RichTest Eclipse Plugin User Manual

1. RichTest Plugin Overview

RichTest plugin (Rich Software Testing) is based on the Eclipse integrated development environment and is written in Eclipse version 4.8, which is recommended for running RichTest.

By installing RichTest on Eclipse, the developer will be able to develop TDD projects faster and easier as fewer test cases are selected and executed in the development phase.

2. Download RichTest Eclipse Plugin

1. Go to https://github.com/MafiZo/RichTest.git and download RichTestPlugin Folder completely.
2. Copy the "Feature" Folder and the "plugins" folder to a specified folder on your computer.
3. The main plugin File is "RichTest_4.2.0.2.jar" that is placed in "plugins" folder available at: https://github.com/MafiZo/RichTest/blob/master/RichTestPlugin/plugins/RichTest_4.2.0.2.jar

3. Installing RichTest Eclipse Plugin

1. In Eclipse, go to Help -> Install New Software... (Fig. 1).

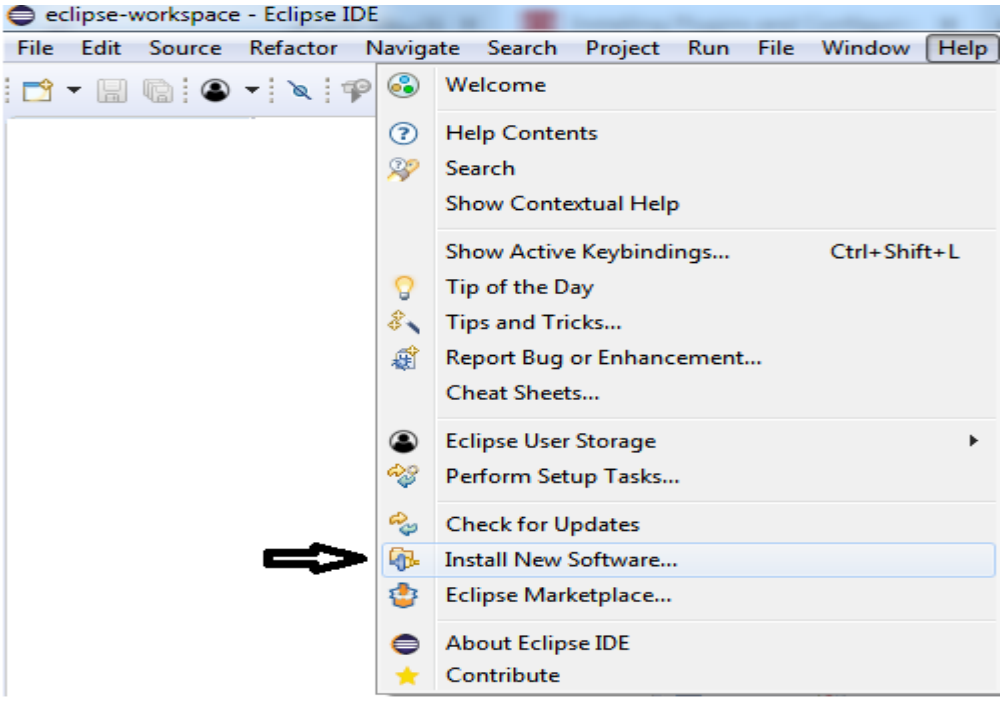

Fig. 1 Plugin Installation

2. In the Install Window, Press the **Add** button ...(Fig. 2)
3. In the **Add Repository** window, press the **Local** button or the **Archive** button to open a file browser.
4. In the file browser, select the Eclipse plugin file that you downloaded, as shown in Fig.2.
5. The **Add Repository** window appears. Press **OK**.
6. The category RichTest Tools appears in the Name area. **Check** the box in front of RichTest Tool and Click **Next**.
7. The installation details are displayed. Press **Next**. The items you checked are listed. Press **Next** again.
8. Accept the terms of the license agreement and click **Finish**.
9. Check the RichTest Tool Category and press the **Next** button to start the Add wizard.
10. Accept the terms of the license agreement and click **Finish**. The installation process starts.
11. When the installation process completes, restart Eclipse.

