Selenium Window Handling Concept:

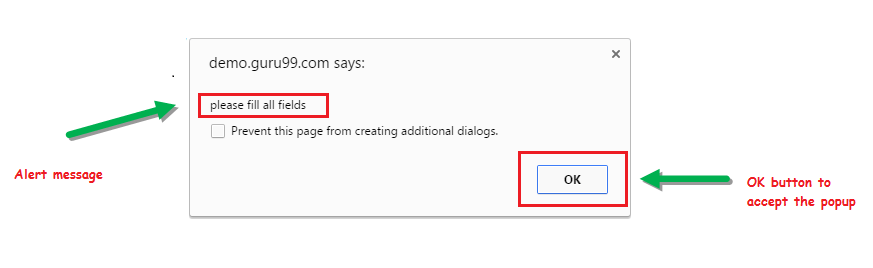
What is Alert?

Alert is a small message box which displays on-screen notification to give the user some kind of information or ask for permission to perform certain kind of operation. It may be also used for warning purpose

Here are few alert types:

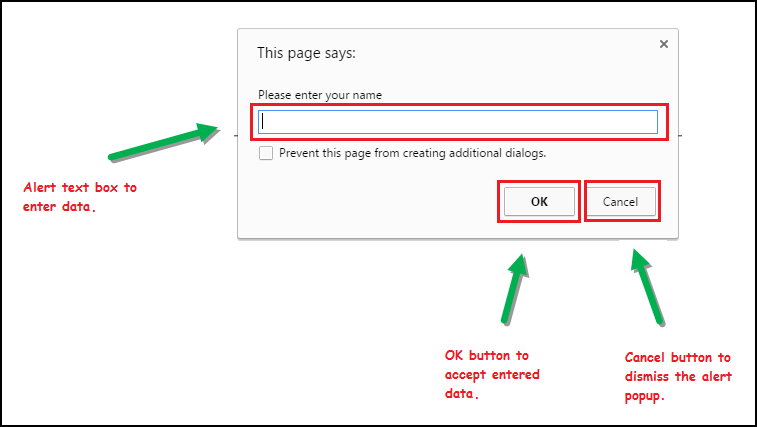
**1) Simple Alert**

This simple alert displays some information or warning on the screen.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph1.png)

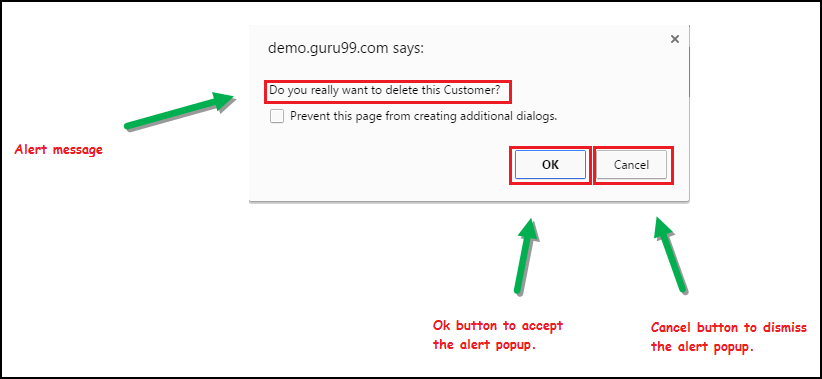
**2) Prompt Alert.**

This Prompt Alert asks some input from the user and selenium webdriver can enter the text using sendkeys(" input…. ").

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph2.png)

**3) Confirmation Alert.**

This confirmation alert asks permission to do some type of operation.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph3.png)

If you want to see the code of each type then follow the below mention site:

**https://www.toolsqa.com/selenium-webdriver/c-sharp/how-to-handle-different-types-of-alert-and-popup-box-in-selenium-csharp/**

**How to handle Alert in Selenium WebDriver**

**There are two types of alerts that we would be focusing on majorly**:

1. Windows-based alert pop-ups
2. Web-based alert pop-ups

**Web-base alerts:**

Alert interface provides the below few methods which are widely used in Selenium Webdriver.

1) void dismiss() **// To click on the 'Cancel' button of the alert.**

driver.switchTo().alert().dismiss();

2) void accept() **// To click on the 'OK' button of the alert.**

driver.switchTo().alert().accept();

3) String getText**() // To capture the alert message.**

driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) **// To send some data to alert box.**

driver.switchTo().alert().sendKeys("Text");

**Window-base alerts:** At times while automating, we get some scenarios, where we need to handle pop ups generated by windows like a print pop up or a browsing window while uploading a file.

