

2013

Automation Samurai Images API

Tutorial on how to implement our API



Automation Samurai presents new API that will enable you to automate anything you see on the screen of your monitor. It is very simple to use and it uses images to automate and test graphical user interfaces. This interface is still experimental .

This is how it works:

- 1) You create a screenshot image of the object that you want to click on or grab a text from it.
- 2) You call an appropriate method from API that will interact with that object on the screen.
- 3) Sometimes when you work with dynamic content you create a region and use it instead of image. All this will be explained in details below. Please keep reading.

The benefits of this approach compare with traditional one is that you don't need to describe a properties of the object to program to identify it, instead you provide image of the object and program will find it and execute actions on it. This approach save a lot of time to developed, it takes much less time to create tests and not required technical expertise to implement.

What kind of application can be automated with this API? The short answer to this question is any application! You can use this API to automate .NET or JAVA application, Flash or Flex and so on.

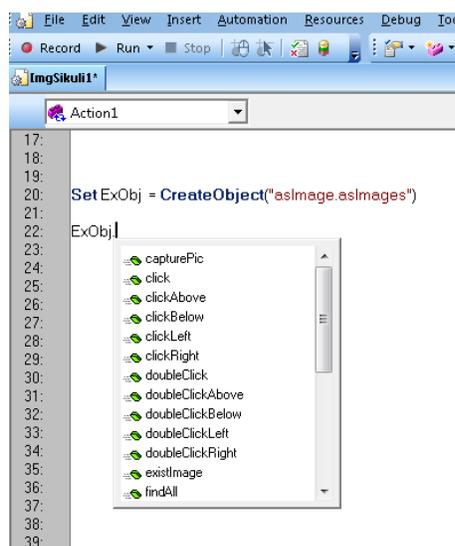
It also possible to build a framework using this API, it can be any of the popular frameworks for example keyword-driven framework and so on.

So let's review some tutorials and examples:

In this tutorial we will walk you through how you can use this API in you QTP tests. First thing that you would need to do is to create object:

```
Set ExObj = CreateObject("asImage.asImages")
```

Once you created object, you can explore all methods that our API has. Please see Picture1.

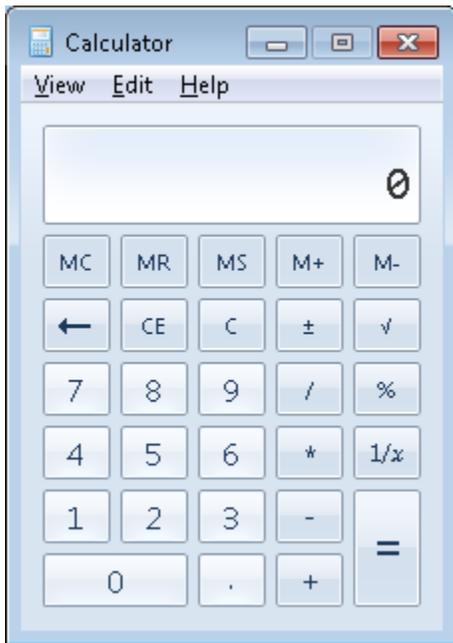


Picture1.

If you would like to interact with any object on your screen you would need to take a screenshot of that object and after taking the screenshot of that object you can click on it, highlight it, type in it, read text from it and so on. Below are API methods:

| Click methods: | Type methods: | Highlight methods: | Get Text methods: | Others methods: |
|--|--|---|---|---|
| <i>Click</i> <i>clickAbove</i> <i>clickBelow</i> <i>clickLeft</i> <i>clickRight</i> <i>doubleClick</i> <i>doubleClickAbove</i> <i>doubleClickBelow</i> <i>doubleClickLeft</i> <i>doubleClickRight</i> | <i>type</i> <i>typeAbove</i> <i>typeBelow</i> <i>typeLeft</i> <i>typeRight</i> <i>KeyUp</i> <i>KeyDown</i> | <i>HighlightImage</i> <i>HighlightText</i> | <i>getTextFromImage</i> <i>getTextFromRegion</i> | <i>findAll</i> <i>capturePic</i> <i>existImage</i> <i>getErrorStatus</i> |

Let's start with the example to explore how it works. Suppose you want to automate Calculator program. (Picture2)



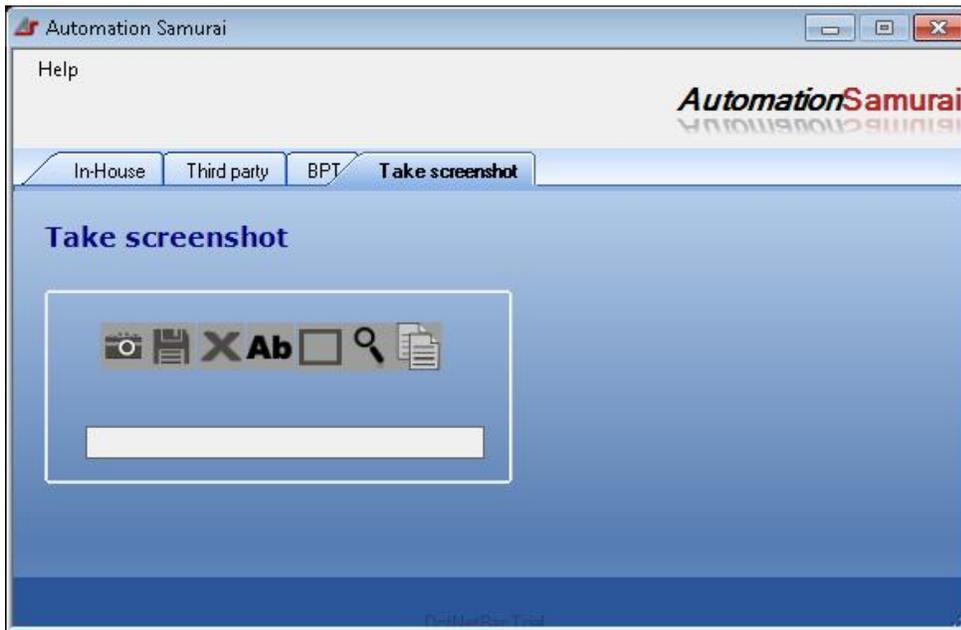
Picture2.