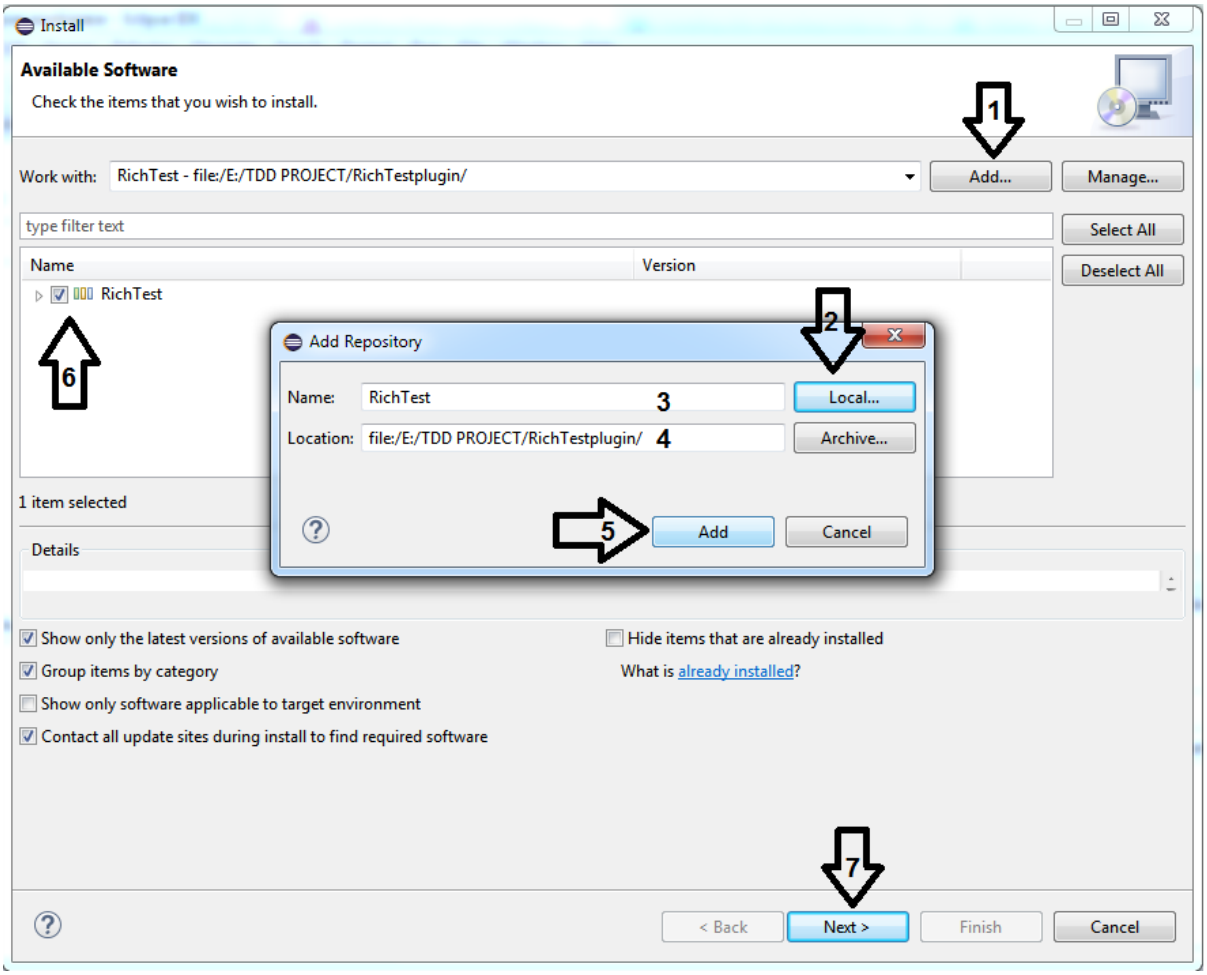

Fig. 2 Install Window

4. Setting Up RichTest Eclipse Plugin

1. In Eclipse, go to Window -> Preferences (Fig. 3).

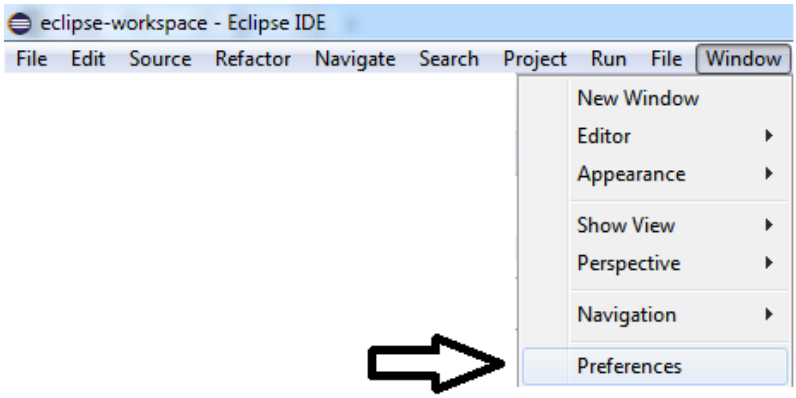

Fig. 3 Plugin Setting

2. In the **Preferences** window, select "RichTest". RichTest preferences will be appear (Fig. 4). **Preferences** such as Automatic/Manual Block Selection, Code Granularity (Coarse/Fine), and Enable/Disable TDD Mode.
3. Copy the address of the **Dependencies** and **Exceptions** folders.
4. Select "**Block Selection**" mode.
5. Select "**Code Granularity**" level.
6. Keep the "**TDD Mode**" option "**Enable**".
7. Click **Apply** button or click **Apply and Close** button.