Handling these pop-ups have always been a little tricky as we know Selenium is an automation testing tool which supports only web application testing, that means, it doesn’t support windows based applications and window alert is one of them. However Selenium alone can’t help the situation but along with some third-party intervention, this problem can be overcome.

There are several third-party tools available for handling window based pop-ups along with the selenium.

**Type:1**

**\*\* So now let’s handle a window based pop up using Robot class.**

Robot class is a java based utility which emulates the keyboard and mouse actions.

Before moving ahead, let us take a moment to have a look at the application under test (AUT).

**Explanation of Application under Test**

As an application under test, we would be using “gmail.com”. I believe the application doesn’t require any more introductions.

**Scenario to be automated**

1. Launch the web browser and open the application – “gmail.com”
2. Enter valid username and password
3. Click on the sign in button
4. Click on the compose button
5. Click on the attach icon
6. Select the files to be uploaded with the window based pop up.

**WebDriver Code using Robot Class**

Please take a note that for script creation, we would be using “Learning\_Selenium” project created in the former tutorial.

**Step 1**: Create a new java class named as “DemoWindowAlert” under the “Learning\_Selenium” project.  
**Step 2**: Copy and paste the below code in the “DemoWindowAlert.java” class.

Below is the test script that is equivalent to the above-mentioned scenario.

|  |  |
| --- | --- |
| 1 | import java.awt.Robot;</pre> |
| 2 | import java.awt.event.KeyEvent; | |

|  |  |
| --- | --- |
| 3 | import org.junit.After; |
| 4 | import org.junit.Before; | |

|  |  |
| --- | --- |
| 5 | import org.junit.Test; |
| 6 | import org.openqa.selenium.By; | |

|  |  |
| --- | --- |
| 7 | import org.openqa.selenium.WebDriver; |
| 8 | import org.openqa.selenium.firefox.FirefoxDriver; | |

|  |  |  |
| --- | --- | --- |
| 9 |  | |
| 10 | | public class DemoWindowAlert { | |

|  |  |  |
| --- | --- | --- |
| 11 | WebDriver driver; | |
| 12 | @Before |

|  |  |
| --- | --- |
| 13 |  |
| 14 | public void setUp() | |

|  |  |
| --- | --- |
| 15 | { |
| 16 | driver=new FirefoxDriver(); | |

|  |  |
| --- | --- |
| 17 | driver.get("https://gmail.com"); |
| 18 | driver.manage().window().maximize(); | |

|  |  |  |
| --- | --- | --- |
| 19 | } | |
| 20 |  |

|  |  |
| --- | --- |
| 21 | @Test |
| 22 | public void testWindowAlert() throws Exception{ | |

|  |  |
| --- | --- |
| 23 |  |
| 24 | // enter a valid email address | |

|  |  |  |
| --- | --- | --- |
| 25 | driver.findElement(By.id("Email")).sendKeys("TestSelenium1607@gmail.com"); | |
| 26 |  |

|  |  |
| --- | --- |
| 27 | // enter a valid password |
| 28 | driver.findElement(By.id("Passwd")).sendKeys("TestSelenium"); | |

|  |  |
| --- | --- |
| 29 |  |
| 30 | // click on sign in button | |

|  |  |  |
| --- | --- | --- |
| 31 | driver.findElement(By.id("signIn")).click(); | |
| 32 | Thread.sleep(30000); |

|  |  |
| --- | --- |
| 33 |  |
| 34 | // click on compose button | |

|  |  |  |  |
| --- | --- | --- | --- |
| 35 | driver.findElement(By.xpath("//div[@class='z0']//div[contains(text(),'COMPOSE')]")).click(); | | |
| 36 | |  |

|  |  |  |
| --- | --- | --- |
| 37 | | // click on attach files icon |
| 38 | driver.findElement(By.xpath("//div[contains(@command,'Files')]//div[contains(@class,'aaA')]")).click(); | | |

|  |  |
| --- | --- |
| 39 |  |
| 40 | // creating instance of Robot class (A java based utility) | |

|  |  |  |
| --- | --- | --- |
| 41 | Robot rb =new Robot(); | |
| 42 |  |

|  |  |  |
| --- | --- | --- |
| 43 | // pressing keys with the help of keyPress and keyRelease events | |
| 44 | rb.keyPress(KeyEvent.VK\_D); |

|  |  |  |
| --- | --- | --- |
| 45 | rb.keyRelease(KeyEvent.VK\_D); | |
| 46 | Thread.sleep(2000); |

|  |  |
| --- | --- |
| 47 |  |
| 48 | rb.keyPress(KeyEvent.VK\_SHIFT); | |

|  |  |
| --- | --- |
| 49 | rb.keyPress(KeyEvent.VK\_SEMICOLON); |
| 50 | rb.keyRelease(KeyEvent.VK\_SEMICOLON); | |

