

# **Desktop Application Monitoring**

# **Recorder & Scripting**

## **User Manual**

8 May 2019



## Table of Contents

1	INT	RODUCTION	4
2	BAS	IC INFORMATION	4
	2.1	SCREEN RESOLUTION AND SCREEN COORDINATES	4
	2.2	LOCATION OF FILES	5
	2.3	Best practices	5
3	INS	TALLING THE APPLICATION	7
4	THE	DESKTOP APPLICATION RECORDER USER INTERFACE	7
	4.1	Edit Scenario Tab	7
	4.2	Test Run Tab	9
	4.3	FILES TAB	10
5	REC	ORDING A SCENARIO	. 10
6	ADI	)/EDIT/DELETE COMMAND	. 12
	6.1	Add	13
	6.2	Edit	14
	6.3	Delete	14
7	VAL	IDATION AND FLOW CONTROL	. 15
	7.1	WAIT FOR APPLICATIONS TO START / WINDOWS TO APPEAR	15
	7.2	WAIT FOR NON-TEXT CONTENT TO APPEAR ON SCREEN	16
	7.3	Assert that Text is Present on the Screen	17
	7.4	Assert that Image is Present on Screen	19
	7.5	PAUSE SCENARIO EXECUTION	20
8	MAT	ГСНING TEXT	. 20
9	WIN	IDOW MANAGEMENT	. 22
	9.1	Position Window	22
	9.2	Focus Window	23

#### Apica Excellence in Cloud Performance

9	.3	MAXIMIZE WINDOW	23
9	.4	Resize Window	23
9	.5	CLOSE WINDOW	24
10	IGNO	DRE TIMING	24
11	OPE	N/SAVE SCENARIOS	24
1	1.1	SAVE A SCENARIO	24
1	1.2	OPEN (LOADING) A SCENARIO	25
12	TEST	Γ RUN A SCENARIO	25
13	СОМ	MAND REFERENCE	26
14	SCEN	VARIO FORMAT	31
15	APP	ENDIX - TESTING A SCENARIO THROUGH THE COMMAND LINE	32
16	APP	ENDIX - TESTING A SCENARIO THROUGH THE API	32
1	6.1	Start a job	32
1	6.2	GET JOBSTATUS	32
1	6.3	Get a result	33

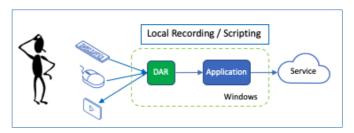


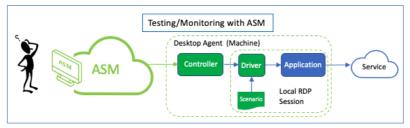
## 1 Introduction



Apica's Desktop Application Monitoring is a set of applications and services that measure Microsoft Windows desktop applications. This document is the User's Guide to recording and scripting user scenarios, to be used to monitor desktop Application availability and performance.

The Desktop Application Recorder (DAR) is the Windows Desktop Application part of the solution. There is also the Desktop Application Agent ("Desktop Agent"), used for executing the checks, and typically installed on the customer premises (a so-called "private agent").





Finally, there is the monitoring part of the solution: A Desktop Application Check (DAC) which is the deployed Scenario that is run from the Desktop

Application Agent for long term monitoring of the targeted Windows application. This is the check that ASM will report the metrics back as part of the analytics.

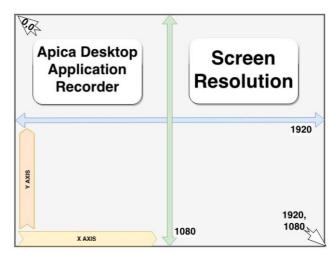
## 2 Basic information

#### 2.1 Screen resolution and screen coordinates

Capturing the desktop application often replies on where the cursor/mouse position is when recording the sequence of events. Many of the commands use the position of the cursor as arguments, e.g. leftClickAt(x,y). So, it is essential to use the same screen resolution for both recording and executing the scenarios.

Default Resolution: The default screen resolution for the Desktop Application Agent (which executes the scenarios) is 1920 x 1080, so Apica recommends this when recording scenarios. The grid is laid out as follows:

- The upper left corner has
- coordinates x=0, y=0.
- The lower right corner has coordinates x=1920, y=1080.
- Where
  - X is horizontal (left/right)
  - Y is vertical (up/down).
- When selecting an area, e.g. with the command assetTextAt(x,y,width,height) the x,y point is in the upper left corner of the area/box



#### 2.2 Location of files

Scenarios are the recorded series of DAR-captured steps. These scenarios are stored in a directory, specified in a properties file.

For the developer installation, the default directory location for scenario files is "C:/apica/asm-desktop-agent/embedded/asm-desktop-agent/scenarios/".

This location is set in the "C:/apica/asm-desktop-agent/embedded/asm-desktop-agent/application.properties" file.

You can change the location of the scenarios by editing this properties file.



Note: It's only possible for the Desktop Application Agent to test-run scenarios that are located in the set default location.

#### 2.3 Best practices

When starting an application (to be tested):

- Avoid using click-on-icon, neither on the desktop nor on the taskbar.
- Avoid the start-menu.

Why: the agent uses a remote desktop session to run the test, and that user/desktop may not have the necessary icon in place (or the same place). Use the startApplication command instead.

Sync the Window Size: After starting the application, Apica recommends either maximizing the window, or positionWindow(x,y) in order to have the coordinates be the same as what was captured in the recording to what is played back during the test session.



Note: Apica recommends avoiding Exact type matches when matching text because its match is very strict and does not accept (as an example) any extra spaces before/after. Apica recommends using "contains" as a preferred match type.

Remember to assert that the application interface has been completely rendered before you continue with the next command. For instance, trying to assert that a certain text string is present on the screen too quickly may fail, purely because it has not been rendered yet. It is recommended that you e.g. use the waitForPixel command first and assert after this.

When using the startApplication command to start the Windows command prompt, use the following arguments (see command reference)

- "application": "C:\\Windows\\System32\\cmd.exe",
- "applicationArguments": "/c start cmd"

When using the startApplication command to run a bat script, use the following arguments (see command reference)

- "application": "C:\\Windows\\System32\\cmd.exe",
- "applicationArguments": "/c start some\_script.bat"

When using the command "assertImage", try selecting an image with many colors and variations as monotoned images are more difficult to match. Apica recommends a size of around 100x100 pixels.

## 3 Installing the application

The Desktop Application Monitoring installation guide is provided separately from this manual.

