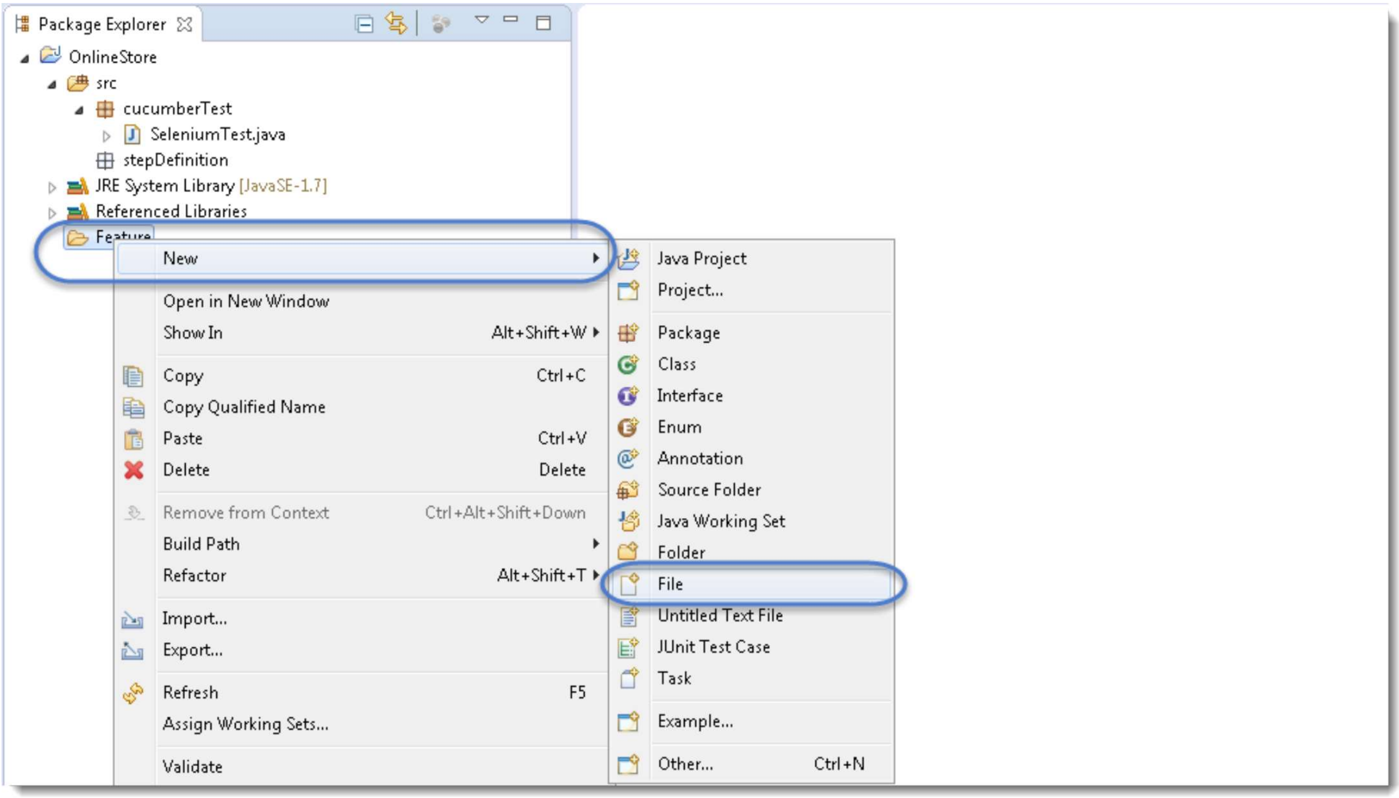
Cucumber proposes to write scenario in the ***Given/When/Then*** format. In the last chapter of ***Cucumber Selenium Java*** test we decided on the LogIn scenario on ***Store.DemoQA.com***. In this chapter we will write a test in Cucumber Format (*Feature File*).