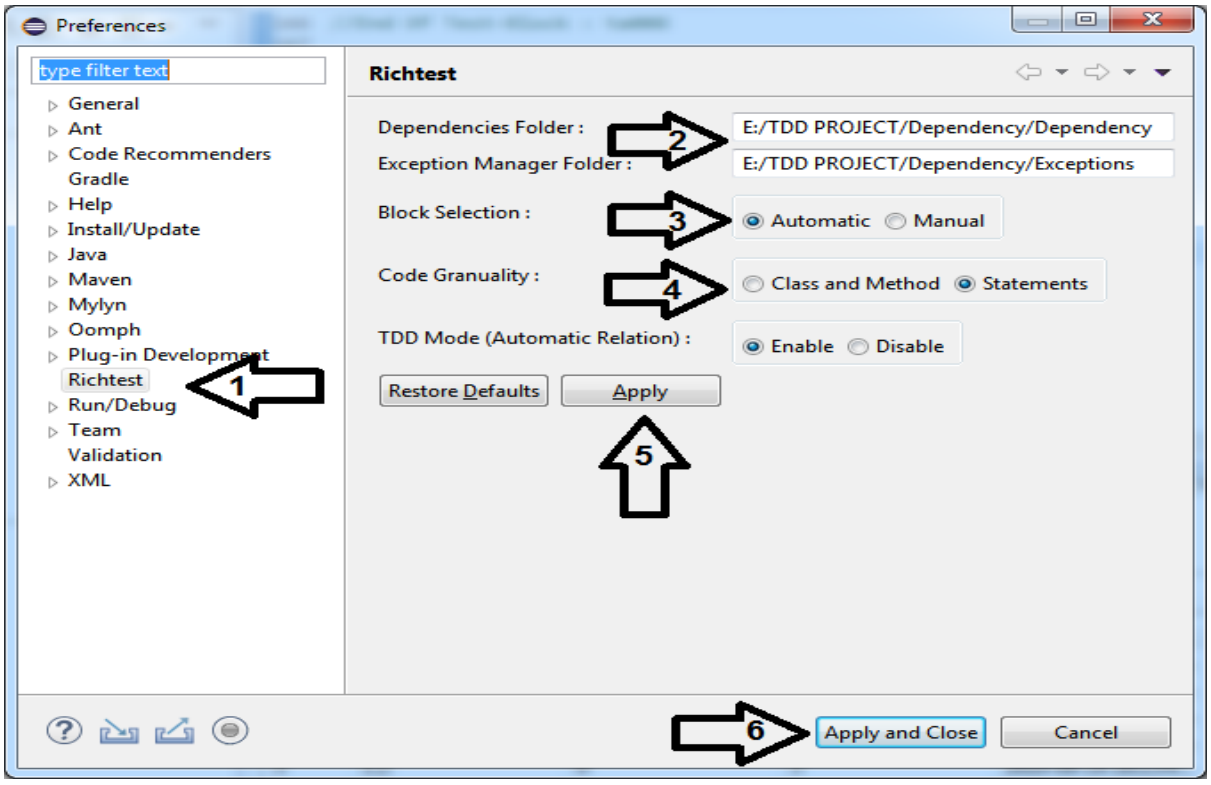

Fig. 4 RichTest Preferences Window

5. How to use RichTest?

Developer writes test case and runs it. If the test case fails, she/he writes production code to pass test. If the above setting shown in Fig. 4 are made (Block Selection=**Automatic**), RichTest automatically creates code blocks and test blocks and link them together.

If the Block Selection option is set to **Manual**, the developer can block any arbitrary part of the code as a block by selecting the desired range and pressing **CTRL+1** simultaneously.

At last the developer can use **Regression Test Wizard**, to select the related test cases automatically to run instead of running all test cases. It is available from:

File -> New -> Others -> Regression test -> Regression Test Wizard

as Shown in Fig. 5. Then select the **Next** button twice as shown in Fig. 6 and Fig 7.