## 4 The Desktop Application Recorder User Interface

The main interface of the application has three tabs:

- Edit Scenario
- Test Run
- Files

#### 4.1 Edit Scenario Tab

Recording and editing scenarios.

dit Scapario Test Due Star		
dit Scenario Test Run Files		
Recording: 1 Disabled O Enabled	Cursor Position: 0:0 2 RGB: 0 0 0	3
Command 4	Arguments	Flags
startStep	stepName: initialStep	
startApplication	application: C:\Windows\System32\calc.exe, name: calc	
waitForTitle	title: Calculator, timeout: 30000, matchingType: contains	
positionWindow	x: 200, y: 200	ignoreTimings
leftClickAt	x: 252, y: 649	
leftClickAt	x: 314, y: 647	
leftClickAt	x: 492, y: 545	
type	text: 12	
leftClickAt	x: 481, y: 701	
assertTextAt	text: 144, x: 432, y: 302, width: 92, height: 52, matchingType: contains	
leftClickAt	x: 505, y: 222	
Add Command Command Details	(10 11	Clear
Command: 8 StartStep	Save Revert	12 Delete
Name: initialStep		



Edit Scenario Guide:

#	ltem	Explanation
1	Recording	Disable or Enable recording
2	Cursor Position	While automatic recording is enabled, the position of the mouse cursor is displayed here
3	Pixel Color	While automatic recording is enabled, the RGB value of the pixel at mouse cursor position is displayed here.
4	Scenario Table	The name of the recorded command
		Arguments: argument values
		Displays enabled flags (if any)
5	Add Command Button	Adds a new command to the scenario
6	Import File Button	Import a scenario from file
7	Clear Button	Clear/Delete all commands
8	Command Details	Shows details of the currently selected command, or a new command
9	Command Drop- Down	Shows the currently selected command, when adding a new command, you may select type of command here
10	Save Button	Click here to save any changes made to the currently selected command, or to save a new command
11	Revert Button	If you made changes to the currently selected command,
		click here to revert to the original arguments
12	Delete	Removes the selected command from the scenario

## 4.2 Test Run Tab

Testing/verifying a new recording or imported scenario.

🔺 Desktop Recorder		- 0	×
Edit Scenario Test Run Files			
Test Run Options			
Delay Between Commands (n	ns): 1000 1	Run in Recorder	
		Run in Agent	
Desktop Agent Install Directo			
Recorded Scenario:			
Command 5	Arguments	Flags	
startStep	stepName: initialStep		
startApplication	application: C:\Program Files\internet explorer\iexplore.exe,		
waitForTitle	title: Blank Page, timeout: 30000, matchingType: contains		
positionWindow	x: 0, y: 0		
leftClickAt	х: 987, у: 40		
leftClickAt	x: 837, y: 292		
waitForTitle	title: Internet Options, timeout: 30000, matchingType: conta		
leftClickAt	x: 411, y: 35		
leftClickAt	x: 390, y: 38		
type	text: www.apicasystems.com		
pressEnter			
assertTextAt	text: trial, x: 382, y: 558, width: 57, height: 31, matchingType		
assertimage	template: /9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgc		
	6 Show last driver log	Show last driver result	7



Test Run Guide:

	Item	Explanation
1	Delay between command slider	Use the slider to set a delay (in ms) between each command in the scenario, the default delay is 1000 ms.
2	Installation Directory	The scenario directory is located here.
3	Run in Recorder	Start a test run of your scenario through the recorder. Note that only a subset of the commands can be played back (executed) by the desktop recorder, to enable full command support you must use "Run in Agent"
4	Run in Agent	Start a test run of your scenario through the real Desktop Agent. See Test Run for more information.
5	Recorded Scenario Table	<ul> <li>The name of the recorded command</li> <li>Arguments: Recorded argument values</li> <li>Displays enabled flags (if any)</li> </ul>
6,7	Show last driver log Show last driver result	- Show log and result

#### 4.3 Files Tab

Opens (loads), or Saves, the scenario in JSON format. Files Guide:

	Item	Explanation
1	Name	Name of scenario
2	Description	Field for optional scenario description
3	Include Global Pause Timing	Adds a pause command in between each command in the scenario, with a delay that corresponds to the delay configured in the "Test Run" tab.
4	Save as / Open	Opens a traditional windows file dialog.

## 5 Recording a scenario

The Desktop Recorder can record the following actions automatically: leftClickAt, rightClickAt, doubleClickAt, mouseMove & dragTo (click and drag), type, pressEnter, pressEscape, pressBackspace, pressTab.



To start the recording select the "Enabled" radio button, under "Automatic Recording".

Scenario Test Run F	iles				A Desktop Recorder				- 0	2
ecording:	Cursor Position:	Pixel Color:			Edit Scenario					
) Disabled 🖲 Enable		RGB: 20 25 2	9		■ Recor <sup>at</sup> ing: ■ Disabled □ Enabled	Cursor Position: 0:0	Pixel Color: RGB: 0 0	0		
ommand	Arguments		Flags		Command	Arguments		Flags		_
rtStep	stepName: initialStep				startStep	stepName: initialStep				
					startApplication	application: C:\Windows\System	32\calc.exe. name: calc			
					waitForTitle	title: Calculator, timeout: 30000,				
					positionWindow	x: 200, y: 200		ignoreTimings		-
					leftClickAt	x: 252, y: 649				
					leftClickAt	x: 314, y: 647				
					leftClickAt	x: 492, y: 545				
					type	text: 12				
					leftClickAt	x: 481, y: 701				
					assertTextAt	text: 144, x: 432, y: 302, width: 92	, height: 52, matchingType:			
										-
Add Command			C	lear	Add Command				Clear	
mmand Details					Command Details					
mmand:		Save Revert		Delete	Command:		Save Revert		Dele	te
artStep	Ignore Timing				StartStep	Ignore Timing				
ime:					Name:					
itialStep					initialStep					
					Innuiscep					

When you are done select the checkbox labeled "Disabled" to stop the recording.

Note: Mouse-clicks on/in the Desktop Recorder are not recorded.

The following commands are recordable:

- leftClickAt
- rightClickAt
- doubleClickAt
- mouseMove
- dragTo
- type
- pressEscape
- presenter
- pressBackspace
- pressTab



## 6 Add/Edit/Delete Command

When you open the Desktop Recorder you will be presented with a new scenario, which contains a single "startStep" command. You may change the name of the step by selecting the command and then entering a new name in the text box labeled "Name", save the command by clicking the "Save" button.