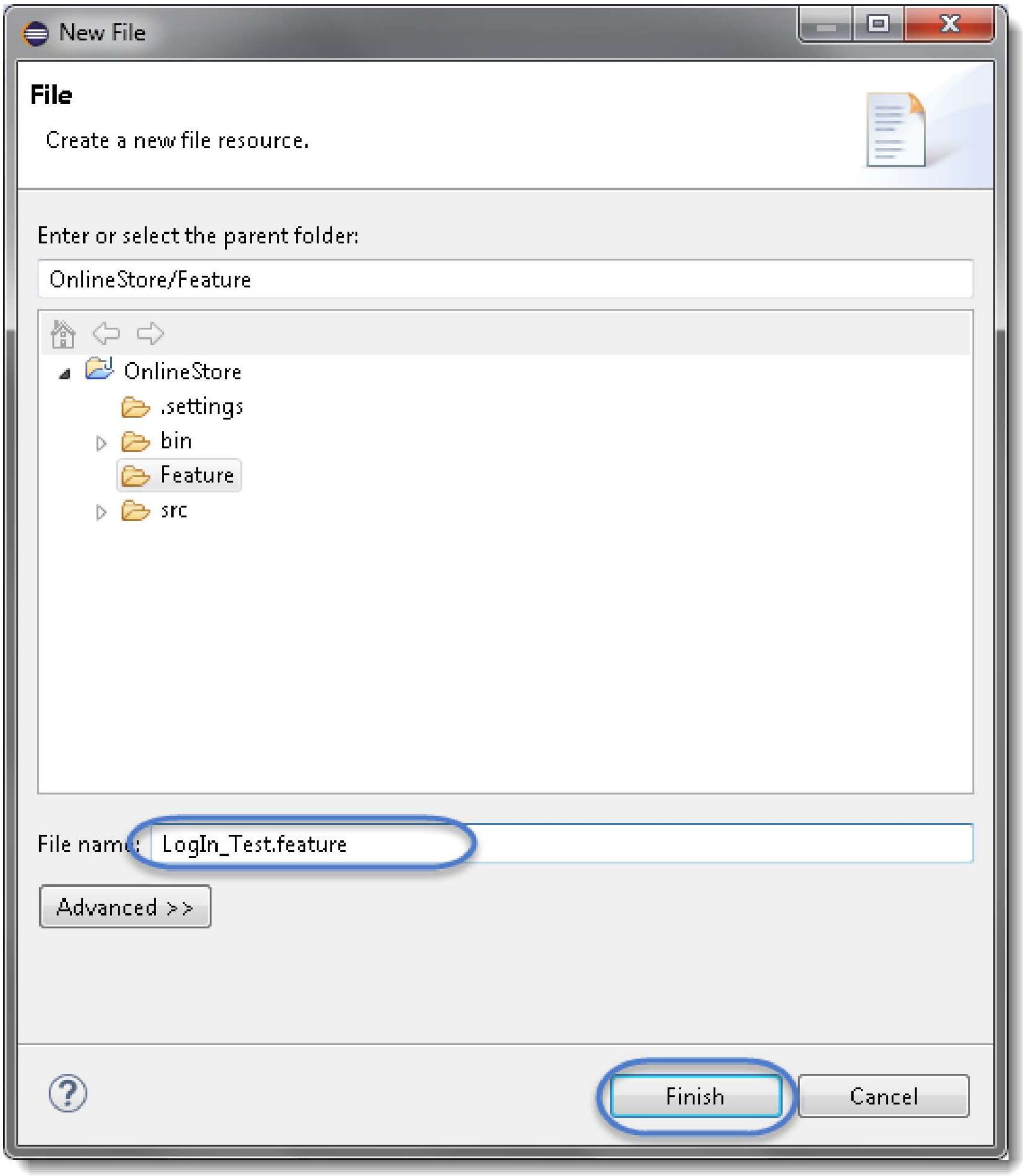
What is Cucumber Feature File?

A ***Feature File*** is an entry point to the Cucumber tests. This is a file where you will describe your tests in Descriptive language (*Like English*). It is an essential part of Cucumber, as it serves as an automation test script as well as live documents. A feature file can contain a scenario or can contain many scenarios in a single feature file but it usually contains a list of scenarios. Let's create one such file.