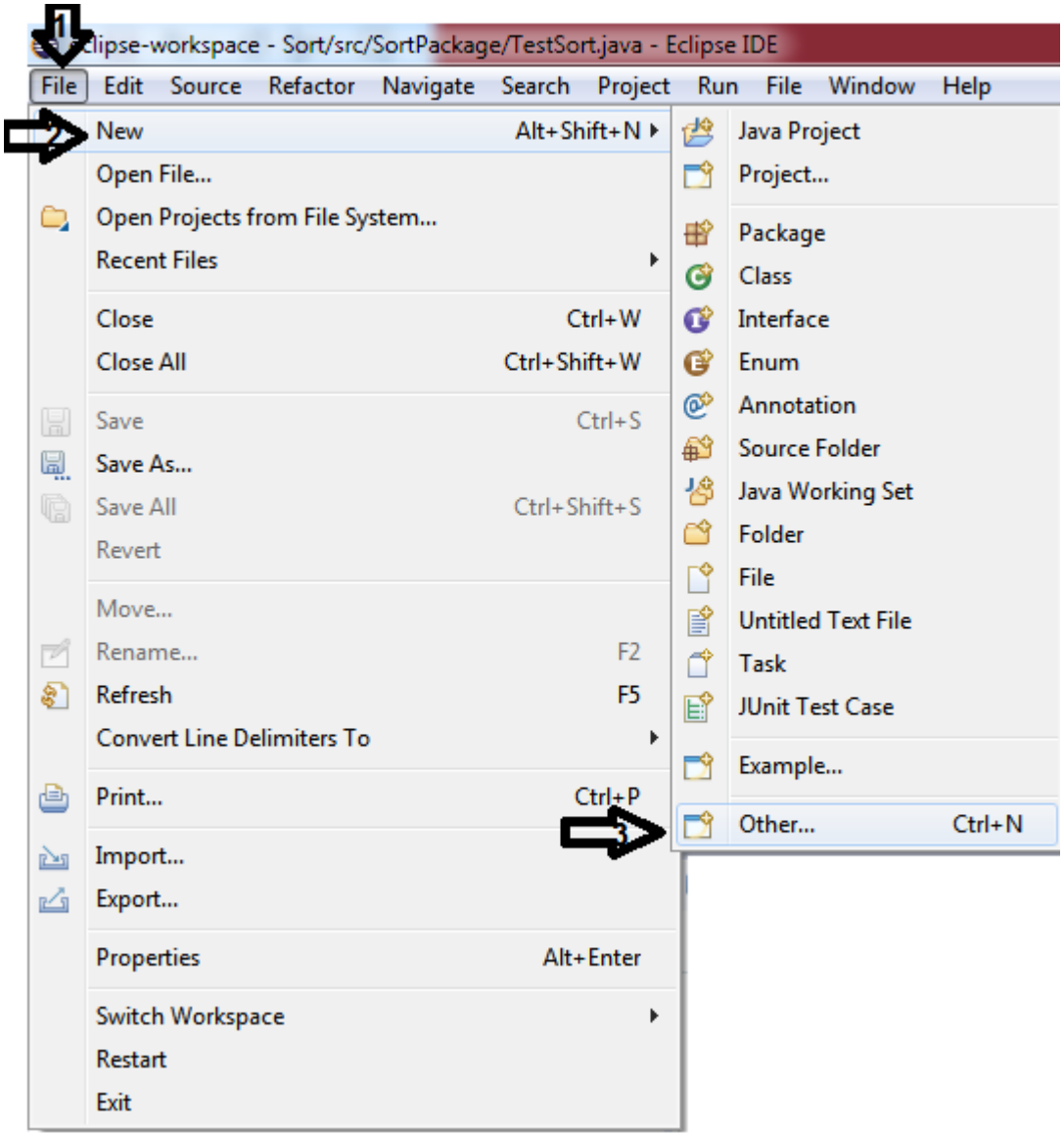

Fig. 5 Regression Test Wizard

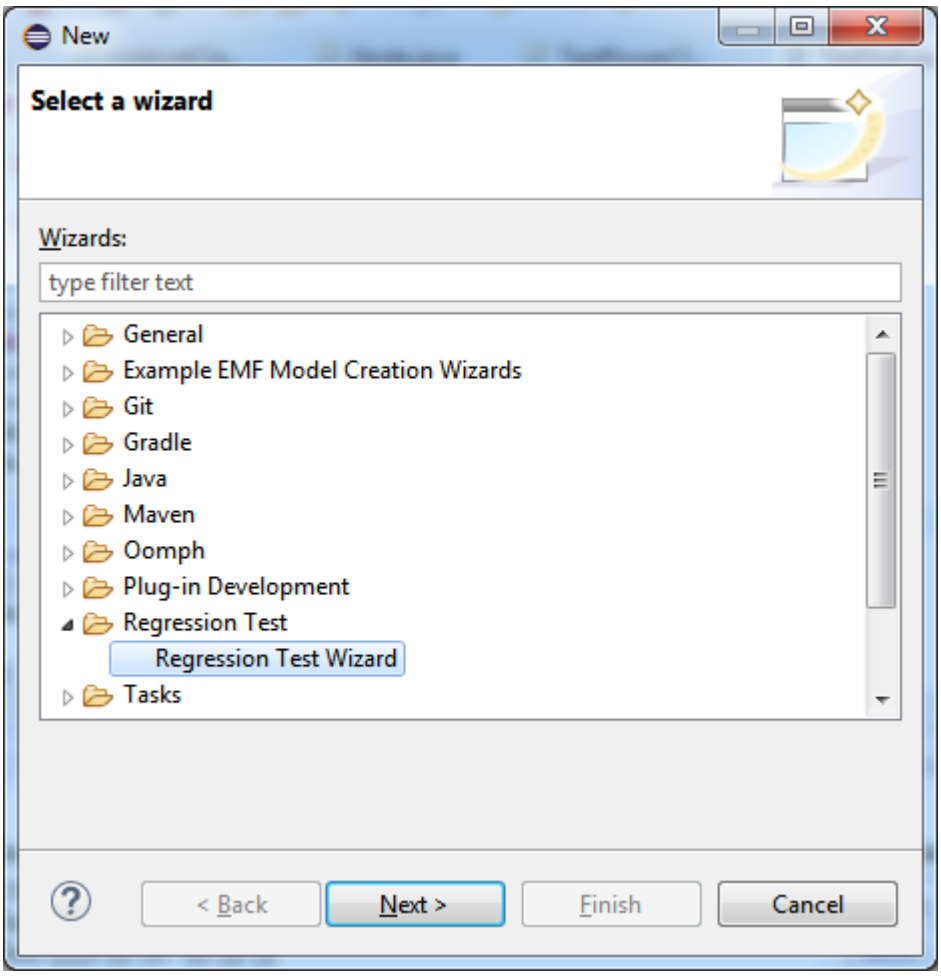

Fig. 6 Selecting Regression Test Wizard

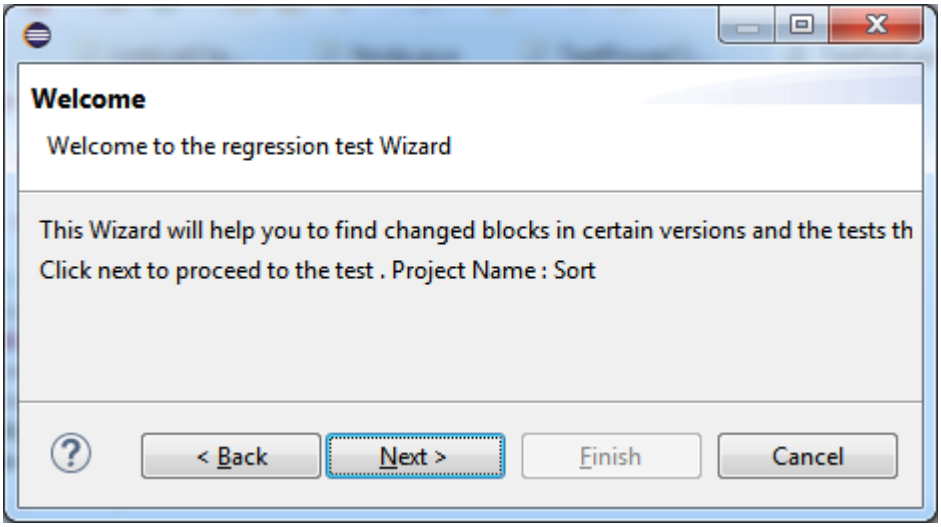

Fig. 7 Regression Test Wizard for the Selected Project

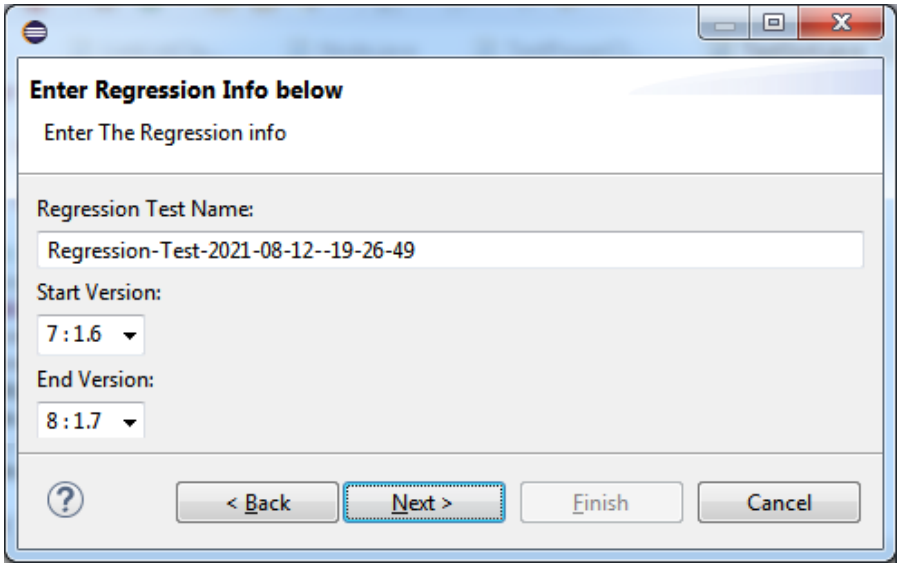

Fig. 8 Specify the Nameand Start and End Version of Regression Test

Fig. 8 shows that developer can select a name for Regression Test. As can be seen, by default the name of the test is "Regression- Test" followed