Desktop Recorder							
dit Scenario Test Run Files							
Recording: Disabled  Enabled	Cursor Position: 564:0	Pixel Co RGB:		25 29			
Command	Arguments				Flags		_
startStep	stepName: initialStep						
Add Command						Clear	
						Clear	
Command Details		Save	Reve	ert		Clear	te
Command Details	→ 🗌 Ignore Timing	Save	Reve	ert			te
Command Details Command:	✓ ☐ Ignore Timing	Save	Reve	ert			te
Command Details Command: StartStep	✓ □ Ignore Timing	Save	Reve	ert			te
Command Details Command: StartStep Name:	✓ 🗌 Ignore Timing	Save	Reve	ert			te
Command Details Command: StartStep Name:	✓ 🗌 Ignore Timing	Save	Reve	ert			te
Command Details Command: StartStep Name:	✓ 🗌 Ignore Timing	Save	Reve	ert			te



#### 6.1 Add

To add a new command, start by clicking "Add Command" and then select a command in the drop-down list labeled "Command".

🔺 Desktop Recorder					-		×	
Edit Scenario Test Run Files								
Recording: O Disabled   Enabled	Cursor Position: 564:0	Pixel Color: RGB: 20	25 29	,				
Command	Arguments			Flags				DoubleClickImage
startStep	stepName: initialStep							DragTo
Add Command Command Details Command: StartStep Name: initialStep	✓ □ Ignore Timing	Save	Revert			Clear	te	FocusWindow LeftClickAt LeftClickAt LeftClickImage MaximizeWindow MouseMove Pause PositionWindow PressArrowKey PressBackspace PressEcape PressEcape PressEcape PressWitAlt PressWitAlt PressWitAlt PressWitAlt PressWitAlt ResizeWindow RightClickAt RightClickAt RightClickAt RightClickAt RightClickAt StrolDown ScrolIUp StartStep StopApplication TakeScreenshot Type WaitForNotPixelColor WaitForRixelColor WaitForRixelColor WaitForRixelColor WaitForRixelColor



You will be presented with several input fields (arguments) which must be filled out before you can use the save for that command.

🔺 Desktop Recorder						-		×
Edit Scenario Test Run Files								
Recording: O Disabled   Enabled	Cursor Position: 1017:0	Pixel Co RGB:		33	33			
Command	Arguments			F	lags			
startStep	stepName: initialStep							
Add Command						(	Clear	
Command Details		Save	Re	vert			Dele	te
AssertTextAt	Ignore Timing							
	dth: Height:	Text:						
0 🗘 0 🗘 0	↓ 0 Capture	Apica						
Matching Type:								
Contains $\checkmark$								

To see the how the different commands work, see the command reference list.

Once you have entered values for all required fields, click "Save" to add the command to your scenario.

#### 6.2 Edit

To edit a command, select the command in the list, edit one or more arguments and then click "Save" to save the changes or "Revert" to revert the changes.

#### 6.3 Delete

To delete a command, select the command in the list and click "Delete" to remove it from the scenario.



<b>A</b>	Desktop Recorder		x	<u>A</u>	Desktop Record	der 📃 🗖 🗙
Edit Scenario Test Run Export				Edit Scenario Test Run Exp	ort	
Automatic Recording: Disabled O Enabled	Cursor Position: 0:0	Pixel Color: RGB: 0 0 0		Automatic Recording:	Cursor Position: 0:0	Pixel Color: RGB: 0 0 0
Command	Arguments	Flags		Command	Arguments	Flags
startStep	stepName: initialStep			startStep	stepName: initialStep	
startApplication	application: C:\Program Files (	x8				
Add Command Impe Command Details Command: StartApplication Application Name: ie Path: C:\Program Files (x86) Select App	v Ignore Timing	Clear Save Revert Delet		Add Command Command Details Command: StartStep Name: initialStep	Import File V  Ignore Timing	Clear Save Revert Delete

## 7 Validation and Flow Control

#### 7.1 Wait for Applications to Start / Windows to Appear

A scenario typically starts with a startApplication command which starts an instance of the application that the user wishes to test. Whenever you start an application that spawns a new window—or whenever you switch between windows (using the focusWindow command)—you should be able to use the waitForTitle command to wait for a new window (with a title that matches the "Title" argument) to appear. If an application opens pop-up windows within the application, these windows are also typically labeled, and you should be able to use the waitForTitle command to pause the scenario execution until these windows appear.

You can also use the *Matching Type* argument to select matching type: exact, glob (global expressions), or "contains".

In the following example, we start by opening Internet Explorer, and then we wait for the title of the currently focused window to match "Blank Page", using the "contains" option, with a timeout set to 30 seconds.

Recording:       Cursor Position:       Pixel Color:         Disabled       Enabled       1634-259       RGB:       229       241       251         command       Arguments       Flags         tartStep       stepName: initialStep       stepName: initialStep         application:       c:\Program Files\internet explorer/iexplore.exe, n         valtForTitle       title: Blank Page, timeout: 30000, matchingType: contains         Add Command       Clear         Command Details       Save         Command Title       Ignore Timing         Title:       Matching Type:         Matching Type:       Timeout (ms)	A Desktop Recorder			-		<	
Ibisabiled Enabled 1634:259 RGB: 229 241 251   Command Arguments Flags   tartStep stepName: initialStep tartApplication: C:\Program Files\Internet explorer\iexplore.exe, n   autForTitle title: Blank Page, timeout: 30000, matchingSype: contains    Add Command Clear  Command:  Save Revert Delete  WaitForTitle  Ignore Timing  Title: Matching Type: Timeout (ms)	Edit Scenario Test Run Files					Ø Blank Page × □ <sup>*</sup>	
Command Arguments Flags CardStep StepName: initialStep Flags CardStep StepName: initialStep Flags CardStep StepName: initialStep Flags CardStepName: Clear CardStepName Clear CardStepName Clear Clear Command Details Command: Save Revert Delete WatforTitle Ignore Timing Title: Matching Type: Timeout (ms)	Recording:	Cursor Position:	Pixel Color:				
tartStep stepName: initialStep application: C\Program Files\Internet explorer\Uexplore.exe, n Add Command Add Command Cclear Command Details Command: Save Revert Delete WaitForItite V   Ignore Timing Title: Matching Type: Timeout (ms)	Disabled O Enabled	1634:259	RGB: 229 241 251				
tartApplication application: C:\Program Files\internet explorer\isexplore.exe, n Add Command Add Command Clear Command Command: Command	Command	Arguments		Flags			
Add Command Clear Command Details Command: WaitForTitle V   Ignore Timing Title: Matching Type: Timeout (ms)	startStep	stepName: initialStep					
Add Command Clear Command Details Command: Save Revert Delete WaitForTitle V Grow Timeout (ms)	startApplication	application: C:\Program Files	\internet explorer\iexplore.exe, n				
Command Details Command: Save Revert Delete WaitForTitle Jignore Timing Title: Matching Type: Timeout (ms)	waitForTitle	title: Blank Page, timeout: 30	0000, matchingType: contains				
Command Details Command: Save Revert Delete WaitForTitle Vignore Timing Title: Matching Type: Timeout (ms)							
Command: Save Revert Delete WattForTitle Jignore Timing Title: Matching Type: Timeout (ms)	Add Command			(	Clear		
WaltForTitle Ignore Timing Title: Matching Type: Timeout (ms)	Command Details						
Title: Matching Type: Timeout (ms)	Command:		Save Revert		Delete		
	WaitForTitle	Ignore Timing				-	
Blank Page Contains V 30000 🖨	Title:	Matching Type:	Timeout (ms)				
	Blank Page	Contains ~	30000 🜩				



