**Creating Object Repository in Selenium WebDriver: XML & Properties file**

**What is an Object Repository?**

An object repository is a common storage location for all objects. In Selenium WebDriver context, objects would typically be the locators used to uniquely identify web elements.

The major advantage of using object repository is the segregation of objects from test cases. If the locator value of one webelement changes, only the object repository needs to be changed rather than making changes in all test cases in which the locator has been used. Maintaining an object repository increases the modularity of framework implementation.

In this tutorial, you will learn-

* [What is an Object Repository?](https://www.guru99.com/object-repository-selenium.html#1)
* [Types of Object Repositories in Selenium Web Driver](https://www.guru99.com/object-repository-selenium.html#2)
* [Selenium Web Driver Object repository using Properties file](https://www.guru99.com/object-repository-selenium.html#3)
* [Selenium WebDriver Object Repository Using XML File](https://www.guru99.com/object-repository-selenium.html#4)

**Types of Object Repositories in Selenium Web Driver**

Selenium WebDriver does not offer an in-built object repository by default. However, object repositories can be built using the key-value pair approach wherein the key refers to the name given to the object and value refers to the properties used to uniquely identify an object within the web page.

The following are the types of object repositories that can be created in Selenium WebDriver.

1. Object Repository using Properties file
2. Object Repository using XML file

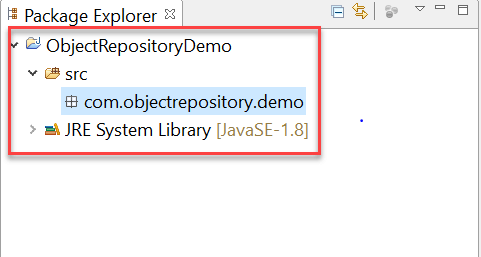
**Selenium Web Driver Object repository using Properties file**

In this approach, properties file is a text file wherein data is stored on the form of key-value pairs. The below tutorial will address the following topics.

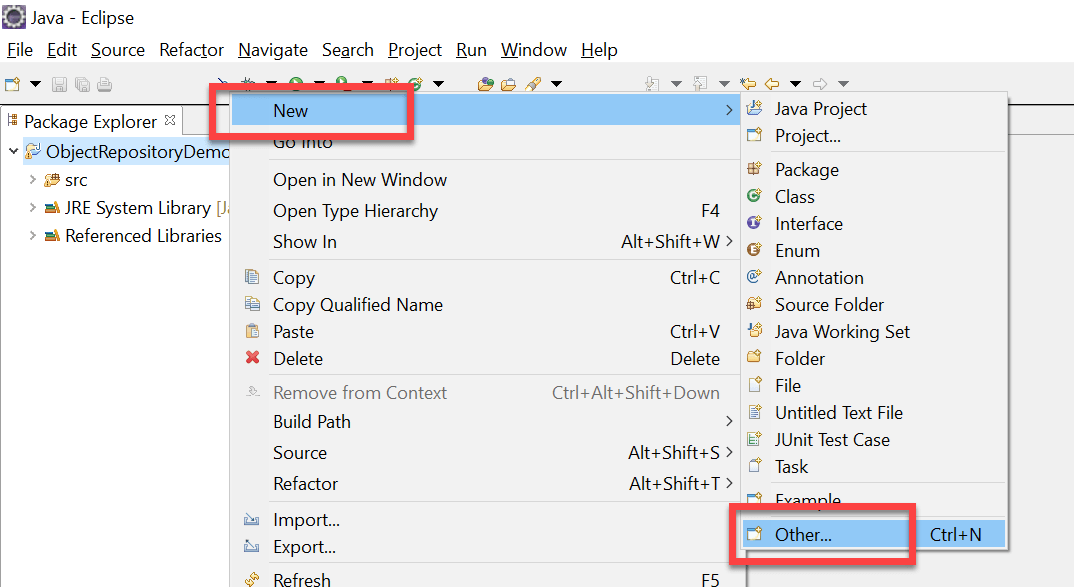
* [Creating a properties file in eclipse](https://www.guru99.com/object-repository-selenium.html#11)
* [Storing data onto properties file](https://www.guru99.com/object-repository-selenium.html#12)
* [Reading data from properties file](https://www.guru99.com/object-repository-selenium.html#13)
* [Using properties file in test scripts](https://www.guru99.com/object-repository-selenium.html#14)