Like we mentioned earlier, we are using Image screenshots to automate applications. So first thing you would need to do is to take all screenshots of the objects that you want to interact with.

Let's verify that 5 + 6 is equal 11.

You can use our UI that we developed to take screenshots. (Picture3)

We will explain it in more details below.



Picture3

For our automation test you would need 4 objects to interact with and that's mean that you would need 4 screenshot images.

First image is button "5" and in the script we will name it as Btn_5.



Second image is button "6" and in the script we will name it Btn_6.



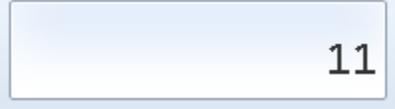
Third Image is button "+" and in script we will name it as Btn_Add.



Fourth Image is button "=" and in the script we will name it as Btn_Equality.



Last one is the display of the calculator where it supposes to display 11 and in the script we will name it as Result.

11

So now when we created all the Images that we need let's create automated script that is verifying that $5+6=11$

```
Dim Btn_5 : Btn_5 = "c:\demo\Btn_5.png" 'Btn_5 is holding path to the image location
Dim Btn_6 : Btn_6 = "c:\demo\Btn_6.png"
Dim Btn_Add : Btn_Add = "c:\demo\Btn_Add.png"
Dim Btn_Equality : Btn_Equality = "c:\demo\Btn_Equality.png"
Dim Result : Result = "c:\demo\Result.png"
```

```
Set ExObj = CreateObject("asImage.asImages")
ExObj.click(Btn_5)
ExObj.click(Btn_Add)
ExObj.click(Btn_6)
ExObj.click(Btn_Equality)
```

'Let's verify that we received the right result

```
If ExObj.existImage(Result)= True then
    MsgBox "PASS"
Else
    MsgBox "FAIL"
End If
```

That's it! We created our first test using Image recognition API.

Sometimes you would have a dynamic content that will prevent you from taking a screenshot Image because it changes depending on the results. Let me show you an example of how to handle dynamic content in your automation test.

Let's use the same script for this task and instead of using last verification lines we will change it into the following:

```
Dim Btn_5 : Btn_5 = "c:\demo\Btn_5.png" 'Btn_5 is holding path to the image location
Dim Btn_6 : Btn_6 = "c:\demo\Btn_6.png"
Dim Btn_Add : Btn_Add = "c:\demo\Btn_Add.png"
Dim Btn_Equality : Btn_Equality = "c:\demo\Btn_Equality.png"
Dim Result : Result = "c:\demo\Result.png"
```

```
Set ExObj = CreateObject("asImage.asImages")
ExObj.click(Btn_5)
ExObj.click(Btn_Add)
ExObj.click(Btn_6)
ExObj.click(Btn_Equality)
```

'Let's verify that we received the right result

```
If ExObj.getTextFromRegion(102,207,195,59) = "11" then
    MsgBox "PASS"
Else
    MsgBox "FAIL"
End If
```

Note, if the text returned from the image is not matched the actual text that is displayed inside Image that's mean that OCR engine that is used inside is not trained to read the text Font used inside that Image and we would need to train our OCR to properly "read" text for that text Font. Please contact our technical support team for more details.

As you can see in our last example we can "read" text from the specified region. (You can also use our UI to easily create a region for your scripts. (Picture3)

Now let's try another example. As you can see we have a picture of 3 identical checkboxes that we would like to click on.

- I have a bike
- I have a car
- I have a bicycle

Below is the script that will click on all of the checkboxes.

First, create screenshot Image of checkbox and save it on hard drive with "checkbox.png" name



Below is the script that highlights and clicks on all checkboxes

'Create Automation Samurai Image Object

```
Set ImgObj = CreateObject("asImage.asImages")
```

'Create Array of identical Images

```
Arr = ImgObj.findAll("C:\checkbox.png")
```

'Iterate through Array and highlight and click on each Image

```
For i =0 to Ubound(Arr)
```

```
    ImgObj.highlightImage(Arr(i))
    ImgObj.click Arr(i)
```

Next

Now let's take a look at our UI and how it can be used to create images for your automation scripts.



Picture4

Toolbar:

button 1: Helps you to take screenshot of the area that you want to run your methods on.

button 2: After you took screenshot you can save it to some location of your choice. Then you can copy file path into your scripts.

button 3: Clears UI from Image and text information

button 4: Get text from Image or Region that you previously created

button 5: Create Region. After Region has been created, you can copy coordinates into your automation scripts.

button 6: Highlights Region or Image on your screen. This helps to make sure that Image or Region is correctly identified by our API.

button 7: Copies coordinates or Image Path located inside Text field into memory. It can be later pasted into your script.

As you can see, this simple UI helps you to create screenshots and Regions that you can use later in your automated scripts.

Those scripts can be developed inside QTP and regular VBScript.

We hope this tutorials helped to understand new features of our Add-in and gave you an idea of how it can help you with automation of your QTP scripts.

Thank you