Next, we open "Internet Options" and wait for the title of the currently focused window to match "Internet Options"

Cursor Position: Pixel Color: 1089:253 RGB: 26 134 219		General Se				
				Content Connection	Dreama	Advanced
		Home page	e	age tabs, type each a	_	
Arguments	Flags		about blank			<u></u>
epName: initialStep						$\sim$
pplication: C:\Program Files\internet explorer\iexplore.exe, n			Line was	and the defend		and take
tle: Blank Page, timeout: 30000, matchingType: contains			Use curr	use defau	t Use r	ew tab
971, y: 182						
833, y: 437		Ŭ		e last session		
tle: Internet Options, timeout: 30000, matchingType: contains		Start	t with home page			
	Clear	Delete t	emporary files, hist ormation.		asswords, ar	d web
			te browsing history	y on exit		
	Delete			Delete	Se	ttings
V Ignore Timing		Appearan	ce			
Iatching Type: Timeout (ms)		Colo	ors Langu	uages Fonts	Acce	essibility
				OK	Cancel	Apply
	Palvane: initialStep pilication: C\Program Files\internet explorer\iexplore.exe, n le: Blank Page, timeout: 30000, matchingType: contains 971, y: 182 833, y: 437 le: Internet Options, timeout: 30000, matchingType: contains Save Revert Save Revert Image Ignore Timing atching Type: Timeout (ms)	Paylame: initialStep	Paper	Paylame: initialStep         iplication: C:\Program Files\internet explorer\explore.exe, n         ie: Blank Page, timeout: 30000, matchingType: contains         971, y: 182         833, y: 437         ie: Internet Options, timeout: 30000, matchingType: contains         Clear         Save         Save         Save         Revert         Delete         ignore Timing         atching Type:         Timeout (ms)	ppName: initialStep   iplication: C\Program Files\internet explorer\explore.exe, n   ite: Blank Page, timeout: 30000, matchingType: contains   971, y: 182   833, y: 437   ite: Internet Options, timeout: 30000, matchingType: contains   ite: Internet Options, timeout: 30000, matchingType: contains    Clear    Clear    Clear  Delete temporary files, history, cookies, saved priorm information.  Delete temporary files, history on exit Delete  Appearance Colors Languages Fonts	pplane: initialStep   iplication: C\Program Files\internet explorer\iexplore.exe, n   ite: Blank Page, timeout: 30000, matchingType: contains   971, y: 182   833, y: 437   ite: Internet Options, timeout: 30000, matchingType: contains   Clear   Save   Revert   Delete   Save   Revert   Delete   Save   Revert   Delete   Save   Revert   Delete   Timeout (ms)   30000 •

#### 7.2 Wait for Non-Text Content to Appear on Screen

If you need to wait for some generic content to appear on the screen, you may be able to use waitForPixelColor command to wait for the RGB value of a specific pixel on the screen, to match a pre-defined value captured by the Desktop Recorder.

In this example, we start by opening Internet Explorer and then navigate to <u>www.apicasystems.com</u>



<b>A</b>	Desktop Recorder	_ <b>_</b> X
Edit Scenario Test Run Export		
Automatic Recording: Disabled      Enabled		Pixel Color: RGB: 255 255 255
Command	Arguments	
startStep	stepName: initialStep	
startApplication	application: C:\Program Files (x86)\l	internet Explorer\iexplore.exe, n
pause	duration: 5000	
waitForTitle	title: Blank Page - Internet Explorer,	, timeout: 0, matchingType: exact
Add Command Impo Command Details Command: StartApplication Application Name:	v Ignore Timing	Clear Save Revert Delete
Application Name: ie Path: C:\Program Files (x86)\ Select App	Interne	

Next, we add a "WaitForPixelColor" command, click "Pick" and then click on the green background of the apicasystems.com site. When the scenario runs (after pressing <enter> to load apicasystems.com), the scenario execution will pause

<b>A</b>	Desktop Recorder	-		x
Edit Scenario Test Run Export				
Automatic Recording:	Cursor Position: Pixel Color: 157:190 RGB: 255 255	255		•
Command	Arguments			^
startApplication	application: C:\Program Files (x86)\Internet Explorer\iex	plore.e	xe,	
pause	duration: 5000			
waitForTitle	title: Blank Page - Internet Explorer, timeout: 0, matchin	gType:	ex	=
type	text: www.apicasystems.com			
pressEnter				
				~
Add Command Impo	rt File	Clea	r	
Command Details				
Command: WaitForPixelColor	Save Revert		Delet	e
X: Y: Red: 269 v 279 v 137	Green: Blue:			

until the background color changes.

7.3 Assert that Text is Present on the Screen



You can use assertText or assertTextAt to assert that a string of text is present on the screen. Note that the recorder is not able to re-play (execute) these commands. To test them, you need to check the "Use full command support" checkbox.

#### assertText

The *assertText* command captures a screenshot of the currently focused application <u>window</u>, feeds the image to the OCR engine (<u>Tesseract</u>) which returns any text detected. The *Text* argument is then compared to the text returned by Tesseract. If it matches, the command is successful. You can also use the *Matching Type* argument to select matching type: exact, glob (global expressions), or "contains".

#### assertTextAt

The *assertTextAt* command captures a screenshot of a specific *part of the screen*, feeds the image to the Tesseract OCR engine which returns any text detected. The *Text* argument is then compared to the text returned by Tesseract. If it matches, the command is successful. You can also use the Matching Type argument to select matching type: exact, glob (global expressions), or "contains".