- On the ***Feature*** folder Right-click and select ***New > File***



- In order for Cucumber to automatically detect the stories (*or features, as they're known in Cucumber*), you need to make sure that they carry the '*feature*' file extension. For example, in this case, I've named my user story '*LogIn_Test.feature*'. Every '*feature*' file conventionally consists of a single feature.



Note*: In case you get a pop up from Eclipse which suggests you to install the better Editor for BDD files, please go ahead and install that. At the bottom of the chapter, steps to install the better editor is given*.

. Write the first cucumber script. In BDD terms the scenario would look like the following.

Cucumber Test Script

```
Feature: Login Action

Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters UserName and Password
        Then Message displayed Login Successfully

Scenario: Successful LogOut
        When User LogOut from the Application
        Then Message displayed LogOut Successfully
```

Note*: This is a simple test in Cucumber. Don’t worry about the syntax if you don’t understand it. Ideally, you should be able to understand the intent of the test just by reading a test in feature file. We will discuss this in more detail in the next chapter.*

Keywords

Now moving forward we have just defined a test. You will notice colored parts of the tests (*Feature, Scenario, Given, When, And and Then*). These are keywords defined by ***Gherkin***. Gherkin has more keywords and we will discuss those in the following tutorials. But to start off we can quickly explain some of the keywords in one line. Note this is not a complete listing of Keywords:

Feature: Defines what feature you will be testing in the tests below

Given: Tells the pre-condition of the test

And: Defines additional conditions of the test

Then: States the post condition. You can say that it is the expected result of the test.

Gherkin

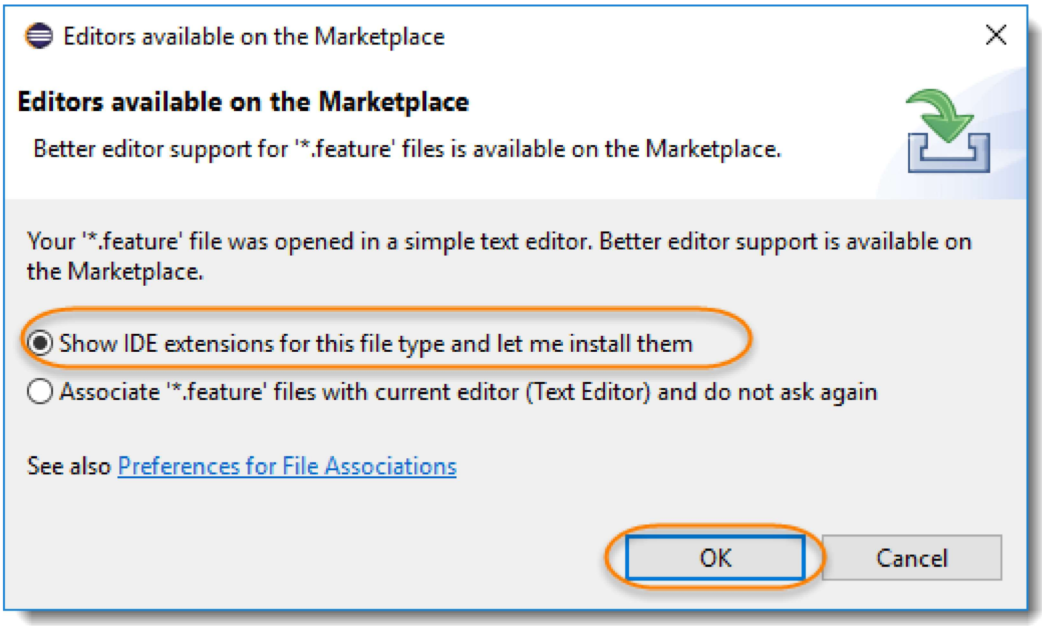
A language above is called ***Gherkin*** and it implements the principles of ***Business readable domain specific language***(*BRDSL*). Domain-specific language gives you the ability to describe your application behavior without getting into details of implementation. What does that mean? If we go back to our tutorial in ***TDD*** we saw that we wrote test code before writing any application code. In a way, we described what is the expected behavior of our application in terms of tests. On TDD those tests were pure Java tests, in your case, those might be a C++ or C# tests. But the basic idea is that those are core technical tests.

