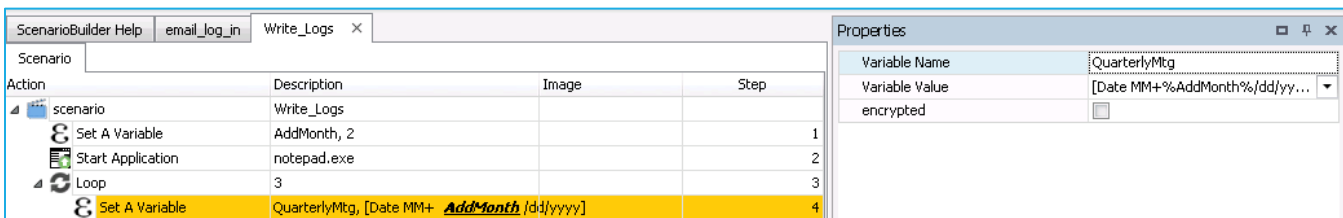
INCREMENT A VARIABLE

ScenarioBuilder can increment a variable so that each time through a loop, the scenario increases a value by a specified amount. The following example demonstrates how to increment a date by three months at a time. To add to the challenge, the first date in the sequence has to be two months from the current date.

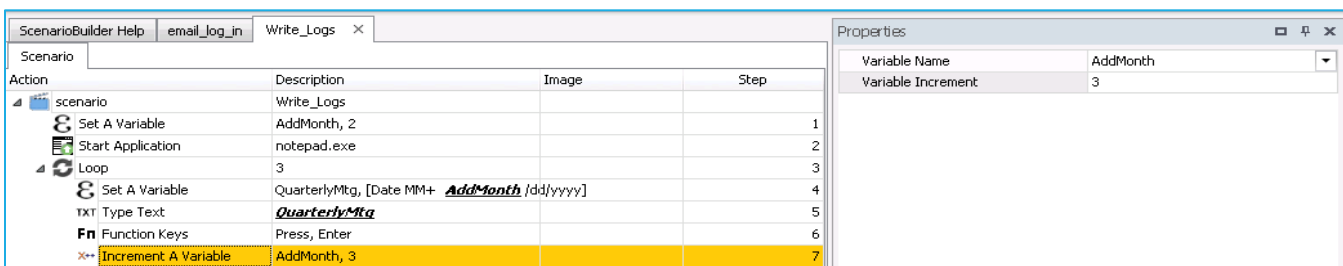
Start by setting a variable that will provide the value by which *another* variable will be incremented. Set the “Variable Value” property to “2”;



Next, the scenario opens Notepad. Then, beneath a “Loop” Action, another “Set A Variable” Action establishes a “QuarterlyMtg” date. Notice the modification which increases the “QuarterlyMtg” by the value of the “AddMonth” variable;



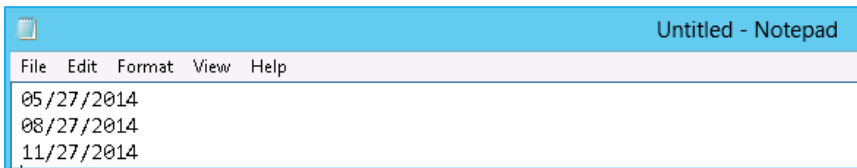
Lastly, the scenario uses “Increment A Variable” to establish the amount by which to increment the “QuarterlyMtg” date. The full scenario looks like this;



The first pass through the loop will type a date two months from today into Notepad (the “AddMonth” “Variable Value” property establishes the initial value at “2”). Before jumping to the top of the loop, the “Increment A Variable” Action operates on the “AddMonth” variable by adding “3” to it (established by the “Variable Increment” property field). Now, next time through the loop, the “AddMonth” value is “5” resulting in the scenario typing a



date five months from today. And finally, the last pass through the loop enters a date eight months from today into Notepad.



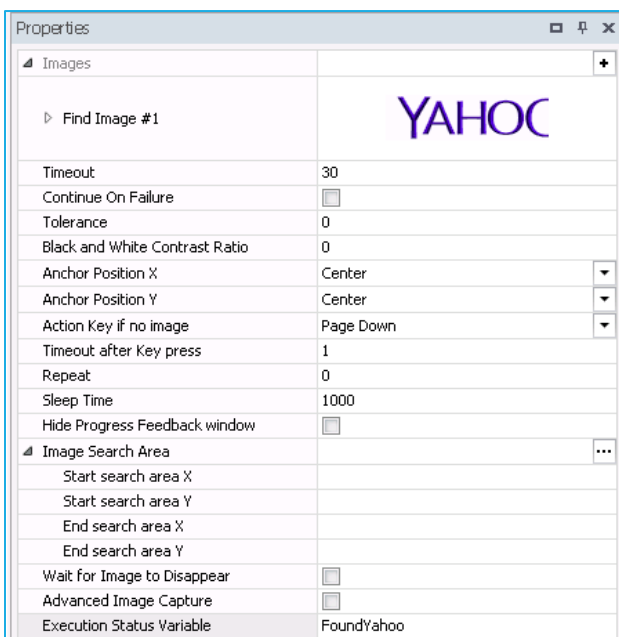
Execution status variables

With an “Execution Status Variable” you can direct a scenario down different paths, depending on the success or failure of an Image or a Window Action. Execution Status Variables are established in the Image or Window Action properties pane and are named by default (the default value can be kept or overwritten);



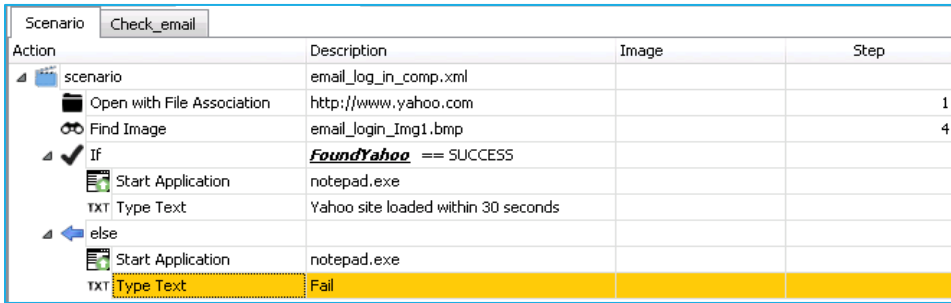
An Execution Status Variable’s result (**success**; **failure**) can be used in conditional statements with a comparative operator (**== (equal)**; **!= (not equal)**) to direct scenario behavior.

The following example uses a “Find Image” Action with a 5 second “Timeout” to look for the “Yahoo” logo after launching www.yahoo.com. The “Find Image” Action has an Execution Status variable called “FoundYahoo”;



The scenario uses an “If” Action that, when the site launches and displays the logo within 30 seconds, opens Notepad and types “Yahoo site loaded within 30 seconds”. Or, if the site fails to load in a timely fashion, opens Notepad and types “Fail” (Note the syntax in the example: *%FoundYahoo%==success*).