<u>Example of assertTextAt</u>: To select the part of the screen that will be passed to Tesseract start by clicking the "Capture" button. A transparent overlay is added on top of the screen.

Click and hold the left mouse button and drag the mouse to select a rectangle on the screen, making sure that the text you wish to detect is completely within this rectangle, once you release the mouse, the arguments in command details should be updated automatically.

In the following example, we start by opening Internet Explorer and then navigate to www.apicasystems.com. Next, we add an "AssertTextAt" command and click "Capture", this adds an overlay on top of the desktop, much like the snipping tool in Windows. Then, we select a part of the screen that covers the text we wish to detect (in this case we selected the word 'TRIAL' in the "REQUEST FREE TRIAL" button), when the mouse button is released the WIDTH, HEIGHT, X and Y arguments should be updated automatically.



			-		
Why Apica Product ~ Compar	🔺 Desktop Recorder		-		×
Apica	Edit Scenario Test Run Files				
	Recording:	Cursor Position: Pixel Color: 1127:239 RGB: 255 25	5		
Performance	Command	Arguments	Flags		^
	leftClickAt	x: 987, y: 40			
Monitoring for the	leftClickAt	x: 837, y: 292			
	waitForTitle	title: Internet Options, timeout: 30000, matchingType: contai			
Enterprise	leftClickAt	x: 411, y: 35			
	leftClickAt	х: 390, у: 38			
Apica helps companies with	type	text: www.apicasystems.com			
	pressEnter				
complex IT infrastructure monito	assertTextAt	text: trial, x: 382, y: 558, width: 57, height: 31, matchingType:			
business critical applications an	assertimage	tempiate: /9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgc			
API	Add Command Command Details Command:	Save Revert		Clear	
	Assertimage	Ignore Timing		Delet	.c
REQUEST FREE TRIAL	Assertimage				
SCHEDULE DEMO	Capture Captured (6	9 × 68 px) <b>Preview</b>			
SCHEDULE DEMO					

Next, in the input field labeled "Text", we enter a string of text to match against and the type of match (exact/contains/glob). When the scenario runs with this condition, it will stop executing if we cannot detect that text within the specified coordinates.

Note that these commands work best when you try to match a single word or string of text, and *that matches are case insensitive* 

#### 7.4 Assert that Image is Present on Screen

Use the "AssertImage": command to stress that a particular image must be present on the screen, as a condition. First, you must capture the image to assert. So, start by clicking the "Capture" button. A transparent overlay is added on top of the screen, click and hold the left mouse button, and drag the mouse to select a rectangle on the screen, once you release the left mouse button the arguments should be updated automatically. Apica recommends a size of around 100x100 pixels.

**Important**: The captured image must be distinctive and not found on any other part of the screen that is being evaluated.



In this example we start by opening Internet Explorer and load www.apicasystems.com. Next, we add an "AssertImage" command and click "Capture", this adds an overlay on top of the desktop, much like the snipping tool in Windows. Then, we select the part of the screen that we wish to use for verification. In this case, we select the Apica logo. So, when the scenario runs, if we cannot detect the Apica logo on the screen, it will stop executing.

〈テリ(ジ) 📤 https://www.apicasystems.com/	٣	📾 🖒 🛛 Search	<b>/ କ</b> ୍ (ଜି ୩	は 🙂 📥 🔜 🖉
🍯 Blank Page 🔺 Apica: Application Perform 🗴 📑				15175
Mhy Apica Product - Compar	🔺 Desktop Recorder			- 🗆 X
Apica	Edit Scenario Test Run Files			
	Recording:	Cursor Position: 1127:239	Pixel Color: RGB: 255 255 25:	5
Performance	Command	Arguments		Flags ^
	leftClickAt	x: 987, y: 40		
Monitoring for the	leftClickAt	x: 837, y: 292		
	waitForTitle	title: Internet Options, timeout: 30000	matchingType: contai	
Enterprise	leftClickAt	x: 411, y: 35		
	leftClickAt	x: 390, y: 38		
Apica helps companies with	type	text: www.apicasystems.com		
	pressEnter			
complex IT infrastructure monito	assertTextAt	text: trial, x: 382, y: 558, width: 57, heig		
business critical applications an	assertimage	template: /9j/4AAQSkZJRgABAQEAVAB	AAD/2w8DAAgGBgc	
API	Add Command			Clear
	Command Details			1000 - 20 - 20 - 20 - 20 - 20 - 20 - 20
	Command:		Save Revert	Delete
REQUEST FREE TRIAL	Assertimage	Ignore Timing		
	Capture Captured (69	9 × 68 px) Preview		
SCHEDULE DEMO				

#### 7.5 Pause Scenario Execution

In some cases, the script/scenario can run faster than the application can respond, causing false execution errors. In these cases, if there is a reason to wait for the application to render a response, use the "Pause" command to pause the scenario execution for X milliseconds.

## 8 Matching text

For any command where we match text (assertTitle, waitFortitle, assertText, assertTextAt, waitForTextAt), the Desktop Application Recorder includes the option to select the type of matching performed (exact/contains/glob). Again: Text matches for these types are <u>case in-sensitive</u>.

#### Exact: The returned text must match the recorded value exactly.

Apica recommends avoiding Exact type matches when matching text because its match is very strict and does not accept (as an example) any extra spaces before/after. Apica recommends using "contains" as a preferred match type.



**Contains:** The returned text must contain the recorded value.

**Glob:** (Global expressions) The returned text must match the recorded value based on a specified pattern of single character using a question mark '?' or multiple/no characters using an asterisk '\*'.

The matching type - glob works as follows:

Wildcard	Description	Example	Matches	Does not match
*	matches any number of any	Law*	Law, Laws, or Lawyer	GrokLaw, La, or aw
	characters including none	*Law*	Law, GrokLaw, or Lawyer.	La, or aw
		*obot	Robot, obot, mobot	I am a robot
		* robot	I am a robot	Obot, I am a obot
?	matches any single character	?at	Cat, cat, Bat, bat	at

## 9 Window Management

## 9.1 Position Window

Use PositionWindow to position the currently focused application window at coordinates X,Y. To place the window in the upper left window, enter coordinates 0,0.