**Step 1) Creating a properties file in eclipse**

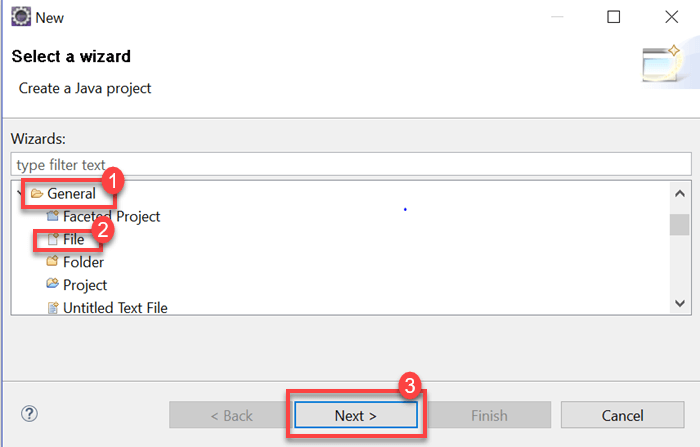
1. To start with, the below java project structure needs to be created in eclipse. Project name and package name can be any valid names.

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj1.png)

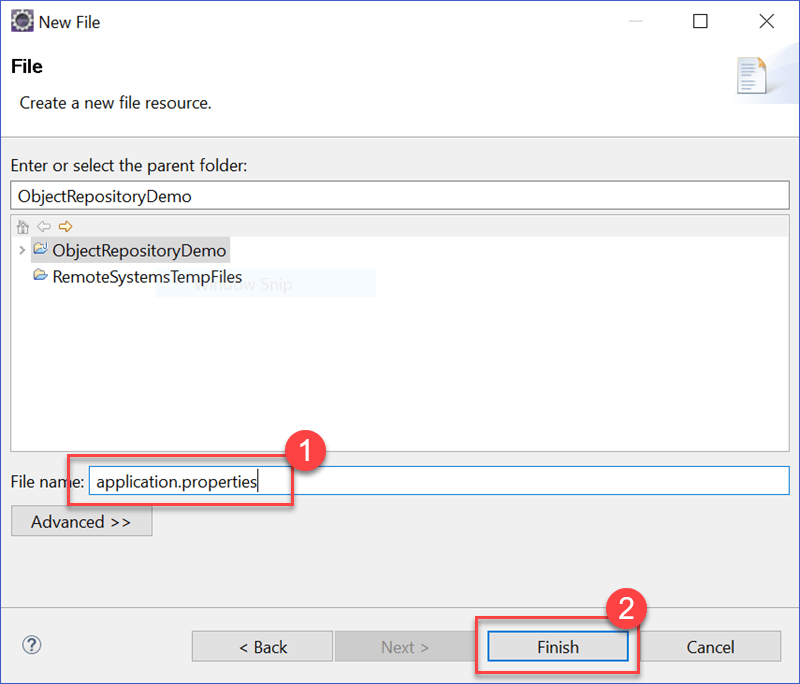
1. Right-click on the main project folder and Select New-> Other

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj2.png)

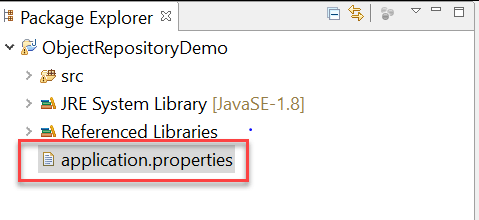
1. In the next window, select General -> File and click on 'Next' button

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj3.png)

1. Provide a valid file name with the extension '.properties' on the new file resource window and click on 'Finish' button

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj4.png)

1. A file named 'application.properties' must be displayed on Project Structure

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj5.png)