by the date and time. Also she/he can select **Start** and **End** version. As can be seen, the last two versions of the project are considered as the **Start** and **End** versions by default. After pressing **Next** button, candidate test will be shown (Fig. 9).

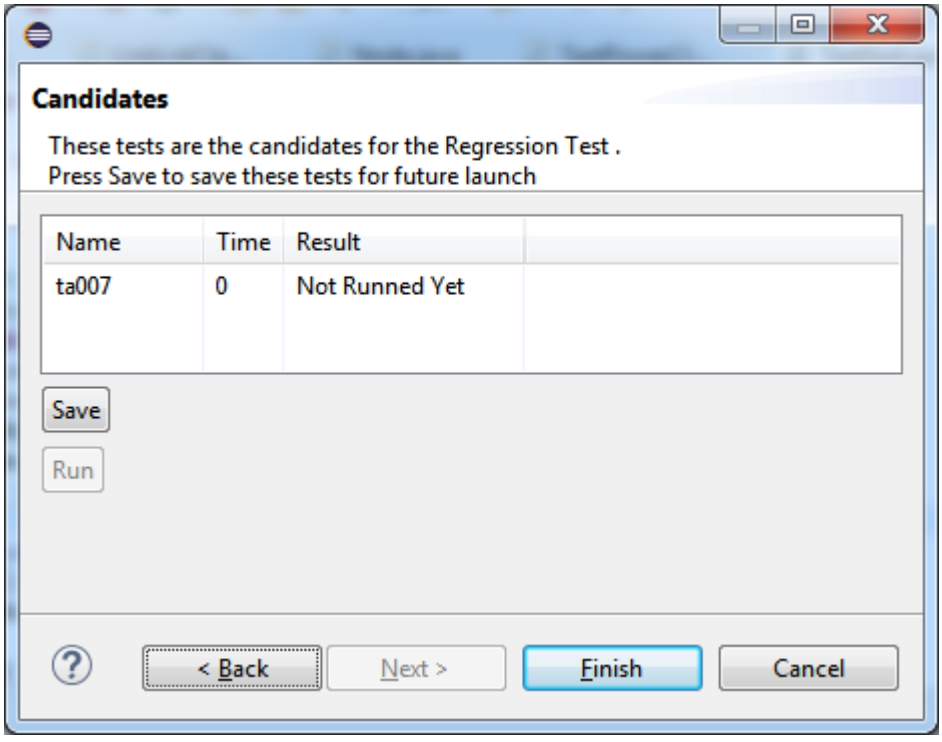

Fig. 9 Candidate Test Cases

Regression Test Wizard produces a list of candidate test cases between the “Start Version” and “End Version” of the program. After specifying the desired Start and End versions, recently added test cases are highlighted, and all test cases associated with the modified code blocks are also nominated.

The developer can **Save** then **Run** the selected test case(s) and then press the **Finish** button (Fig. 10). This Regression Test will be also available from Regression Test View. In Regression Test View each Regression Test could be **Export** to Excel format and also could be **Run**. Fig. 11 shows RichTest Regression Test Selection Process.