The scenario looks like this;



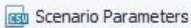
Action	Description	Image	Step
scenario	email_log_in_comp.xml		
Open with File Association	http://www.yahoo.com		1
Find Image	email_login_img1.bmp		4
If	FoundYahoo == SUCCESS		
Start Application	notepad.exe		
TXT Type Text	Yahoo site loaded within 30 seconds		
else			
Start Application	notepad.exe		
TXT Type Text	Fail		

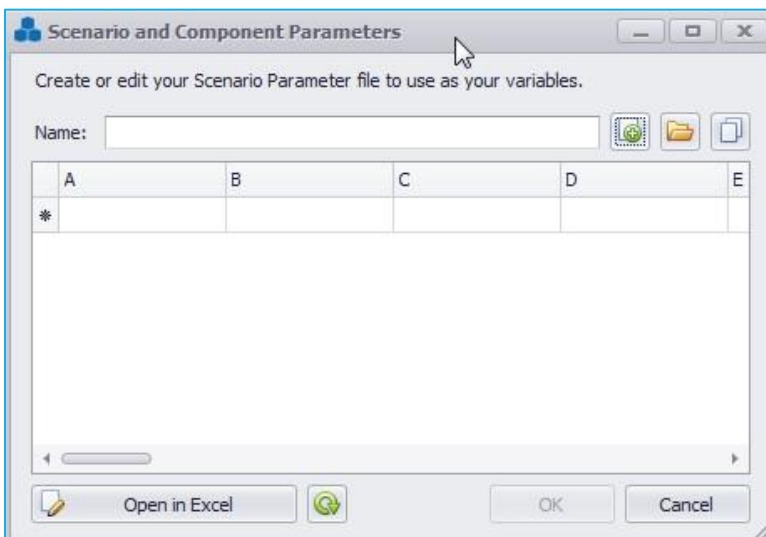
Using a CSV file to store values for scenario parameters


To genuinely replicate real world impact on an application, parameterizing data is critical. In ScenarioBuilder you can access unlimited rows of data values, stored in comma separated values (CSV) files, to make each playback of a scenario a unique, real user experience.

CREATE A NEW SCENARIO PARAMETER CSV FILE

Start by creating a file that contains the values to be used for Scenario Parameters. Click the “Scenario Parameter”

 icon in the ScenarioBuilder tool bar. The Scenario Parameters window opens;



Provide a name for the file and click the “Open in Excel”  button in the Scenario Parameters window;

Microsoft Excel™ or other default CSV editor will open in a new window;

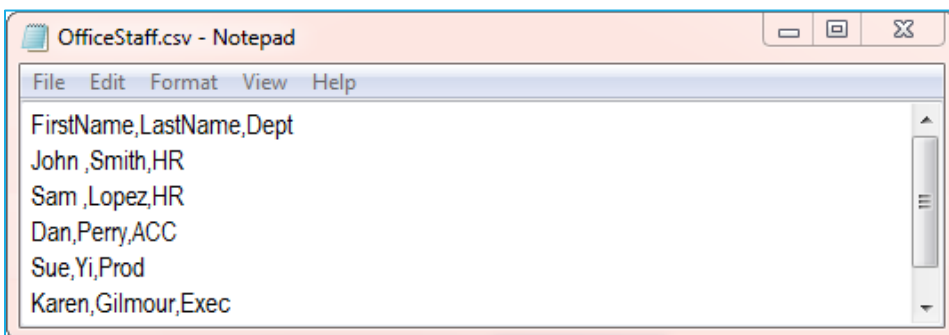
	A	B	C	D	E	F
1	ColName1	ColName2				
2						
3						
4						
5						

ScenarioBuilder gets you started by providing two column headings. Feel free to use the generic headings or overwrite them. Enter data values for the rows below the headings. Each column represents a different value to be used in a scenario. Below is a sample file containing office staff;

	A	B	C
1	FirstName	LastName	Dept
2	John	Smith	HR
3	Sam	Lopez	HR
4	Dan	Perry	ACC
5	Sue	Yi	Prod
6	Karen	Gilmour	Exec

If using Excel, as in the above example, simply “Save” the file and it will be stored in CSV format in the current Project’s “Variables” folder. Exit Excel.

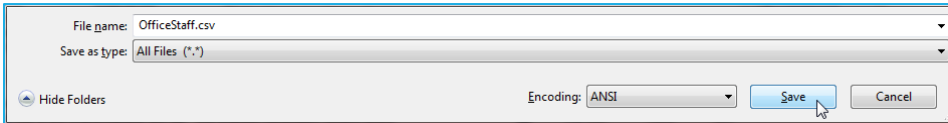
If using a text editor like Notepad, include commas between headers and values (no spaces);




Save the file and it will be stored in CSV format in the current Project’s “Variables” folder. Close text editor.

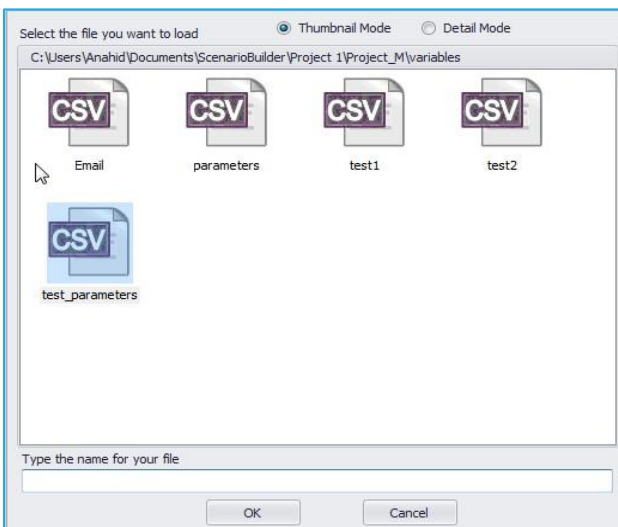


Tip If the editor does not automatically save the file in CSV format, use the editor’s “Save As” option and change the “Save as type” to “All Files (*.*)”; add the .csv extension to the file name and “Save”;



EDIT AN EXISTING SCENARIO PARAMETER CSV FILE

Click the folder  icon in the Scenario Parameters window to open the current Project’s “Variables” folder;

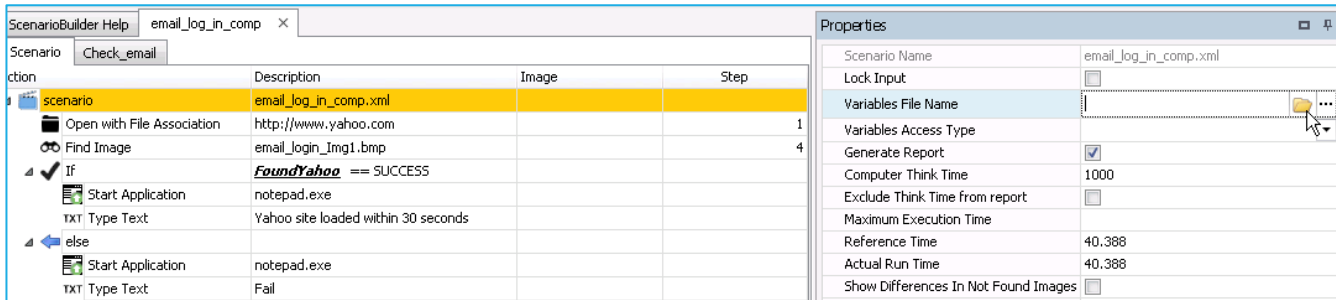


Double-click the desired file and it will open in your default CSV editor.