|  |  |  |
| --- | --- | --- |
| 51 | rb.keyRelease(KeyEvent.VK\_SHIFT); | |
| 52 |  |

|  |  |
| --- | --- |
| 53 | rb.keyPress(KeyEvent.VK\_BACK\_SLASH); |
| 54 | rb.keyRelease(KeyEvent.VK\_BACK\_SLASH); | |

|  |  |  |
| --- | --- | --- |
| 55 | Thread.sleep(2000); | |
| 56 |  |

|  |  |
| --- | --- |
| 57 | rb.keyPress(KeyEvent.VK\_P); |
| 58 | rb.keyRelease(KeyEvent.VK\_P); | |

|  |  |
| --- | --- |
| 59 |  |
| 60 | rb.keyPress(KeyEvent.VK\_I); | |

|  |  |  |
| --- | --- | --- |
| 61 | rb.keyRelease(KeyEvent.VK\_I); | |
| 62 |  |

|  |  |
| --- | --- |
| 63 | rb.keyPress(KeyEvent.VK\_C); |
| 64 | rb.keyRelease(KeyEvent.VK\_C); | |

|  |  |  |
| --- | --- | --- |
| 65 | Thread.sleep(2000); | |
| 66 |  |

|  |  |
| --- | --- |
| 67 | rb.keyPress(KeyEvent.VK\_ENTER); |
| 68 | rb.keyRelease(KeyEvent.VK\_ENTER); | |

|  |  |  |
| --- | --- | --- |
| 69 | Thread.sleep(2000); | |
| 70 | } |

|  |  |
| --- | --- |
| 71 |  |
| 72 | @After | |

|  |  |  |
| --- | --- | --- |
| 73 | public void tearDown() | |
| 74 | { |

|  |  |  |
| --- | --- | --- |
| 75 | driver.quit(); | |
| 76 | } |

|  |  |
| --- | --- |
| 77 | } |

**Type:2**

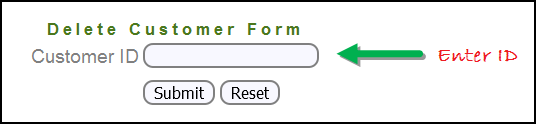
**Now we automate the given below scenario.** (this is for web base alert)

(this example is just for understanding the window popup and alert handling concept, it might be fail while manual execution because of invalid URL,)

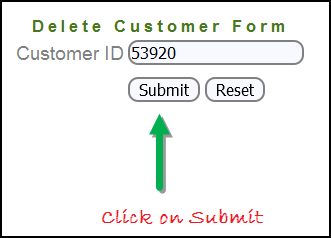
In this scenario, we will use to illustrate Selenium Alert handling.

**Step 1)** Launch the web browser and open the site "http://Actitime.com/ "

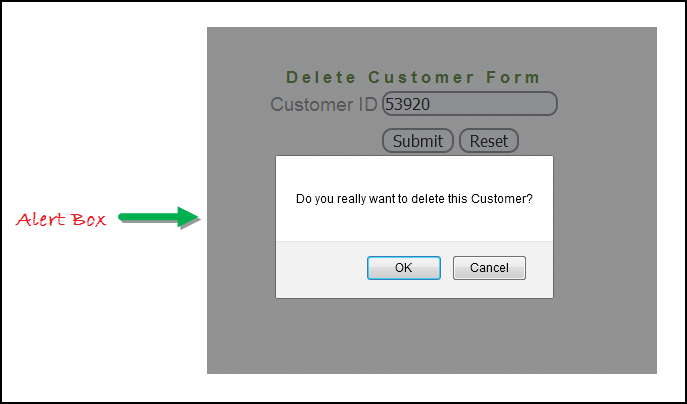
**Step 2)** Enter Any Customer id.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph8.png)

**Step 3)** After entering the customer ID, Click on the "Submit" button.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph9.png)

**Step 4)** Reject/accept the alert.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph10.png)

**Handling Alert in Selenium Webdriver using above scenario**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.NoAlertPresentException;

import org.openqa.selenium.Alert;

public class AlertDemo {

public static void main(String[] args) throws NoAlertPresentException,InterruptedException {

System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Alert Message handling

driver.get("http://Actitime.com/ ");

driver.findElement(By.name("cusid")).sendKeys("53920");

driver.findElement(By.name("submit")).submit();

// Switching to Alert

Alert alert = driver.switchTo().alert();

// Capturing alert message.