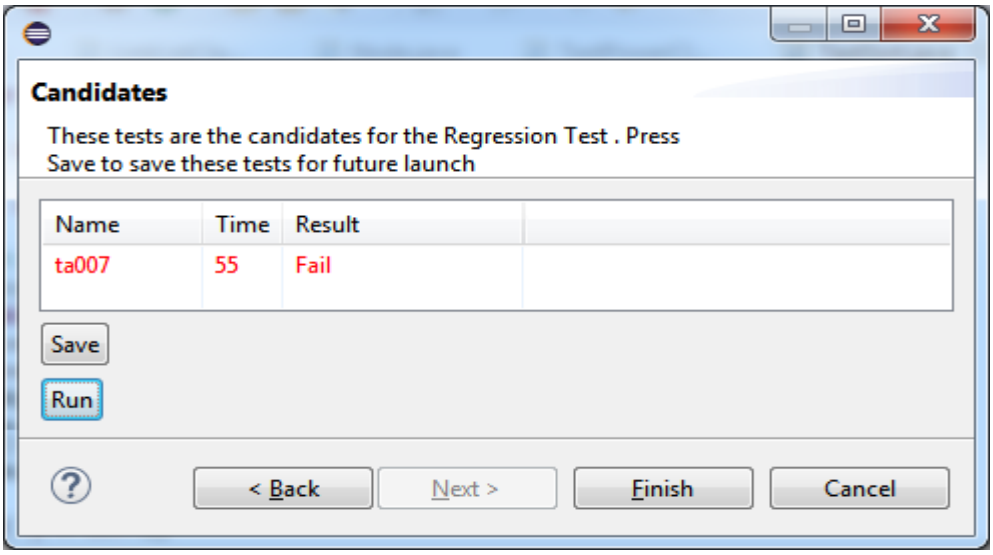

Fig. 10 Save and Run Candidate Test Cases

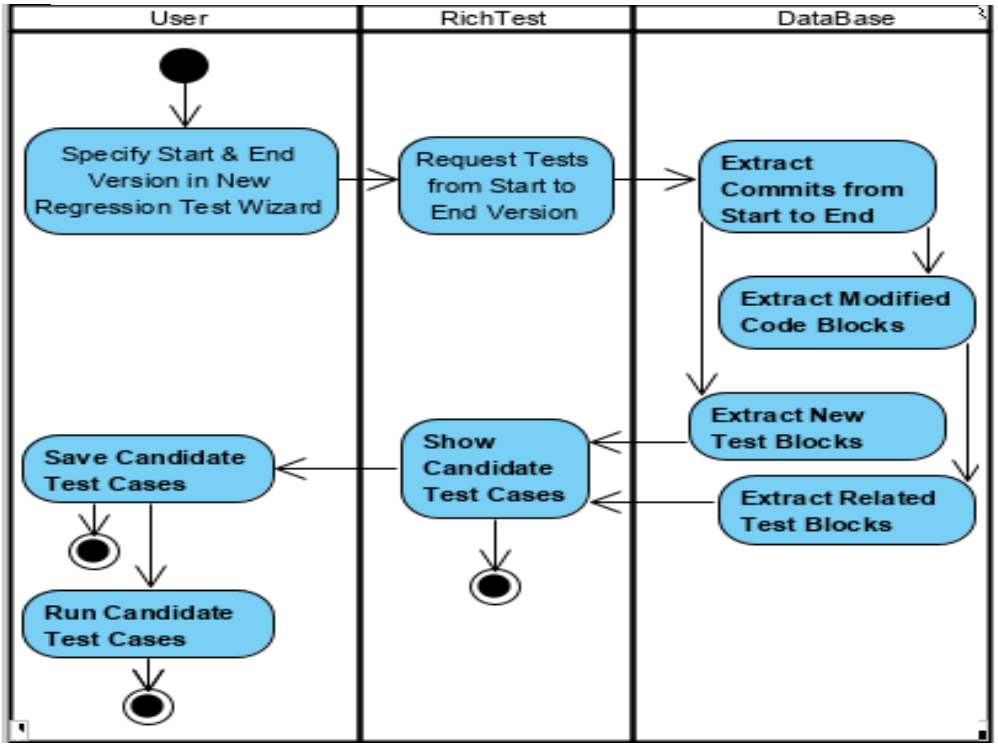

Fig. 11 RichTest Regression Test Selection Process

6. Facilities of RichTest Eclipse Plugin

RichTest offers several widgets, such as Block Information View, Commit View, Version Manager View, Regression Test View, and Compare View to facilitate the use of RichTest available from Window->Show View ->Other -> RichTest which is explained below:

BlockInfoView:

It is possible to display the Block List and the relationship between code blocks and test blocks, as well as manage the relationship manually.

Fig. 12 illustrates that test case “ta009” is the only test block that is connected to code block “ca005”. “Manage” key lets developer manage the relation manually as seen in Fig. 13.