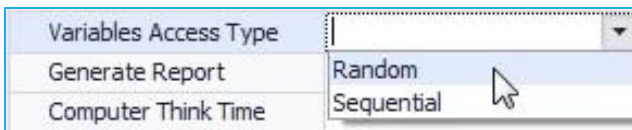
ATTACH A SCENARIO PARAMETER CSV FILE

Scenario Parameter CSV files can be attached at the scenario level for global access, or at the Component level for access limited to the Component. Unless your intention is to have the scenario loop through a Component multiple times, retrieving a new row from the CSV file with each loop, attach the file at the scenario level. To do so, highlight the scenario name (top row) in the Scenario Window, and click the “Variables File Name” property. Enter the name of the desired file, or navigate to it by clicking the “Open” icon;




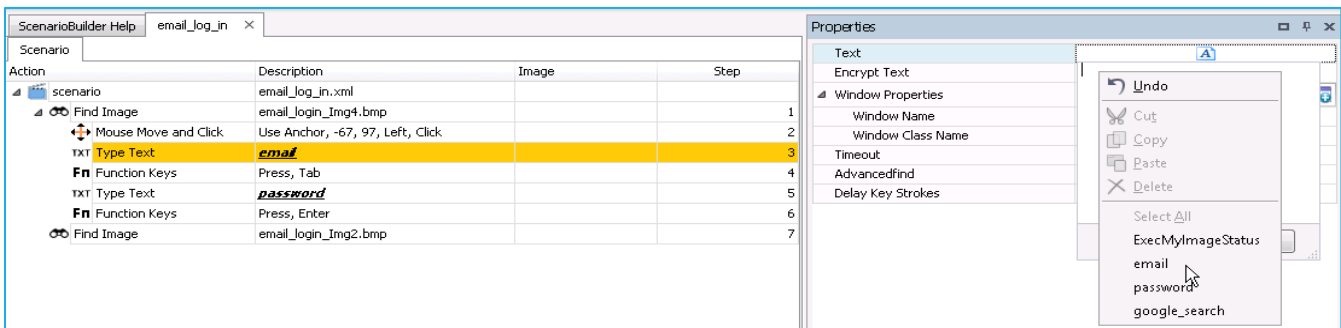


In the “Variable Access Type” field select “Random” or “Sequential” to determine how values are retrieved from the CSV file;



ADD SCENARIO PARAMETERS

Now that you have created a CSV file and attached it to the Scenario, you are ready to add the Scenario Parameters. Add a “Type Text”  Type Text Action to the scenario. In the “Properties” pane, right click inside the “Text” field;



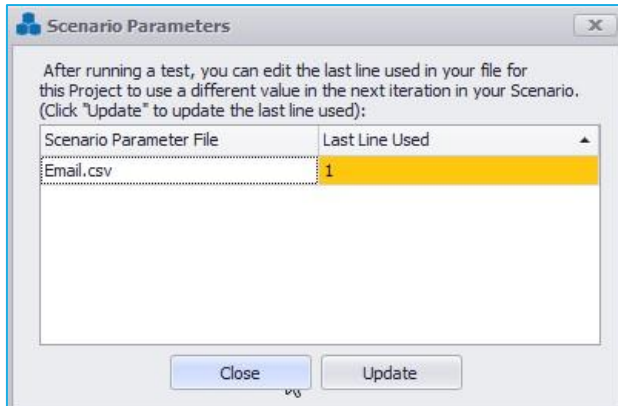
At the bottom of the pop-up menu, click the CSV file to reveal all of the fields (column headers) from the attached CSV file (file names are framed in percentage (%) signs). Select one of the fields. Typing the name of the field into the Text property field (framed with percentage signs) is also an acceptable method of adding the field.



LAST LINE USED IN SCENARIO PARAMETER CSV FILE



After executing a playback of the Scenario, click the “Modify CSV lines” under the “Run” section of the tool bar to display the “Last Line Used” from the attached CSV file;



If accessing the CSV file sequentially, edit the “Last Line Used” value to stage the scenario to retrieve a desired value from the file on the next execution (if accessing randomly, adjusting the “Last Line Used” value will not impact the next playback).

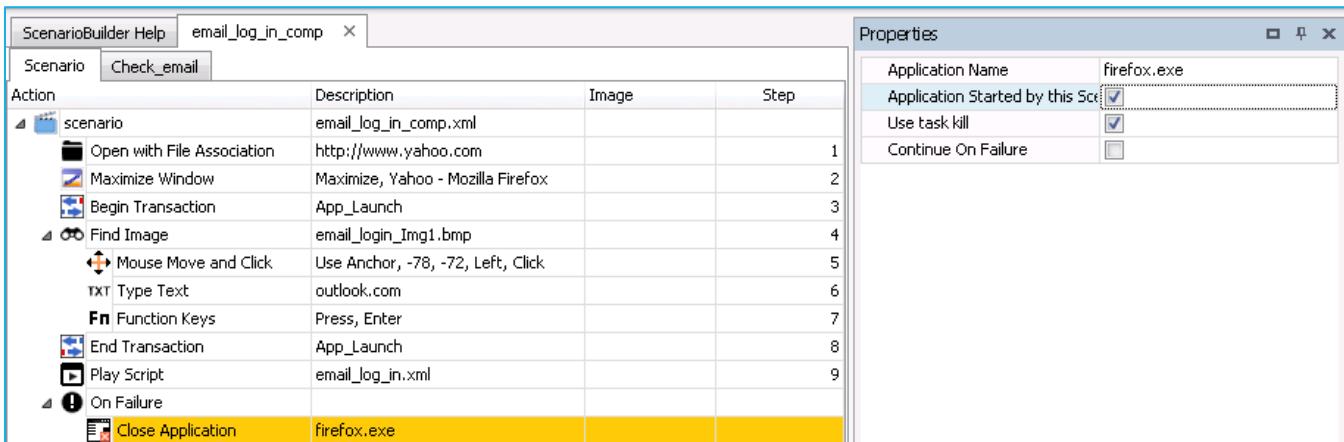


For more about Variables actions, please check out “[Use Variables Actions](#)” in our [Video Tutorials](#) series.

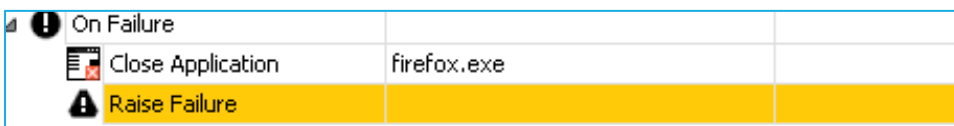


On Failure

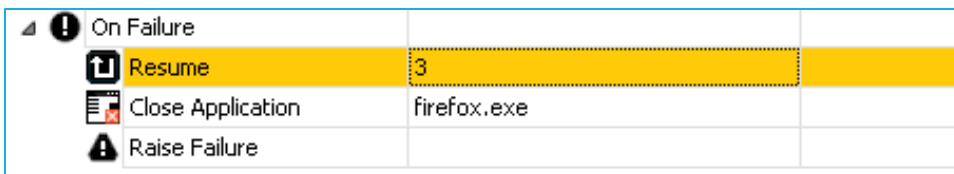
The “On Failure” section of a scenario determines what Action(s) to take when a Scenario or Component fails. Add the “On Failure” Action to the end of the scenario and drag the appropriate Actions underneath it. In the event of a failure, the scenario will skip to the “On Failure” section and execute the Actions. “On Failure” can be as simple as closing the application, thereby resetting the desktop for another iteration of the scenario (see below), or as complicated as undoing multiple changes made by the scenario up to the failed step.



When a scenario resorts to the “On Failure” steps, it does not report the execution as a failure, *unless* the “Raise Failure” property is enabled or a “Raise Failure” Action is included in the “On Failure” section.