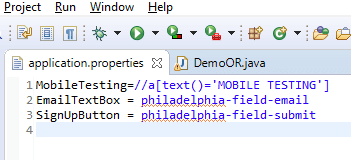
**Step 2) Storing data onto properties file**

1. Data is stored in properties file in the form of key-value pairs, with the key being unique across the file.
2. We will try to use the properties file to identify webelements using locator values.
3. Open application.properties file in Eclipse and store the following data

MobileTesting=//a[text()='MOBILE TESTING']

EmailTextBox = philadelphia-field-email

SignUpButton = philadelphia-field-submit

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj6.png)

4) For this tutorial, the following demo website is being used: <http://demo.guru99.com/selenium/guru99home/>. Here is Test scenario:

* Click on Mobile Testing link using XPATH
* Navigate back
* Enter data onto email textbox using ID
* Click on the Sign Up button using ID

**Step 3) Reading data from properties file**

1. Reading data from properties file can be done using the built-in Properties class provided in java.util package.
2. Initially, an object of Properties class need to be created as below

Properties obj = new Properties();

1. We need to create an object of FileInputStream class with the path to properties file

FileInputStream objfile = new FileInputStream(System.getProperty("user.dir")+"\\application.properties");

1. Reading data from properties file can be done using load method offered by Properties class in java. The below code demonstrates the usage of load method.

Properties obj = new Properties();

FileInputStream objfile = new FileInputStream(System.getProperty("user.dir")+"\\application.properties");

obj.load(objfile);

String mobileTesting = obj.getProperty("MobileTesting");

The string 'mobileTesting' will contain the XPATH to identify the Mobile Testing link within the webpage.

**Step 4) Using properties file in test scripts**

Properties file can be used in test scripts by reading data from a properties file and passing the data as a parameter to the findElement method. The below code demonstrates the usage of data read from properties file in test scripts.

driver.findElement(By.xpath(obj.getProperty("MobileTesting"))).click();

driver.findElement(By.id(obj.getProperty("EmailTextBox"))).sendKeys("[testguru99@gmail.com](mailto:testguru99@gmail.com)");

driver.findElement(By.id(obj.getProperty("SignUpButton"))).click();

The below is the complete code used for the above test scenario.

package com.objectrepository.demo;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Properties;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class DemoOR {

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

// Create WebDriver Instance

WebDriver driver;

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

driver = new ChromeDriver();

driver.get("http://demo.guru99.com/selenium/guru99home/");

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

// Load the properties File

Properties obj = new Properties();

FileInputStream objfile = new FileInputStream(System.getProperty("user.dir")+"\\application.properties");

obj.load(objfile);

// Nagigate to link Mobile Testing and Back

driver.findElement(By.xpath(obj.getProperty("MobileTesting"))).click();

driver.navigate().back();

// Enter Data into Form

driver.findElement(By.id(obj.getProperty("EmailTextBox"))).sendKeys("[testguru99@gmail.com](mailto:testguru99@gmail.com)");

driver.findElement(By.id(obj.getProperty("SignUpButton"))).click();

}

}

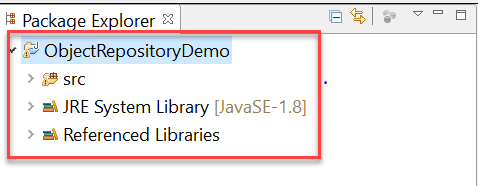
**Selenium WebDriver Object Repository Using XML File**

XML stands for Extensible Markup Language. An XML File uses Document Object Model(DOM) as the basic structure. XML File format will replicate the HTML format upon which the webpage is constructed. Below is the list of topics that will be covered.

