Market Logs:

Parminder-[parminder@enterprisesolutioninc.com](mailto:parminder@enterprisesolutioninc.com)

Faiz-[faiz.khan@bravensinc.com](mailto:faiz.khan@bravensinc.com)

Victor-[victor@platotechinc.com](mailto:victor@platotechinc.com)

Page Object framework:

The chief problem with script maintenance is that if 10 different scripts are using the same page element, with any change in that element, you need to change all 10 scripts. This is time consuming and error prone.

A better approach to script maintenance is to create a separate class file which would find web elements, fill them or verify them. This class can be reused in all the scripts using that element. In future, if there is a change in the web element, we need to make the change in just 1 class file and not 10 different scripts.

This approach is called **Page Object Model(POM)**. It helps make the code **more readable, maintainable**, and **reusable.**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class SiteLogin {

WebDriver driver;

By userSiteName = By.name("uid");

By passwordSite = By.name("password");

By titleText =By.className("barone");

By login = By.name("btnLogin");

public SiteLogin(WebDriver driver){

this.driver = driver;

}

//Set user name in textbox

public void setUserName(String strUserName){

driver.findElement(userSiteName).sendKeys(erName);;

}

//Set password in password textbox

public void setPassword(String strPassword){

driver.findElement(passwordSite).sendKeys(strPassword);

}

//Click on login button

public void clickLogin(){

driver.findElement(login).click();

}

//Get the title of Login Page

public String getLoginTitle(){

return driver.findElement(titleText).getText();

}

/\*\*

\* This POM method will be exposed in test case to login in the application

\* @param strUserName

\* @param strPasword

\* @return

\*/

public void loginToSite (String strUserName,String strPasword){

//Fill user name

this.setUserName(strUserName);

//Fill password

this.setPassword(strPasword);

//Click Login button

this.clickLogin();

}

}

Handling frames : We can use Robot framework for handling different frames.

Robot rb = new Robot();

Synchronized(rb)

{

Rb.notifyAll();

}

Obj.switchTo().alert().accept();

Rb.mousemove(780,160);

Rb.KeyPress(KeyEvent.vk\_ENTER);

}

Basic ANT commands:

ANT clean

ANT built

ANT compile

ANT run

Invoke ANT – This is used in Jenkins