Add a “Resume” Action to the “On Failure” section to retry the scenario from the point of the failure;



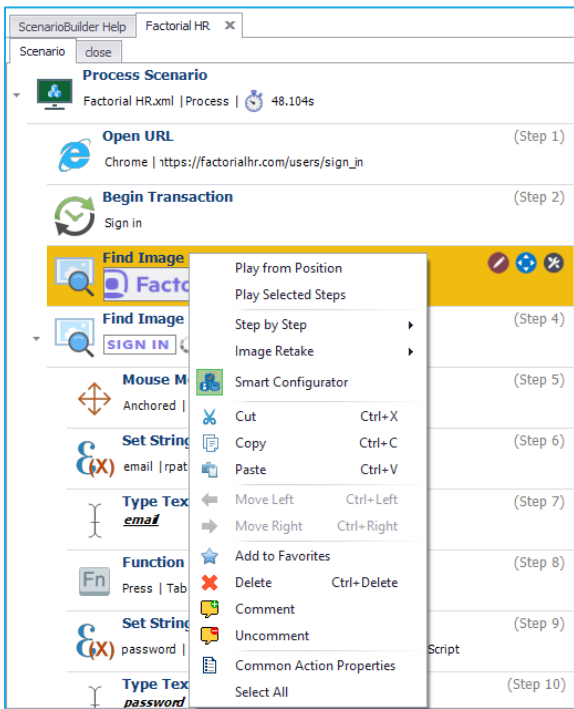
Set the “Retries” value to establish how many times to attempt the “Resume”. After the retries are exhausted, the scenario will advance through the subsequent steps of the “On Failure” section.




Tip For more about On Failure actions, please check out “[Using On Failure Action](#)” in our [Video Tutorials](#) series.







Action context menu

Right click on any Action you have added to the Scenario Window, from either the “Tree” or “Graphical” view to see a menu of options;






<u>Icon</u>	<u>Name</u>	<u>Description</u>
	Play from Position	Play the scenario starting at the highlighted step.
	Play Selected Step(s)	Play the highlighted step(s) only.
	Step by Step	Play the scenario with “Step by Step” playback



 Play	Image Retake	Play the scenario with “Image Retake” playback
	Smart Configurator	View, search and modify scenarios images based on different failed screenshots
 Cut	Cut Selection	Cut (copy to clipboard, and remove) the highlighted step(s).
 Copy	Copy Selection to Clipboard	Copy the highlighted step(s) to the clipboard.
 Paste	Paste Selection from Clipboard	Paste the latest “copied” item from the clipboard to the scenario.
 Move Left	Move Selection to the Left	Move a child Action left, toward the parent level.
 Move Right	Move Selection to the Right	Move an Action under the preceding Action, making it a child of the preceding. The child Action will only execute if the parent Action is successful.
 Add to Favorites	Add to Favorites	Add an Action to the Favorites folder to be called later in the current scenario or in a future scenario (from the same Project). Actions saved to Favorites include their child Actions, and they retain their Property settings.
 Delete	Delete Selection	Delete the highlighted step(s).




 Comment	Comment out	<p>To bypass step(s) without removing from the Scenario Window, they can be “commented out”. Steps that have been “commented out” appear in italics. To hide steps that have been “commented out”, click “Show/Hide Comments” option in “View” menu.</p>
 Uncomment	Uncomment	<p>To re-activate “commented out” step(s), right-click on the step(s) and select “Uncomment”.</p>
 Common Action Properties	Common Action Properties	<p>Selects (highlights) all similar Actions in scenario and applies changes made to Properties across all selected Actions.</p>
Select All	Select All	Select all Actions in scenario.



Scenario properties

Scenario “Properties” establish the playback settings for each scenario. To edit the scenario properties, click the top row in the Scenario Window. The scenario properties display in the “Properties” pane;

Properties <input checked="" type="checkbox"/> Pin This	
Scenario Name	Factorial HR.xml
Scenario Type	Process
Tag(s)	
Lock Input	<input type="checkbox"/>
Variables File Name (CSV)	
Variables Access Type	
Generate Report	<input checked="" type="checkbox"/>
Computer Think Time	1000
Exclude Think Time from report	<input type="checkbox"/>
Maximum Execution Time	
Reference Time	48.104
Actual Run Time	48.104
Default Image Tolerance	0
Show Differences In Not Found Images	<input type="checkbox"/>
Hide Balloon	<input type="checkbox"/>
Failure Message	%_scenario% Failed at the step number %_step% %_action% %_warning% iteration: %_iteration%
Message Font Size	15
Project	Project_Test1
Created By	ruser001
Creation Date	10/15/2018 11:04:21 AM
Last Modified By	robot001
Last Modified Date	11/27/2018 02:47:31 PM
Scenario ID	1539601461586
ScenarioBuilder Version	8.3.9.2
Description	
Do Not Close Opened Processes	<input checked="" type="checkbox"/>
Dataset Name	F_hr
Variables File Name (Excel)	C:\Users\robot001.WIN-ASOQ7GI...  ...
Sheet Name	
Headers Start Row	1
Headers End Row	4
Headers Delimiter	:
Scenario Section	

Scenario Name: The name of the scenario.

Scenario Type: The type of scenario.

- **Process Scenario:** Perform a business process. e.g. create invoice.
- **Global Scenario:** Perform actions that are the same across a project.
- **Flow Scenario:** Master script that call other process scenarios to simulate a business flow. e.g. create customer, create invoice, receive payment.

Lock Input: If this field is checked, the keyboard and mouse will be locked during the execution of the scenario.



Variables File Name (CSV): If using Scenario Parameters from a CSV file, attach the CSV file here. Click the folder icon to browse for the desired CSV file. Refer to the preceding section, [Scenario Parameters](#), for more information about working with variables.

Variables Access Type (optional): Determines whether the variable fields in the CSV file will be played back sequentially or randomly.

Generate Report: Check this field if you would like a report generated for the playback.

Computer Think Time: Specify the time interval between each step of execution. Values are in milliseconds; the default is 1000 milliseconds.

Exclude Think Time from report: If Checked, the “Computer Think Time” set above will be excluded from the reported execution time.

Maximum Execution Time: Set the time allowed for the scenario to complete its playback. This parameter is useful to determine if applications are responding in a timely manner. If the playback does not complete within the time specified, the scenario will fail and a screenshot will be taken of the failed scenario step.

Reference Time: This field is used as the standard to which subsequent iterations of the scenario are compared.

Actual Run Time: The complete duration of the scenario from start to end. “Actual Run Time” does not factor out “Think Time” or response time of steps suspended by “Timer” Actions (as compared to “Reference Time” which does exclude these values).

Default Image Tolerance: If value is greater than 0, then the find image actions will execute a final try to find the image with this tolerance value. (The Tolerance is maximum percentage of differences allowed for a successful image search. By default the value is 20.

Warning: A tolerance higher than 50% might cause false positives.

Show Differences in Not Found Image: Check this field if you would like to see the differences between the not found images.

Hide Balloon: If checked, the step-by-step scenario status balloon will not display for each rUser session in AppLoader, and for each ScenarioStation in AppsWatch.

Failure Message: This is the message generated when the scenario fails. You can customize the message by using the following variables:



%_error%: default message

%_scenario%: Failed scenario Name

%_component%: Failed Component Name

%_step%: Failed Step Number

%_action%: Failed Action

%_type%: Failure Type (test or system)

Message font size: Set the failure message font size (Use a smaller font size for long messages).

The remaining fields (Created By, Creation Date, Last Modified By, Last Modified Date, ScenarioBuilder Version) are read-only and show the general properties of your scenario file. This information is automatically populated by the system.

Description: Put a description about scenario.

Do Not Close Opened Processes: If checked, the scenario will not close all of the processes that it opened.

Dataset name: The Microsoft Excel option should only be used for Robotic Process Automation (RPA). Give a short name that will be used to identify the worksheet (e.g. Emp)

Variables File Name (Excel): Select the file name and location. Create a file using Microsoft Excel if it does not exist already. (e.g. Employees.xlsx)

Sheet Name: The worksheet to point to. Leave blank if using the default one.

Headers Start Row: The line where the headers are located. Default value is one.