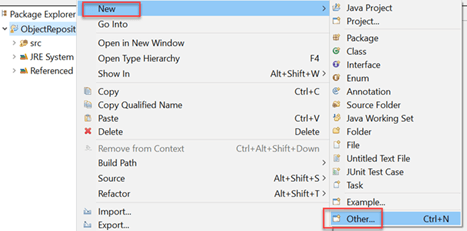
* [Creating an XML file in eclipse](https://www.guru99.com/object-repository-selenium.html#21)
* [Storing data onto XML file](https://www.guru99.com/object-repository-selenium.html#22)
* [Reading data from XML file](https://www.guru99.com/object-repository-selenium.html#23)
* [Using XML file in test scripts](https://www.guru99.com/object-repository-selenium.html#24)

**Step 1) Creating an XML file in eclipse**

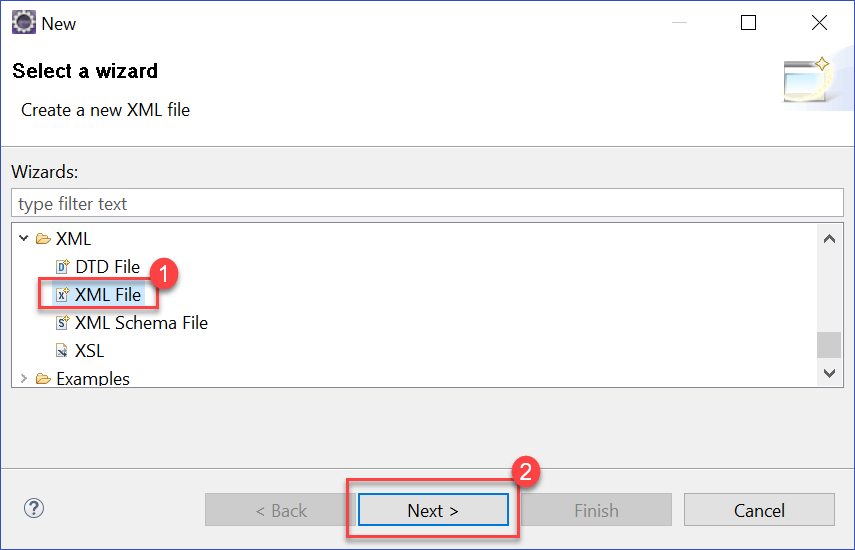
1. The below java project structure needs to be created in Eclipse.

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj7.png)

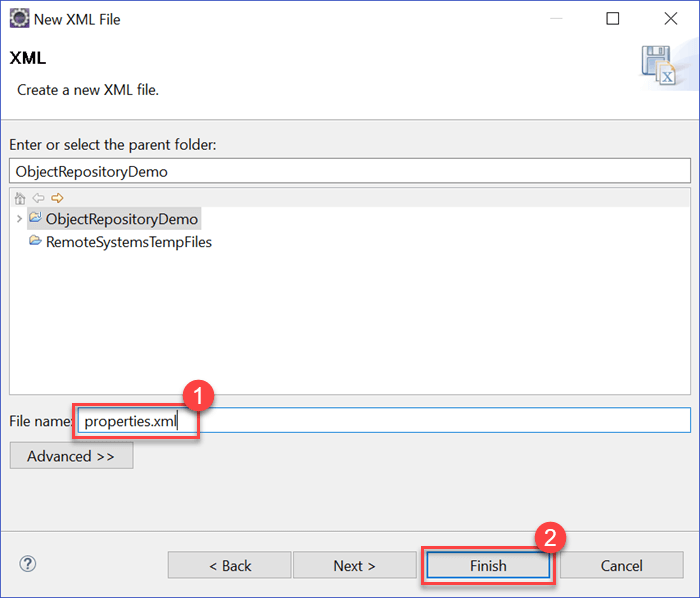
1. Right-click on the project folder, select New -> Other

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj8.png)

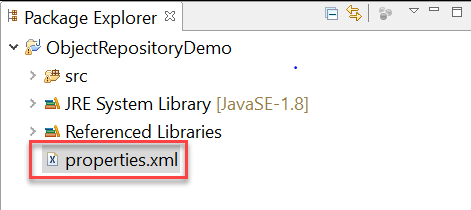
1. Select the XML File within the XML folder and click on 'Next' button

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj9.png)

1. Enter a valid XML File name and click on 'Finish' button

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj10.png)

1. An XML file will be added to the project folder as shown below

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj11.png)

**Step 2) Storing data onto XML file**

Data can be stored in XML file in the form of Document Object Model (DOM). For simplicity sake, we can use the below test scenario as an example.