If we now come back to ***BDD/BRDSL*** we will see that we are able to describe tests in a more readable format. In the above test, it's quite clear and evident, just by reading, what test would do. At the same time of being a test it also documents the behavior of an application. This is the true power of *BDD/BRDSL* and it will become the power of cucumber eventually because cucumber works on the same principles.

Steps to install the Natural Eclipse Editor for Gherkin

You get this option automatically when try to create a new file with .feature ext. But if you do not get that one, you can anytime go to Eclipse Marketplace and look for the same to install it.

1)Just select the first option of ***Show IDE extensions*** if it is not pre-selected and click OK.



Editors available on the Marketplace

Editors available on the Marketplace

Better editor support for '*.feature' files is available on the Marketplace.

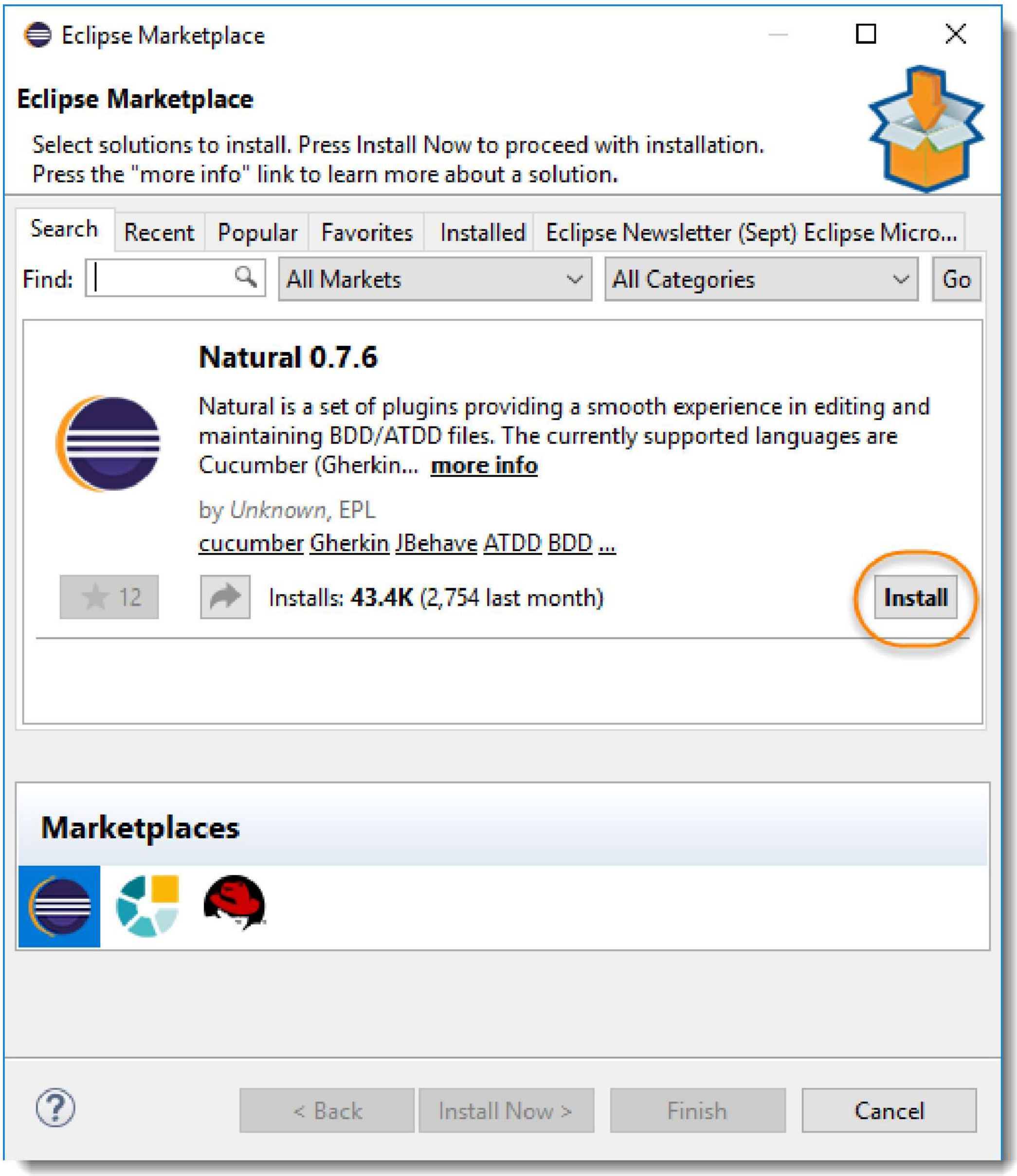
Your '*.feature' file was opened in a simple text editor. Better editor support is available on the Marketplace.