Headers End Row: If the document contains multiline headers indicate the last line number.

Headers Delimiter: Headers separator. e.g. Emp:office:phone

Scenario Sections: This is for Global scenarios only. Specify the section name that can be called from other scenarios.

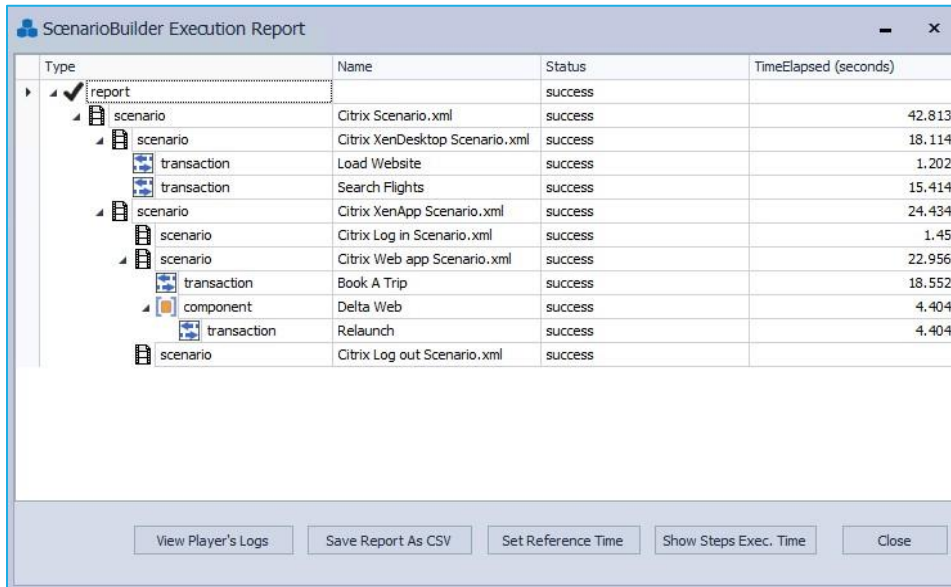


Scenario playback

Execution report

After the scenario playback, ScenarioBuilder will generate an execution report.

In case of success:



Type	Name	Status	TimeElapsed (seconds)
report		success	
scenario	Citrix Scenario.xml	success	42.813
scenario	Citrix XenDesktop Scenario.xml	success	18.114
transaction	Load Website	success	1.202
transaction	Search Flights	success	15.414
scenario	Citrix XenApp Scenario.xml	success	24.434
scenario	Citrix Log in Scenario.xml	success	1.45
scenario	Citrix Web app Scenario.xml	success	22.956
transaction	Book A Trip	success	18.552
component	Delta Web	success	4.404
transaction	Relaunch	success	4.404
scenario	Citrix Log out Scenario.xml	success	

Buttons at the bottom: View Player's Logs, Save Report As CSV, Set Reference Time, Show Steps Exec. Time, Close

The report will display the scenario as well as any components, transactions or sub-scenarios associated with it.

It will also display the execution time for the scenario as a whole as well as its transactions, components and sub-scenarios. The first time the Scenario is played successfully, the execution time will automatically be set as the Reference Time.

For subsequent playbacks the Reference Time will have to be set by clicking on “**Set Reference Time**”. The Reference Time will then be used to compare all the other iterations during the load test. The difference will be used to calculate the “Delay Time” and “Delay Percentage” in the Load Test Reports.


View Player's Logs can be used to view Player's most recent logs.

Save Report As CSV can be used to save ScenarioBuilder Execution Report as a CSV file.

Show Steps Exec. Time can be used to show the execution time for each step of the scenario.

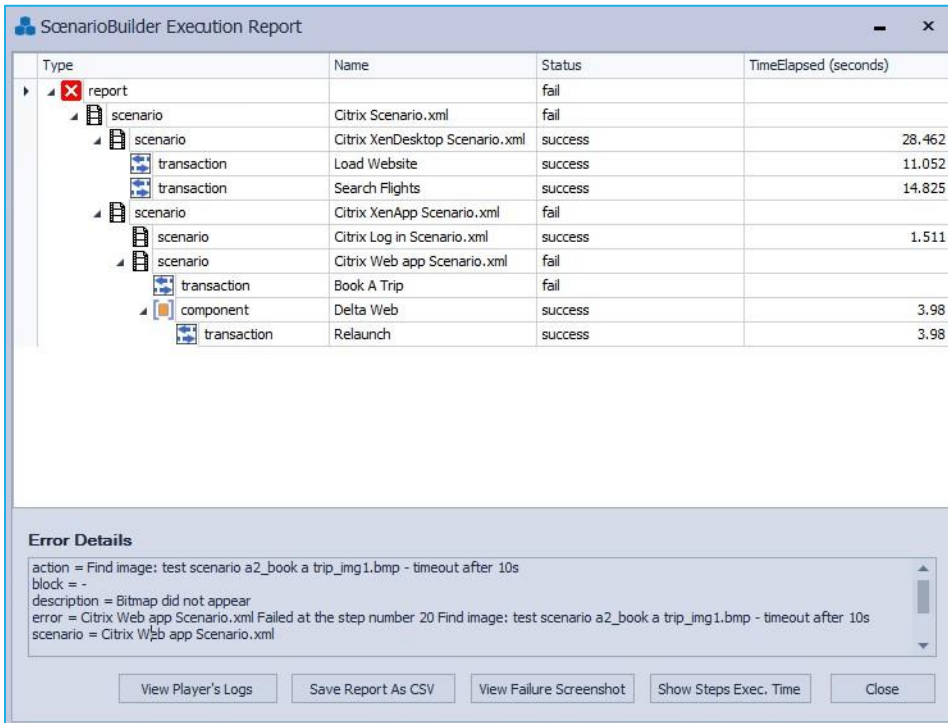
View Failure Screenshot can be used to show the failure screenshot.



 You can also view failure screenshot with higher quality in BMP format by adding the following line in *awplayer.ini* file in the ScenarioBuilder folder.

screenshottype=bmp

In case of failure:



Type	Name	Status	TimeElapsed (seconds)
report		fail	
scenario	Citrix Scenario.xml	fail	
scenario	Citrix XenDesktop Scenario.xml	success	28.462
transaction	Load Website	success	11.052
transaction	Search Flights	success	14.825
scenario	Citrix XenApp Scenario.xml	fail	
scenario	Citrix Log in Scenario.xml	success	1.511
scenario	Citrix Web app Scenario.xml	fail	
transaction	Book A Trip	fail	
component	Delta Web	success	3.98
transaction	Relaunch	success	3.98

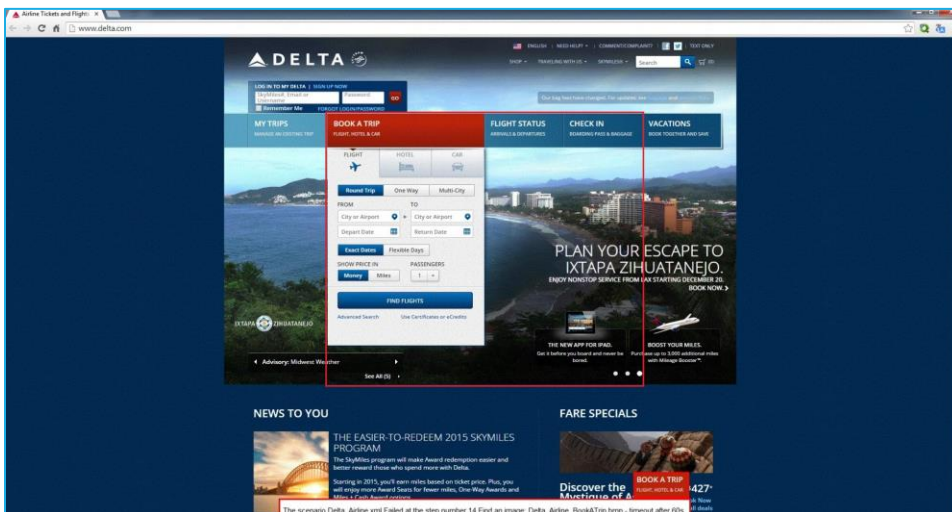
Error Details

```

action = Find image: test scenario a2_book a trip_img1.bmp - timeout after 10s
block = -
description = Bitmap did not appear
error = Citrix Web app Scenario.xml Failed at the step number 20 Find image: test scenario a2_book a trip_img1.bmp - timeout after 10s
scenario = Citrix Web app Scenario.xml
    
```