* Click on Mobile Testing link using XPATH
* Navigate Back to Home page
* Enter data onto email textbox using ID
* Click on the Sign Up button using ID

The below is the format of XML File to be used.

<menu>

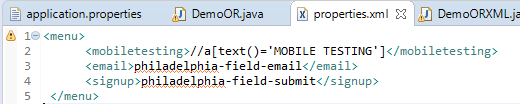
<mobiletesting>//a[text()='MOBILE TESTING']</mobiletesting>

<email> philadelphia-field-email</email>

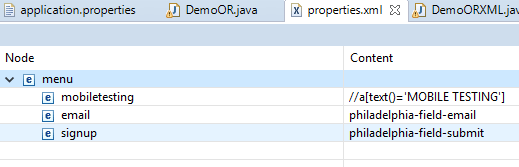
<signup> philadelphia-field-submit </signup>

</menu>

Store the above XML code in properties.xml

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj12.png)

In the design tab you will see

[](https://cdn.guru99.com/images/1/111617_0438_CreatingObj13.png)

**Step 3) Reading data from XML file**

1. Reading data from XML file can be accomplished using the built-in 'dom4j' class in java. Please note that you need to add the below JAR files into the buildpath of your project before proceeding with the code.

* jaxen.jar
* dom4j-1.6.jar

2. Below is the code to read data from XML file.

File inputFile = new File(System.getProperty("user.dir") +"\\properties.xml");

SAXReader saxReader = new SAXReader();

Document document = saxReader.read(inputFile);

String mobileTesting = document.selectSingleNode("//menu/mobiletesting").getText();

String emailTextBox = document.selectSingleNode("//menu/email").getText();

String signUpButton = document.selectSingleNode("//menu/signup").getText();

3. Initially, we need to create a File object and pass it as a parameter to the 'read' method of SAXReader class. Once the XML file data is read successfully, we can access individual nodes of XML document using the 'selectSingleNode' method.

**Step 4) Using XML file in test scripts**

XML file can be used in test scripts by reading data from XML file and passing the data as parameter to the findElement method. The below code demonstrates the usage of data read from XML file in test scripts.

driver.findElement(By.xpath(mobileTesting)).click();

driver.findElement(By.id(emailTextBox)).sendKeys("[testguru99@gmail.com](mailto:testguru99@gmail.com)");

driver.findElement(By.id(signUpButton)).click();

The below code demonstrates the use of XML file in selenium WebDriver

package com.objectrepository.demo;

import java.io.\*;

import java.util.\*;

import org.dom4j.\*;

import org.dom4j.io.SAXReader;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class DemoORXML {

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

// Creating WebDriver Instance

WebDriver driver;

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

driver = new ChromeDriver();

driver.get("http://demo.guru99.com/selenium/guru99home/");

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

// Reading XML File

File inputFile = new File(System.getProperty("user.dir") +"\\properties.xml");

SAXReader saxReader = new SAXReader();

Document document = saxReader.read(inputFile);

String mobileTesting = document.selectSingleNode("//menu/mobiletesting").getText();

String emailTextBox = document.selectSingleNode("//menu/email").getText();

String signUpButton = document.selectSingleNode("//menu/signup").getText();

//Navigating to Mobile Testing and back

driver.findElement(By.xpath(mobileTesting)).click();

driver.navigate().back();

//Entering Form Data

driver.findElement(By.id(emailTextBox)).sendKeys("[testguru99@gmail.com](mailto:testguru99@gmail.com)");

driver.findElement(By.id(signUpButton)).click();

}

}

[**Download the WebDriver Eclipse Project**](https://drive.google.com/uc?export=download&id=1IweUfq7atfRYYGSVoR6N6o1bO0GtT46w)

**Summary:**

* An object repository is a common storage location for all objects
* Selenium WebDriver does not offer an in-built object repository by default
* You can create 2 Types of Object Repository in Selenium
  1. Object Repository using Properties file
  2. Object Repository using XML file
* The properties file is a text file wherein data is stored on the form of key-value pairs
* XML File format will replicate the HTML format upon which the webpage is constructed.