String alertMessage= driver.switchTo().alert().getText();

// Displaying alert message

System.out.println(alertMessage);

Thread.sleep(5000);

// Accepting alert

alert.accept();

}

}

**Output of this code :**

When you execute the above code, it launches the site. Try to delete Customer ID by handling confirmation alert that displays on the screen, and thereby deleting customer id from the application.

## How to handle Selenium Pop-up window using Webdriver

In automation, when we have multiple windows in any web application, the activity may need to switch control among several windows from one to other in order to complete the operation. After completion of the operation, it has to return to the main window i.e. parent window.

In selenium web driver there are methods through which we can handle multiple windows.

**Driver.getWindowHandles();**

To handle all opened windows by web driver, we can use "Driver.getWindowHandles()" and then we can switch window from one window to another in a web application. Its return type is Iterator<String>.

**Driver.getWindowHandle();**

When the site opens, we need to handle the main window by **driver.getWindowHandle()**. This will handle the current window that uniquely identifies it within this driver instance. Its return type is String.

To explain it we can consider guru99 site example:-

To handle multiple windows in Selenium WebDriver, We follow the following steps.

Now, we will automate the given below scenario to see how to handle multiple windows using Selenium Webdriver.

In this scenario, we will use "Guru99" demo site to illustrate window handling.

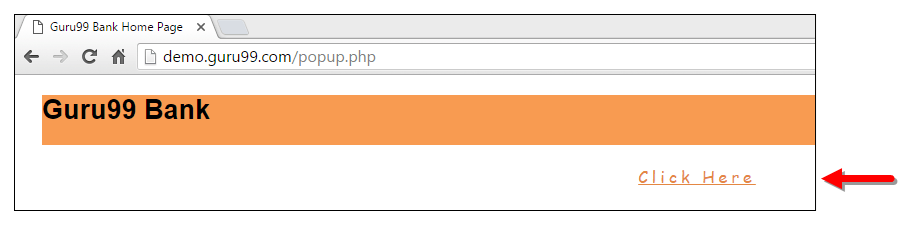
**Step 1)** Launch the site.

Launch the browser and open the site " **http://demo.guru99.com/popup.php** "

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph11.png)

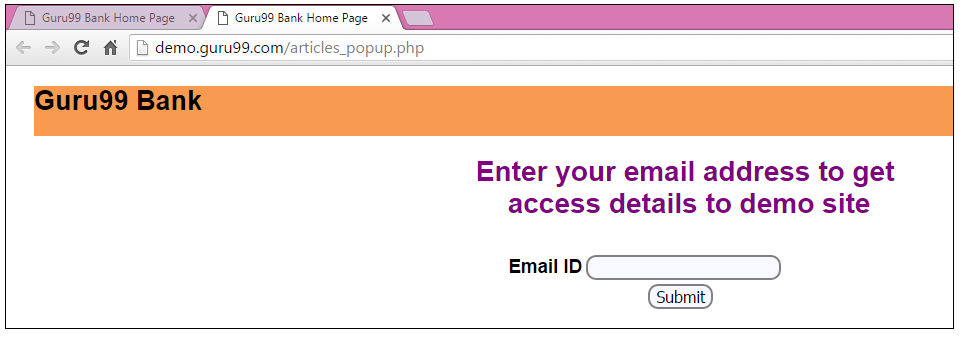
**Step 2)** Click on link "Click Here ".

When the user clicks on the " Click Here " link, new child window opens.

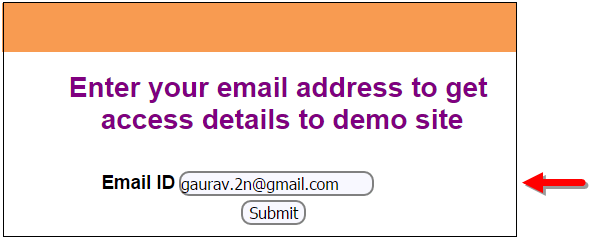
[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph12.png)

**Step 3)** New Child Window opens.

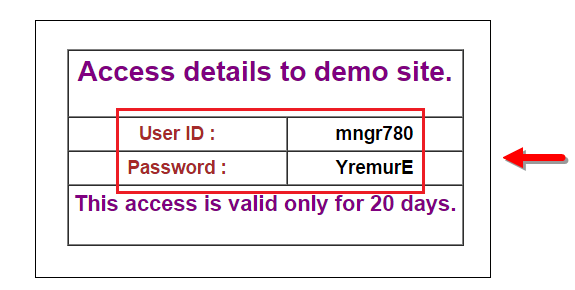
A new window opens, ask the user to enter email id and submit the page.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph13.png)