View Player's Logs | Save Report As CSV | View Failure Screenshot | Show Steps Exec. Time | Close

For example:



The screenshot shows the Delta website interface. At the bottom, a red error message is visible: "The scenario Delta_Airline xml Failed at the step number 14 Find an image: Delta_Airline_BookATrip.bmp - timeout after 60s".



Running scenarios from the command line

This section contain information on running scenarios using command-line parameters and via AWPlayer.

In general, the AWPlayer command line is as follows:

```
AWPlayer.exe [projectfolder] [folder] [filename] [report] [predefinedsscreenshotfilename]
```

projectfolder: This option will have an absolute path to the project folder structure of ScenarioBuilder.

folder: This option will have an absolute path to a folder where all of the files for a scenario exist.

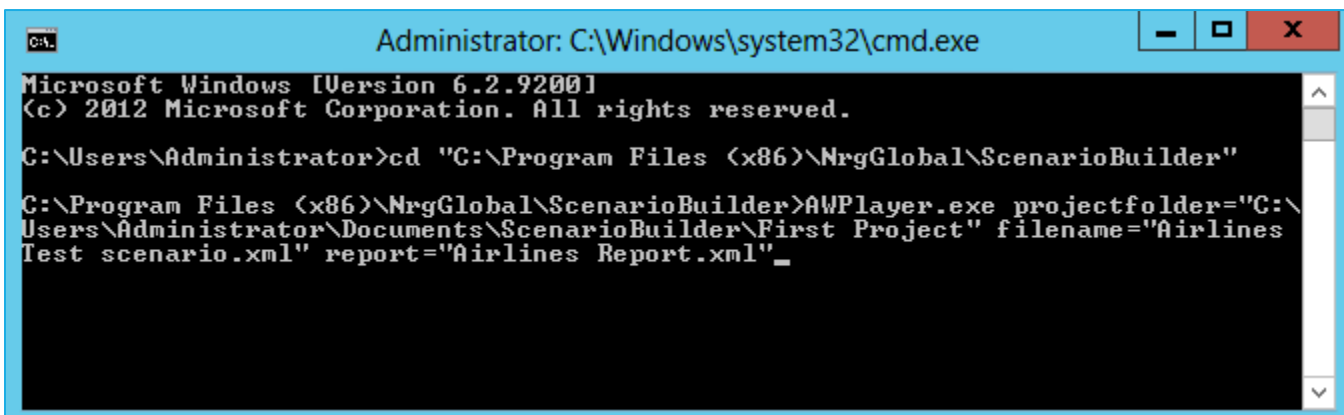
filename: This option will have the name of the scenario name. (i.e Test Scenario.xml)

report: This option will have the name of the xml report that awplayer will generate after running the test.

predefinedsscreenshotfilename: This option will specify an absolute path and the filename of the failure screenshot if it exists.

Here is an example of running script via *AWPlayer* with *CLI*:

In case of success:



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd "C:\Program Files (x86)\MrgGlobal\ScenarioBuilder"
C:\Program Files (x86)\MrgGlobal\ScenarioBuilder>AWPlayer.exe projectfolder="C:\Users\Administrator\Documents\ScenarioBuilder\First Project" filename="Airlines Test scenario.xml" report="Airlines Report.xml" _
```

AWPlayer will start playing the scenario;

After it is done with the execution, it will show you the logs of the execution as follows:



```

Administrator: C:\Windows\system32\cmd.exe

Starting: AwPlayer Version 8.1.2.0
Build date: Oct 12 2015
Build time: 11:14:26

Using Project Folder: C:\Users\Administrator\Documents\ScenarioBuilder\First Pro
ject\
Info: AwPlayer is running under the user Administrator
Start executing scenario: Airlines Test scenario
-> Step 1: Play scenario Open Delta airlines scenario.xml
Start executing scenario: Open Delta airlines scenario
-> Step 1: Open http://www.delta.com
-> Step 2: Find image: open delta airlines scenario_img1.bmp - timeout after 1
0s
End executing scenario: Open Delta airlines scenario => Status Success!
-> Step 2: Play scenario Delta airlines Check in scenario.xml
Start executing scenario: Delta airlines Check in scenario
-> Step 1: Starting transaction Check in
-> Step 2: Find image: delta airlines check in scenario_check in_img1.bmp - ti
meout after 10s
-> Step 3: Move mouse and click
-> Step 4: Find image: delta airlines check in scenario_check in_img2.bmp - ti
meout after 10s
-> Step 5: Move mouse to coordinates (-63, 31) and click
-> Step 6: Set variable Confirmation Number to '102015'
-> Step 7: Type text "102015"
-> Step 8: Keyboard keys: Press tab
-> Step 9: Ending transaction Check in
End executing scenario: Delta airlines Check in scenario => Status Success!
-> Step 3: Play scenario Close Delta airlines scenario.xml
Start executing scenario: Close Delta airlines scenario
-> Step 1: Close application (using Alt-F4 keys)
End executing scenario: Close Delta airlines scenario => Status Success!
End executing scenario: Airlines Test scenario => Status Success!
<

```

Then, after scenario and all sub-scenarios have finished running, *AWPlayer* saves the report of the scenario in a new folder with the scenario name.



In case of failure:

```

Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd "C:\Program Files (x86)\MrgGlobal\ScenarioBuilder"

C:\Program Files (x86)\MrgGlobal\ScenarioBuilder>AwPlayer.exe projectfolder="C:\Users\Administrator\Desktop\First Project" filename="Airlines Test scenario.xml"
report="Airlines Report.xml" predefinedsscreenshotfilename="C:\Users\Administrator\Documents\ScenarioBuilder\airline"

-----
Starting: AwPlayer Version 8.1.2.0
Build date: Oct 12 2015
Build time: 11:14:26
-----