● Show IDE extensions for this file type and let me install them

○ Associate '*.feature' files with current editor (Text Editor) and do not ask again

See also Preferences for File Associations

. Natural is the name of the plugin, so this can also be found ***Eclipse Marketplace***. Just click Install.



Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Natural 0.7.6

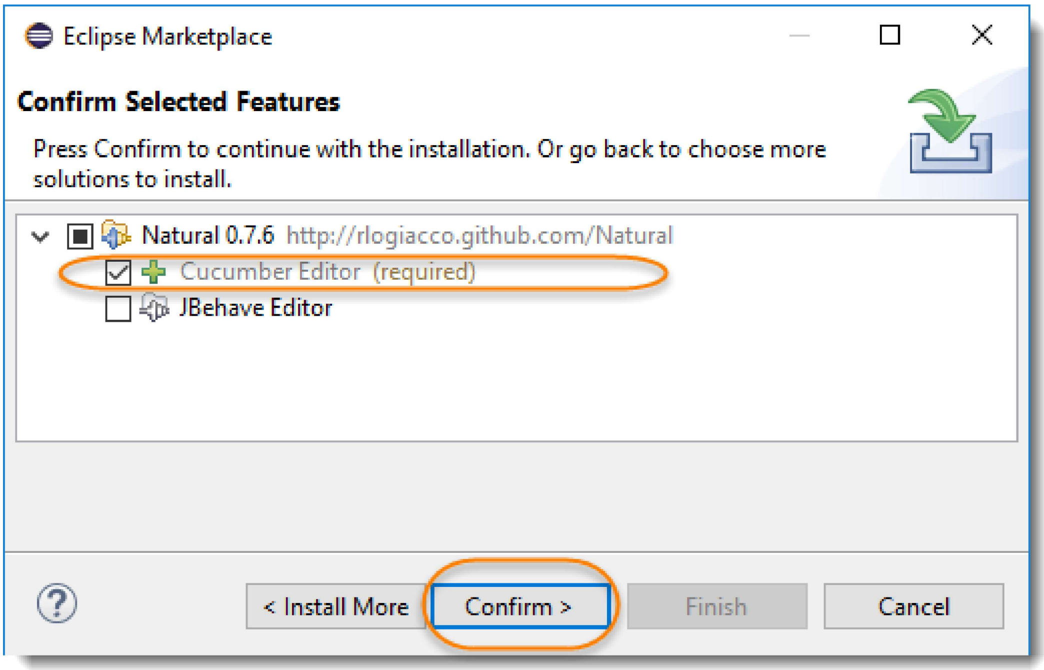
Natural is a set of plugins providing a smooth experience in editing and maintaining BDD/ATDD files. The currently supported languages are Cucumber (Gherkin... more info

by *Unknown*, EPL

cucumber Gherkin JBehave ATDD BDD ...

Installs: **43.4K** (2,754 last month)

. This will give you an option to select, whether you like to use it for Cucumber or JBehave(*Another BDD Framework*). Go for Cucumber.