📥 Desktop Recorder	– 🗆 X
Edit Scenario Test Run Files	
Recording:	Cursor Position: Pixel Color: 1127:239 RGB: 255 255
Command	Arguments Flags
startStep	stepName: initialStep
startApplication	application: C:\Program Files\internet explorer\iexplore.exe,
waitForTitle	title: Blank Page, timeout: 30000, matchingType: contains
positionWindow	х: 0, у: 0
leftClickAt	x: 987, y: 40
leftClickAt	x: 837, y: 292
waitForTitle	title: Internet Options, timeout: 30000, matchingType: contai
leftClickAt	x: 411, y: 35
leftClickAt	x: 390, y: 38
type	text: www.anicasystems.com
Add Command	Clear
Command Details	
Command:	Save Revert Delete
PositionWindow	✓ ☐ Ignore Timing
X: Y: 0 • 0 •	

#### 9.2 Focus Window

Use	Untitled - Notepad	- 🗆 X				
FocusWindow	File Edit Format View Help		Desktop Recorder  Edit Scenario Test Run Files			- 0 ×
to switch			Recording: Disabled  Enabled	Cursor Position: 1127:239	Pixel Color: RGB: 255 255 255	
focus to			Command	Arguments		Flags
another			startStep	stepName: initialStep		rings
			fotusWindow	title: Untitled - Notepad		
window			· · · · · ·			
defined by						
Window title.						
Example:						
switch focus			Add Command			Clear
Switch locus			Command Details			
to the			Command: FocusWindow	Ignore Timing	Save Revent	Delete
			Title	- I ignore many		
Notepad			Untitled - Notepad			
window:						
window.		1				
			1007 - AMERICA			AND CONCEPTION

#### 9.3 Maximize Window

Use MaximizeWindow to maximize the currently-focused application window.

dit Scenario Test Run	Files						
Recording:		Cursor Position:	Pixel Co	lor:			
Disabled O Ena	abled	0:0	RGB:	0	0	0	
Command	Argun	ients		Flag	gs		
startStep	stepNa	me: initialStep					
maximizeWindow							
resizeWindow	width:	400, height: 300					
Add Command						Clear	

#### 9.4 Resize Window

Use ResizeWindow to resize the currently focused application window to a selected WIDTH, and HEIGHT.

Desktop Recorder					-		×
dit Scenario Test Run	Files						
Recording:		Cursor Position:	Pixel Co	lor:			
Disabled O Enabled	oled	0:0	RGB:	0	0	0	
Command	Argu	ments		Fla	gs		
startStep	stepN	ame: initialStep					
maximizeWindow							
resizeWindow	width	: 400, height: 300					
Add Command	width	: 400, height: 300		[		Clear	
	width	: 400, height: 300	Save	R	evert	Clear	te:
Add Command Command Details	width	400, height: 300           Ignore Timing	Save	Re	evert		te:



#### 9.5 Close Window

Use CloseWindow to close the currently focused application window.

👠 Desktop Recorder			-		×
Edit Scenario Test Run Fi	les				
Recording:	Cursor Position:	Pixel Col	or:		
Disabled	ed 0:0	RGB:	0 0	0	
Command	Arguments		Flags		
startStep	stepName: initialStep				
maximizeWindow					
resizeWindow	width: 400, height: 300				
closeWindow					
Add Command				Clear	
Command Details					
Command:		Save	Revert	Dele	te
CloseWindow	🗸 🗌 Ignore Timing				

## 10 Ignore Timing



The Ignore Timing checkbox next to Commands is a flag that can optionally be toggled On/Checked or Off/Unchecked. When this box is checked, the flag is enabled and the duration of the command

(the time it took to execute the command) won't be included in the total duration of the test.

## 11 Open/Save Scenarios

11.1 Save a Scenario

To save your scenario to file, start by selecting the "Files" tab.

Enter a name for your scenario in the input field labeled "Name" (1). You may also add a Description (2). Click "Save as..." (3) to continue.

Browse to a location and click "Save" to save the scenario to file.

dit Scenario Test Run F	iles		
Scenario Details			
Name	1		
kalk1			
Description	2		
Include Global Pause	Timing		
-			
Global Pause Timing (ms)			
1000			
Save as	Open		
Save as	Open		
Save as Recorded Scenario:	Open		
	Open Arguments	Flags	
Recorded Scenario:		Flags	
Recorded Scenario: Command	Arguments	Flags	
Recorded Scenario: Command startStep	Arguments stepName: initialStep	Flags	
Recorded Scenario: Command startStep startApplication	Arguments stepName: initialStep application: C/Windows/System32(calc.exe, name: calc	Flags ignoreTimings	
Recorded Scenario: Command startStep startApplication waitForTitle	Arguments stepName: initialStep application: CrWindows/System32\calc.exe, name: calc tttle: Calculator, timeout: 30000, matchingType: contains		
Recorded Scenario: Command startStep startApplication waitForTitle positionWindow	Arguments stepName: initialStep application: cl/Windows/System32/calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains x: 200, y: 200		
Recorded Scenario: Command startStep startApplication waitForTitle positionWindow leftClickAt	Arguments stepName: initialStep application: C:\Windows\System32\calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains x 200, y: 200 x 252, y: 649		
Recorded Scenario: Command startApplication waitForTitle positionWindow leftClickAt leftClickAt	Arguments stepName: initialStep application: C:\Windows\System32\calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains x 200, yz 200 x 252, y: 649 x 314, y: 647		
Recorded Scenario: Command startApplication waitForTitle positionWindow leftClickAt leftClickAt leftClickAt	Arguments stepName: initialStep application: C:\Windows\System32\calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains x 200, y: 200 x 252, y: 649 x 314, y: 647 x 492, y: 545		
Recorded Scenario: Command startStep startApplication waitForTitle positionWindow leftClickAt leftClickAt leftClickAt type	Arguments stepName: initialStep application: CrWindows/System32\calc.exe, name: calc tttle: Calculator, timeout: 30000, matchingType: contains x 200, y: 200 x 252, y: 649 x 314, y: 647 x 492, y: 545 text: 12		



### 11.2 Open (loading) a

scenario

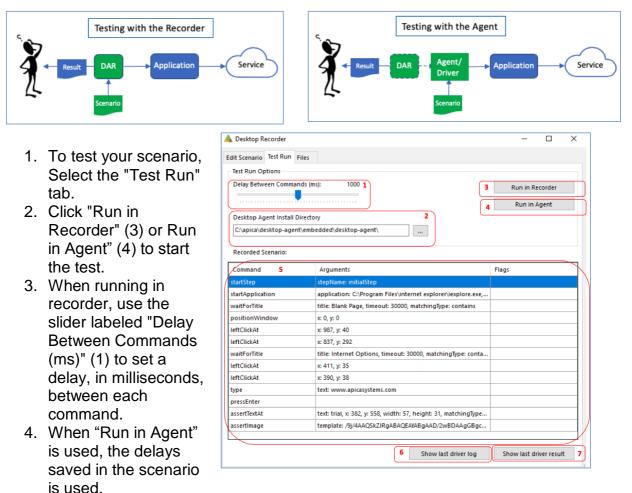
The scenarios are stored in JSON format. You can load a previouslycreated scenario from a file by clicking the "Open" button.