**Step 4)** Enter your email ID and submit.

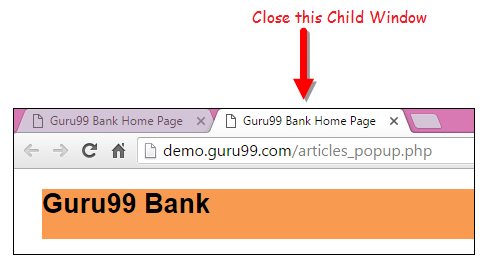
[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph14.png)

**Step 5)** Display the Access Credentials on submitting the page.

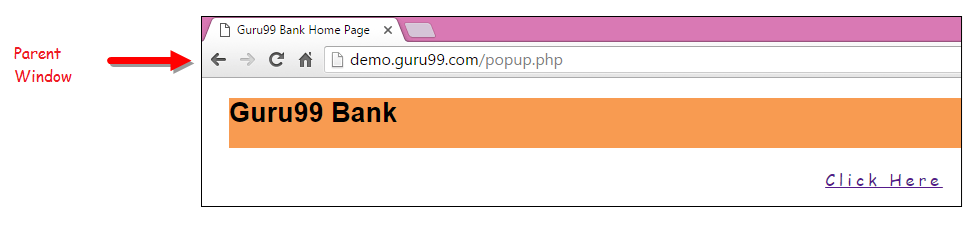
[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph15.png)

When you execute the code, you will see the child window is open in new tab.

1. Close the Child window on which credentials are displayed.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph16.png)

1. Switch to the parent window.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph17.png)

**Handling multiple windows in selenium webdriver using above scenario.**

import java.util.Iterator;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class WindowHandle\_Demo {

public static void main(String[] args) throws InterruptedException {

WebDriver driver=new FirefoxDriver();

**//Launching the site.**

driver.get("http://demo.guru99.com/popup.php");

driver.manage().window().maximize();

driver.findElement(By.xpath("//\*[contains(@href,'popup.php')]")).click();

**// we need to store the parent window to come on it at the end**

String MainWindow=driver.getWindowHandle();

**// To handle all new opened window.**

**//we are using set<String> to avoid duplicate windows**

Set<String> s1=driver.getWindowHandles();

**// Before you can access a collection through an iterator, you must obtain one. Each of the collection classes provides an iterator( ) method that returns an iterator to the start of the collection. By using this iterator object, you can access each element in the collection, one element at a time.**

**In general, to use an iterator to cycle through the contents of a collection,**

Iterator<String> i1=s1.iterator();

**//hasNext() is present in iterator interface which tells us either element is present or not**

while(i1.hasNext())

{

String ChildWindow=i1.next();

if(!MainWindow.equalsIgnoreCase(ChildWindow))

{

**// Switching to Child window**

driver.switchTo().window(ChildWindow);

driver.findElement(By.name("emailid"))

.sendKeys("[gaurav.3n@gmail.com](mailto:gaurav.3n@gmail.com)");

driver.findElement(By.name("btnLogin")).click();

// Closing the Child Window.

driver.close();

}

}

**// Switching to Parent window i.e Main Window.**

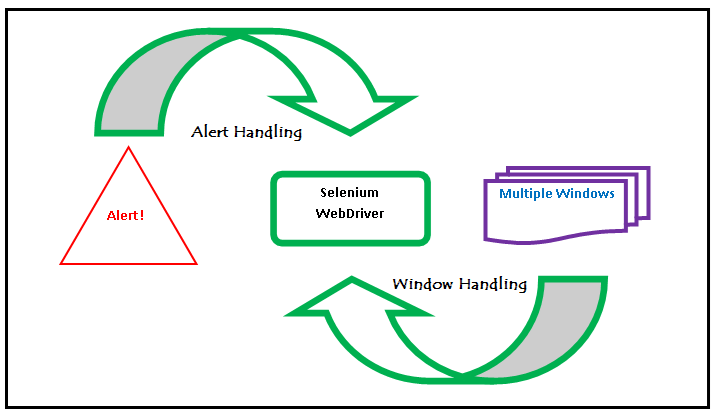
driver.switchTo().window(MainWindow);

}

}

**Output:**

When you execute the above code, it launches the site and on clicking the link "Click here," it opens up a child window in a new tab. You can close the child window, and switch to the parent window once the operation is completely done. Hence handling more than one window in the application.

[](https://www.guru99.com/images/3-2016/032216_1314_AlertPopuph18.png)

**Conclusion:**

* We defined the types of alert and shown them with a screen shot.
* Demonstrated handling the Alert with Selenium WebDriver using particular scenario.
* Handled multiple windows with Selenium WebDriver using particular scenario.