Using Project Folder: C:\Users\Administrator\Desktop\First Project\
Info: AwPlayer is running under the user Administrator
Start executing scenario: Airlines Test scenario
-> Step 1: Play scenario Open Delta airlines scenario.xml
Start executing scenario: Open Delta airlines scenario
-> Step 1: Open http://www.delta.com
-> Step 2: Find image: open delta airlines scenario_img1.bmp - timeout after 10s
End executing scenario: Open Delta airlines scenario => Status Success!
-> Step 2: Play scenario Delta airlines Check in scenario.xml
Start executing scenario: Delta airlines Check in scenario
-> Step 1: Starting transaction Check in
-> Step 2: Find image: delta airlines check in scenario_check in_img1.bmp - timeout after 10s
STEP FAILURE: error = Bitmap did not appear type = execution type error: the test did not succeed step = 2 action=Find image: delta airlines check in scenario_check in_img1.bmp - timeout after 10s
End executing scenario: Delta airlines Check in scenario => Status Failure!
STEP FAILURE: error = Bitmap did not appear type = execution type error: the test did not succeed scenario = Delta airlines Check in scenario.xml block = - step = 2 action=Find image: delta airlines check in scenario_check in_img1.bmp - timeout after 10s
End executing scenario: Airlines Test scenario => Status Failure!
<
error = Bitmap did not appear type = execution type error: the test did not succeed scenario = Delta airlines Check in scenario.xml block = - step = 2 action=Find image: delta airlines check in scenario_check in_img1.bmp - timeout after 10s

```



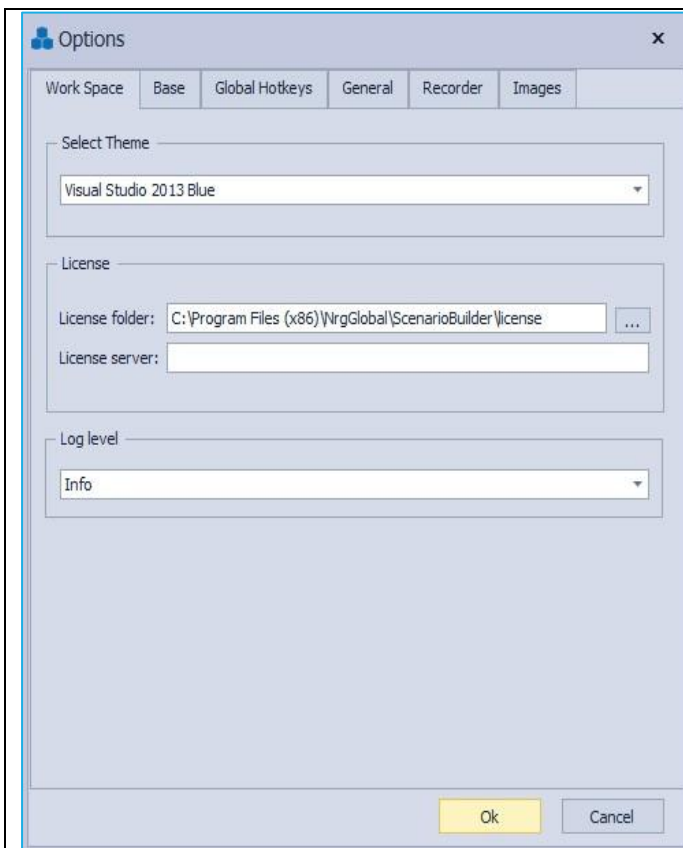
Options

The “Options” menu contains ScenarioBuilder configuration settings for:

- Work Space
- Base
- Global Hotkeys
- Splash Screen Settings
- Recorder

Click “Options” in the “File” menu or go to “View->Options”

WORK SPACE OPTIONS



Select Theme: Change the skin of ScenarioBuilder.

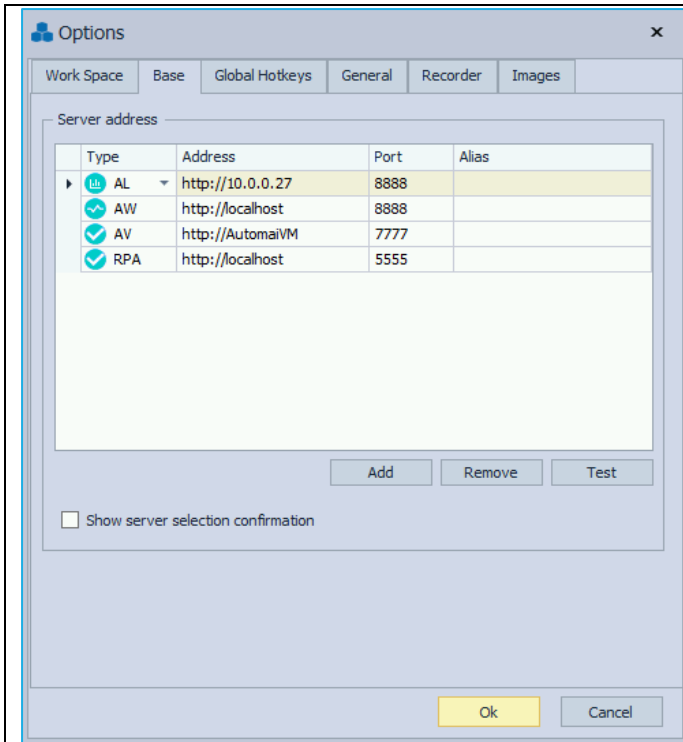
License folder: Select the license folder where all projects will reside. Click the ellipses to navigate to the folder.

License Server: Put the License Server IP address for “Permanent” license and “Token” for “Floating” license.

Log Level: Shows the level of logs to be shown when debugging



BASE OPTIONS



Select the Address and Port of the AppLoader Controller or the AppsWatch Base to which scenarios will be sent.

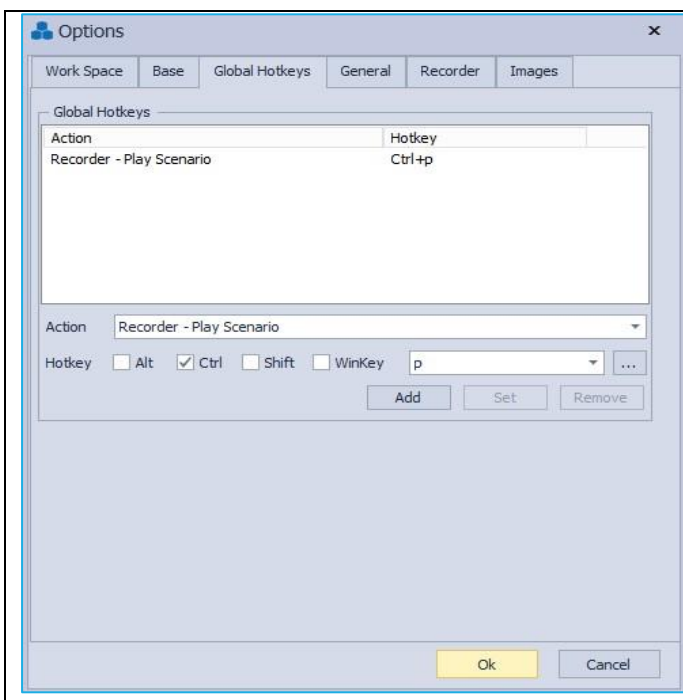
Click “Test” button to confirm communication between ScenarioBuilder and the Controller or Base.

This setting establishes the destination for

scenarios when the  Send scenario option is used.

- ❖ A version can be added to a scenario if “AV” or “RPA” type is selected when sending scenarios to AppVerify or RPA Manager.

GLOBAL HOTKEYS OPTIONS



Manage hotkeys that trigger Actions or menu options in the Scenario Recorder and ScenarioBuilder standard view.

A set of Scenario Recorder hotkeys have been pre-configured. These hotkeys can be edited or removed.

To create a new hotkey:

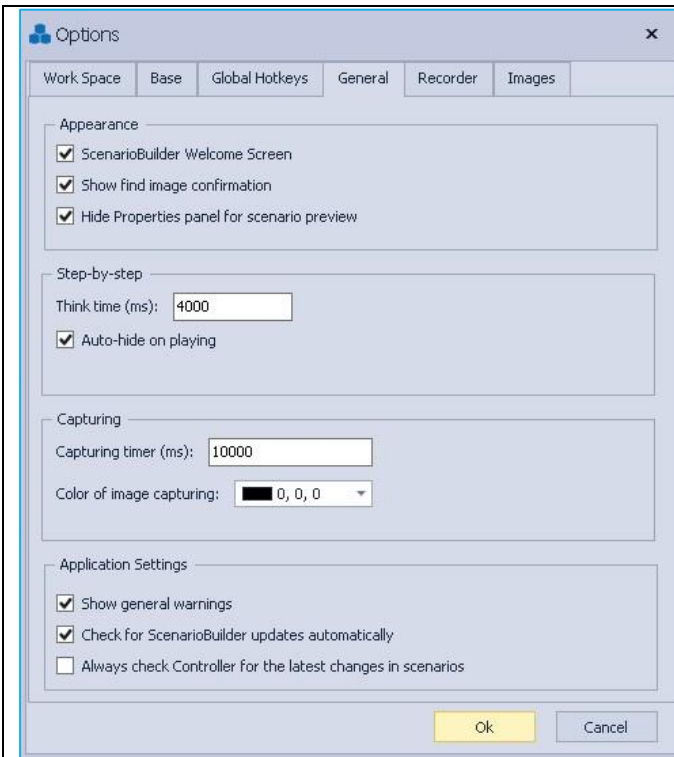
Select an Action from the drop down box;

Select the desired hotkey from the drop down box.

Add *Alt* and/or *Ctrl* and/or *Shift* and/or *WinKey* to further identify the hotkey.



GENERAL OPTIONS



ScenarioBuilder Welcome Screen: Enable or disable the ScenarioBuilder welcome screen.

Show find image confirmation: If enabled, a pop-up confirmation window occurs after each image capture while in Scenario Recorder.