Browse to and select your scenario file and click "Open" to load it into the Desktop Application Recorder.

dit Scenario Test Run	Files		
Scenario Details			
Name			
kalk1			_
Description			
Include Global Pau Global Pause Timing (m 1000	-		
Save as	Open		
Save as Recorded Scenario: Command	Open	Flags	
Recorded Scenario: Command		Flags	
Recorded Scenario: Command startStep	Arguments	Flags	
Recorded Scenario: Command startStep startApplication	Arguments steptiame: initialStep	Flags	
Recorded Scenario: Command startStep startApplication waitForTitle	Arguments steptiame: initialStep application: C\Windows\System32\calc.exe, name: calc	Flags	
Recorded Scenario: Command startStep startApplication waitForTitle positionWindow	Arguments steptiame: initialStep application: C:\Windowr/System32\calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains		
Recorded Scenario: Command startStep startApplication waitForTitle positionWindow leftClickAt	Arguments stepMame: initialStep application: C:\Windows/System32\calc.exe, name: calc litite: Calculator, timeout: 30000, matchingType: contains x: 200, y: 200		
Recorded Scenario: Command startStep startApplication waitForTitle positionVindow leftClickAt leftClickAt	Arguments           stepHame: initialStep           application: C:\Windows\System32\calc.exe, name: calc           title: Calculator, timeout: 30000, matchingType: contains           x 200, y: 200           x 252, y: 649		
Recorded Scenario: Command startStep startApplication waitForTitle positionVindow leftClickAt leftClickAt leftClickAt	Arguments <b>Iteptiane: initialStep</b> application: Ci\Windows\System32\calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains x 200, y: 200 x 252, y: 649 x 314, y: 647		
Recorded Scenario: Command startStep startApplication waitForTitle positionWindow leftClickAt leftClickAt leftClickAt type	Arguments steptiame: initialStep application: C:\Windows/System32\calc.exe, name: calc title: Calculator, timeout: 30000, matchingType: contains x 200, y: 200 x 252, y: 649 x 314, y: 647 x 492, y: 545		
Recorded Scenario:	Arguments         steptiame: initialStep         application: C:\Windowr\System32\calc.exe, name: calc         title: Calculator, timeout: 30000, matchingType: contains         x: 200, y: 200         x: 252, y: 649         x: 314, y: 647         x: 492, y: 545         text: 12	ignoreTimings	

## 12 Test Run a Scenario

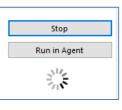
You can test either with the local Desktop Recorder or you can run it on the Agent.





5. While a test is running, a "Stop" button appears, replacing "Run in...". Click this to stop the run.

6.	After a test "Run in Agent", you can view the Result (7	7)
	and the Log (6).	-



Command	Elapsed (milliseconds)	Message
startStep		
pause	1003	
startApplication	13	View
pause	1016	Result
waitForTitle	531	ICeSuit
pause	1001	
positionWindow	15	Driver log for run 7fd85965-4826-48a7- bca1-ed688831afc0 —
pause	1002	2019-02-28 13:14:55.746 INFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command leftClickAt to executable class 2019-02-28 13:14:55.746 INFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command pause to executable class
leftClickAt	45	2019-02-28 13:14:55.762 INFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command leftClickkt to executable class 2019-02-28 13:14:55.777 INFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command pause to executable class
pause	1001	2019-02-28 13:14:55.773 IHFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command type to esecutable class 2019-02-28 13:14:55.783 IHFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command pause to case 2019-02-28 13:14:55.753 IHFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command pause to case
leftClickAt	46	2019-02-28 1314455.793 IHFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command spectra to executable class 2019-02-28 1314455.793 IHFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command assertication text to executable class
pause	1001	2019-02-28 13:14:55.793 INFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command pause to executable class 2019-02-28 13:14:55.808 INFO 3012 [main] c.a.d.desktop.DesktopScenarioFactory : Resolving command assertImage to executable class
waitForTitle	0	2019-02-28 13:14:56.811 INFO 3012 [pool-1-thread-1] c.a.d.desktop.ApplicationsManagerImp1 : Creating application 'IE', binary = C: \\Frogram Files\internet explorer\iexplore.exe 2019-02-28 13:14:56.824 INFO 3012 [pool-1-thread-1] c.a.d.desktop.DesktopApplication : Started application (binary = C:\Frogram
pause	1015	FileSinternet explore/isepiore.exe, args = [], FDD = 5550. 2016-02.27 ±13:50.080 FERD 3012, FODDL=Thread-11.c.a.d desiron Tass
		2019-02-02 13:15:16:15 BEDROS 3012 [pool-i-thread-1] c.a.desktopagent.d 2019-02-02 13:15:15:95 BEDROS 3012 [pool-i-thread-1] c.a.desktopagent.d 2019-02-02 13:15:15:95 BEDROS 3012 [pool-i-thread-1] c.a.desktopagent.d [socre = 17:95] 1 Sage boundaries = java.axt.Rectangle 2019-02-02 13:15:16:10 BURO 3012 [main] c.a.d.driverXpplicatioRNumer completed 2019-02-02 13:15:16:10 BURO 3012 [nain] c.a.d.driverXppleted 2019-02-02 13:15:16:10 BURO 3012

Note that only the following commands will be executed (re-played) when using the "Run in Recorder" option:

#### "Run in Recorder" Only Options

startApplication	stopApplication	pause
positionWindow	focusWindow	maximizeWindow
resizeWindow	closeWindow	leftClickAt
rightClickAt	doubleClickAt	mouseMove
dragTo	type	pressEscape
pressEnter	pressBackspace	pressTab
assertTitle	waitForTitle	assertPixelColor
assertNotPixelColor	waitForPixelColor	waitForNotPixelColor

The "Run in Recorder" option is typically used in the beginning of the scripting, to verify mouse and keyboard commands.