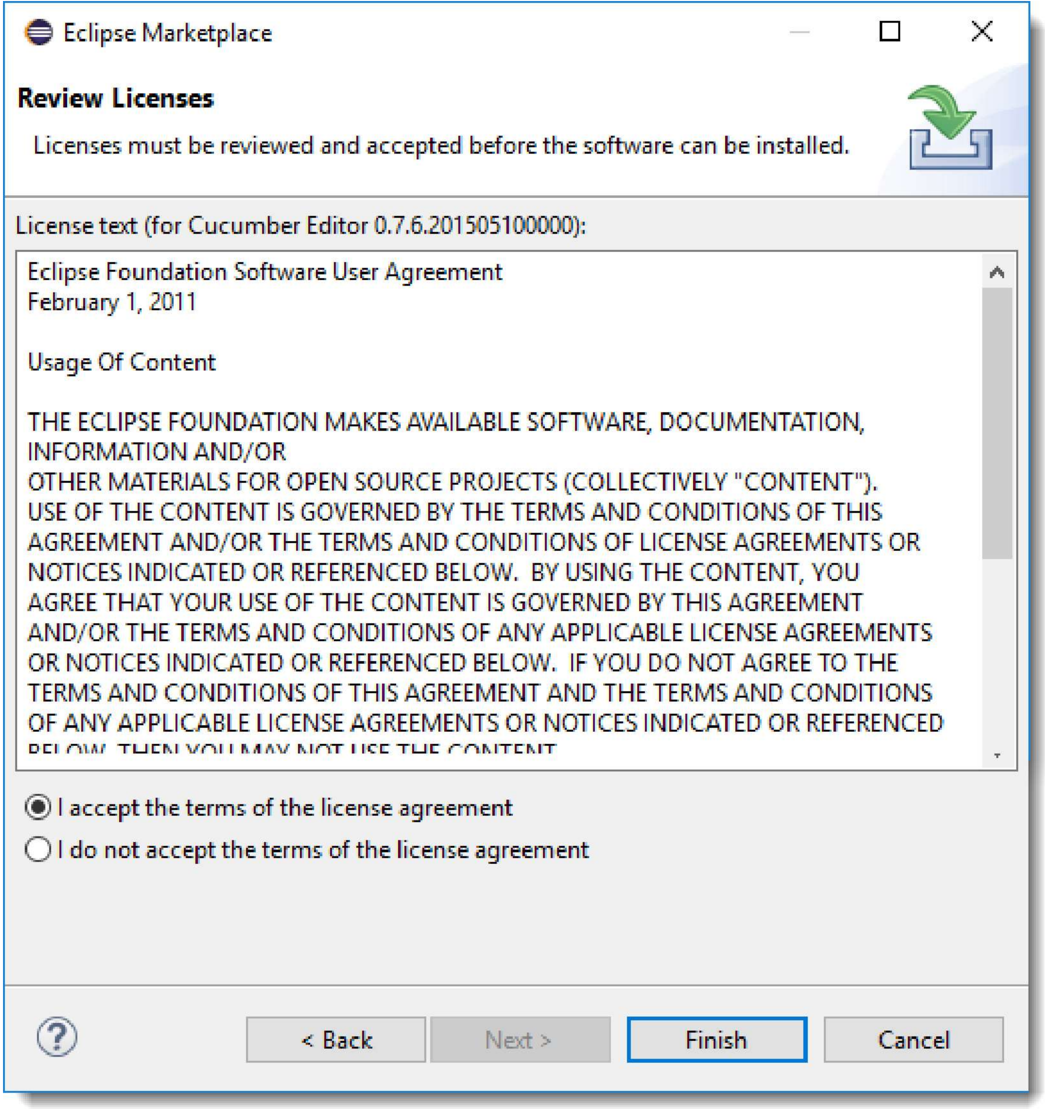


Confirm Selected Features

Press Confirm to continue with the installation. Or go back to choose more solutions to install.

- ▣ Natural 0.7.6 http://rlogiacco.github.com/Natural
  - ☑ Cucumber Editor (required)
  - □ JBehave Editor

. The last step is to accept the Terms and Conditions.



Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

License text (for Cucumber Editor 0.7.6.201505100000):

Eclipse Foundation Software User Agreement
February 1, 2011

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT

● I accept the terms of the license agreement

○ I do not accept the terms of the license agreement

***Note**: Once done, it may ask you to restart the Eclipse, if not then it is suggested to restart the eclipse after installing any plugins.*