Hide Properties panel for scenario preview: Enable or disable the Properties window.

Think time: Set Think time for Step-by-step playback, this will determine how much time the playback spends on each step. (Time is in milliseconds)

Auto-hide on playing: Enable or disable the Step-by-step menu during playback

Capturing timer : Set the Time to capture an image. (Time is in milliseconds)

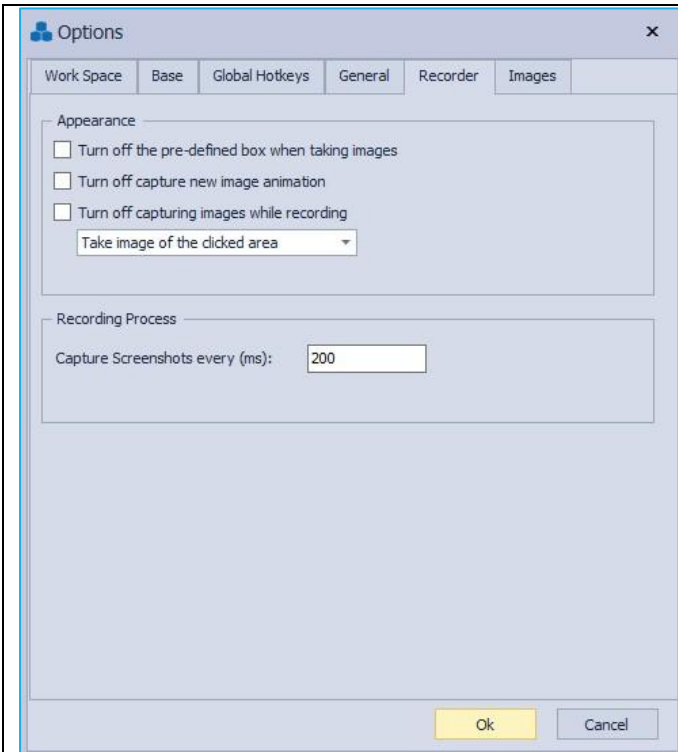
Color of image capturing: Set the capturing color.

Check for ScenarioBuilder updates automatically: If enabled, ScenarioBuilder will check for updates automatically.

Always check Controller for the latest changes in scenarios: If enabled, ScenarioBuilder will check the selected Controller/Base to see if a newer version of a scenario is available, and offers to modify the changes when opening the scenario.



RECORDER OPTIONS



Turn off the pre-defined box when taking images: If checked, image capturing will require pressing [CTRL], then clicking and dragging with mouse around the desired image.

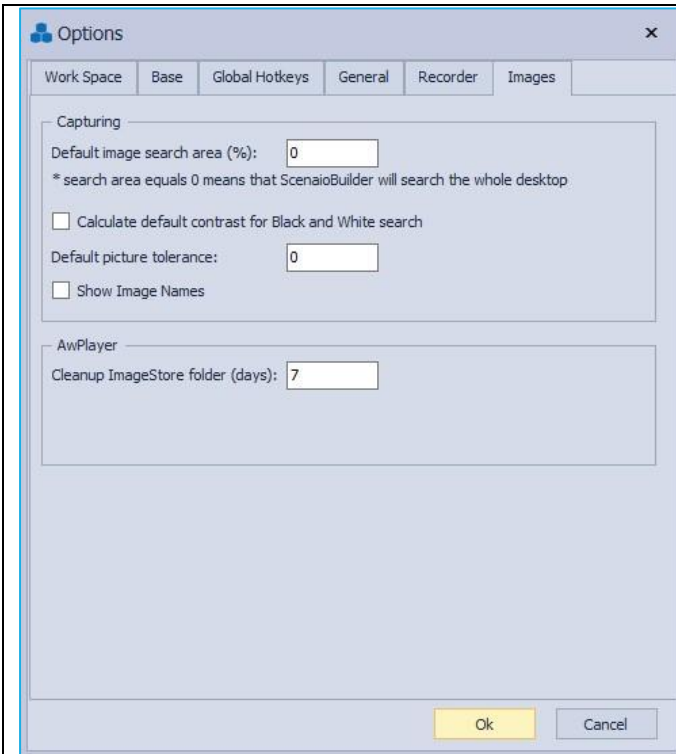
Turn off image change confirmation: Enables or disables image change confirmation.

Turn off capturing images while recording: Enables or disables “Capture Images Automatically”.

Capture Screenshots every (ms): During the recording process, Scenario Recorder checks the screen for the current anchor image, alerting the user when the anchor image has changed or is no longer present. Use this setting to establish how often Scenario Recorder checks for changes.



IMAGES



Default image search area: This setting establishes the area for image search. The default is 1000 percent.

Calculate default contrast for Black and White search: Enables or disables to calculate the default contrast for Black and White search.

Default picture tolerance: This setting establishes the tolerance for image. The default is 0 percent.

Show Image Names: Enable or disable to show images names in the middle pane.

Cleanup ImageStore folder (days): The failure screenshots will be deleted from ImageStore folder after the specified number of days.



Getting help and support

Free POC!

Need a Proof of Concept? Our experts will help establish ScenarioBuilder's viability so that your organization, from stakeholders down to data entry clerks, is 100% behind your decision to go forward with ScenarioBuilder!

Support Team

If at any point during your trial you need more help, our technicians are waiting to help you! Don't hesitate to call our experts at +1 833-288-6624 (+1 833-AUTOMAI) or email Automai Support support@automai.com / help@automai.com.

Help Center

For one-on-one help, create a support request at [Automai Help Center](#). A support engineer will follow up with you quickly. When submitting your support request, please provide as much detail as possible. This will help us resolve your issue faster.

Online Help

If you can't find what you need in the documentation, you can ask questions, search for answers, exchange comments and suggestions on our [Help Center](#), find answers to your [Frequently Asked Questions](#) and watch [Training Videos](#).

Send us feedback

We love to hear your thoughts on ScenarioBuilder and your ideas for future features and improvements. Also, occasionally bugs do happen. If you find one, we'll do our best to fix it.

ScenarioBuilder "Help" section includes a ticket submission link for you to submit your problems, questions and suggestions about the product. Let us know what you think!