## 13 Command Reference



Group	Command	Description	Arguments	Notes	Recorder Support: recording	Recorder Support: replaying
Applications	startApplication	Start application	application, name, applicati onArguments	application is path to binary. name creates a reference to the started application. applicationA rguments - comma- separated list of arguments as single string (e.g. "- - argumentOn e, argumentTw 0, argumentThr ee=value")	Manually	+
	stopApplication	Quit application	name	name is reference to started application. This command will then close it.	Manually	+
Windows	focusWindow	Switch focus to target window X	title		Manually	
	closeWindow	Close the current window.			Manually	+
	positionWindow	Position current window at location X,Y	х, у		Manually	+
	resizeWindow	Resize current window to WIDTH:HEIGHT	width, height		Manually	+
	maximizeWindow	Maximize the current window.			Manually	+



	assertTitle	Assert that the current window title matches X	title, matchingType (optional, possible values: 'exact' (default)   'glob'   'contains')	Manually	+
	waitForTitle	Wait for current window title to match X	title, timeout (optional, in milliseconds), m atchingType (optional, possible values: 'exact' (default)   'glob'   'contains')	Manually	+
Mouse	moveMouse	Move mouse pointer to coordinates X,Y	х, у	+	+
	dragTo	Hold left mouse and move pointer to X,Y	х, у	+	+
	leftClickAt	Left click at coordinates X,Y	х, у	+	+
	rightClickAt	Right click at coordinates X,Y	х, у	+	+
	doubleClickAt	Double click at coordinates X,Y	х, у	+	+
	scrollUp	Scroll up X notches.	notches	Manually	
	scrollDown	Scroll down X notches.	notches	Manually	
Keyboard	type	Simulate keystrokes for each character in STRING	text	+	+
	pressWithAlt	Press key with alt. Key must be alphanumeric.	key	Manually	
	pressWithCtrl	Press key with ctrl. Key must be alphanumeric.	key	Manually	
	pressEnter	Presses enter.		+	+
	pressBackspace	Presses backspace.		+	+
	pressEscape	Presses escape.		+	+
	pressTab	Presses tab.		+	+



	pressFunctionKey	Presses a function key	key	Key is a number between 1 and 12 and corresponds to F1-F12.		+
	pressArrowKey	Presses an arrow key	direction	direction is one of: up   down   left   right	Manually	
Flow Control	pause	Pause for X ms	duration		Manually	+
Text	assertText	Assert that TEXT is present in active window	text, matchingT ype (optional, possible values: 'exact' (default)   'glob'   'contains')		Manually	
	assertTextAt	Assert TEXT is present within X:Y:WIDTH:HEIGHT	text, x, y, width, height, matchin gType (optional, possible values: 'exact' (default)   'glob'   'contains')		Manually	
	waitForTextAt	Wait until TEXT is present within X:Y:WIDTH:HEIGHT , or until timeout expires	text,x,y,width,h eight,timeout (o ptional, in milliseconds), m atchingType (optional, possible values: 'exact' (default)   'glob'   'contains')			
Image Recognition	assertImage	Assert that the IMAGE (base 64 image data) is present in active window	template, scoreType (optional, possible values: SSD (default)   NCC)		Manually	
	leftClickImage	Locate IMAGE (base 64 image data) on the screen and perform a left click in the center	template, score Type (optional, possible values: SSD (default)   NCC)		Manually	



	rightClickImage	Locate IMAGE (base 64 image data) on the screen and perform a right click in the center	template, score Type (optional, possible values: SSD (default)   NCC)	Manually	
	doubleClickImage	Locate IMAGE (base 64 image data) on the screen and perform a double click in the center	template, score Type (optional, possible values: SSD (default)   NCC)	Manually	
Screenshots	takeScreenshot	Takes a screenshot.		Manually	
Misc.	startStep	Create a new group of commands	stepName	Manually	
Validation	assertPixelColor	Assert that pixel color (RGB value) at coordinates X,Y matches RED,GREEN,BLUE	x, y, red, green, blue	Manually	+
	assertNotPixelColo r	Assert that pixel color (RGB value) at coordinates X,Y does not match RED,GREEN,BLUE	x, y, red, green, blue	Manually	+
	waitForPixelColor	Wait until the pixel color (RGB value) at coordinates X,Y matches RED,GREEN,BLUE	x, y, red, green, blue, timeout	Manually	+
	waitForNotPixelCol or	Wait until the pixel color (RGB value) at coordinates X,Y does not match RED,GREEN,BLUE	x, y, red, green, blue, timeout	Manually	+

## 14 Scenario Format

The format of an exported/saved scenario is JSON, and looks as follows:

```
ł
    "name": "Example Scenario Name",
    "description": "A simple description",
    "global_pause": 1000,
    "include_global_pause_timings": false,
    "variables": {},
    "commands": [
      {
         "command": "startApplication",
         "args":{
            "name": "notepad",
            "application":"C:\\Windows\\system32\\notepad.
exe″
         },
         "ignore_timing":true
      },
        {
            "command": "type",
            "args": {
                "text": "typing!"
            },
            "ignore timing": false
        }
    ]
}
```

A scenario file can be edited manually. Before uploading, Apica recommends testing it in the Desktop Application Recorder, or at least verify the JSON formatting.



# 15 Appendix - Testing a scenario through the command line

(go to the installation directory of the Desktop Agent)

> java -Djava.awt.headless=false -jar driver-0.jar -scenarioId=<name\_of\_scenario> --runId=abc123

This will run the scenario using the Agent, and show the console window during execution, which might be useful for debugging.

## 16 Appendix - Testing a scenario through the API

#### NOTE: 19-05-07 NOT YET SUPPORTED

When the Desktop Agent has been installed as a service (see installation manual), you can command it to execute checks (i.e. dispatching a job) through the API (example: using Postman).

Important: This will initiate the run through the service (which will initiate a new local RDP session etc.). After finishing the job, all RDP sessions will be terminated. You MUST follow the complete installation guide before this will work.

16.1 Start a job

```
(post) http://<host>:8080/job
Headers
Accept:application/json
Content-Type:application/json
Body
{"debug": true, "format_version": 0, "job_timeout": 60, "scenario_id":
"<name-of-scenario>.json", "screenshots": true}
Expected response
{
    "id": "8b2e9be586714bd5896ff10de52ea99e" // The jobId
}
```

Note: If the command results in a 500-error, it may be because the scenario does not exist.

16.2 Get JobStatus

```
(get) http://<host>:8080/job/<jobId>
Expected response
{
    "id": "<jobId>",
    "status": "finished" // or running, failed etc.
}
```

Apica AB | Garvargatan 9 | SE-112 21 Stockholm, SWEDEN | +46 8-400 273 27 http://www.apicasystems.com/ © 2019 Apica | Page 32



#### 16.3 Get a result

(get) http://<host>:8080/result/<jobId>
Header
Accept:application/xml

#### Expected response

{A result in XML format}