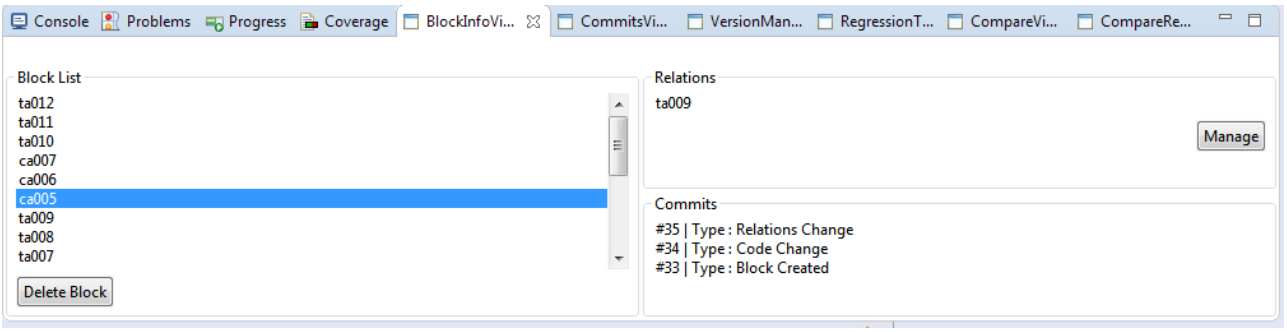

Fig. 12 Block Info View

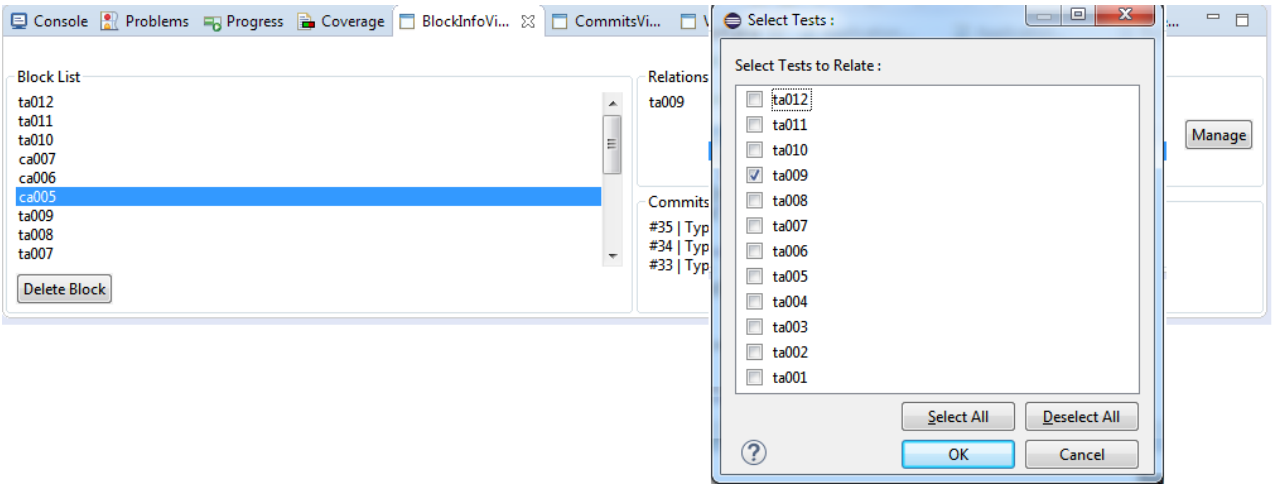

Fig. 13 Manuall Management Window of BlockInfo View

CommitView:

It is possible to show all blocks creation and modification and also filter all versions and commits of each block. Fig. 14 illustrates the Commit View. Developer can see the type of modification and block content.

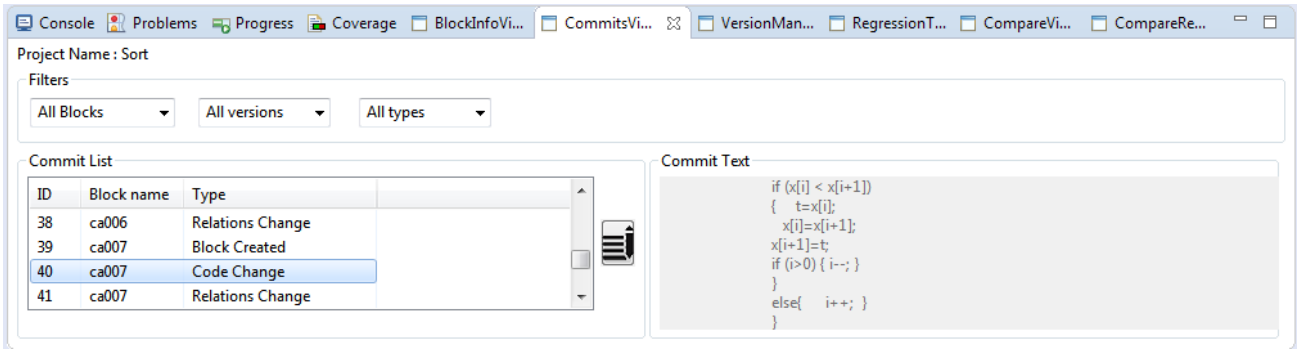

Fig. 14 Commit View

VerssionManagerView:

It is possible to set a new version for the projects. Fig. 15 illustrates Version Manager View that shows all versions of project as well as creates new version.

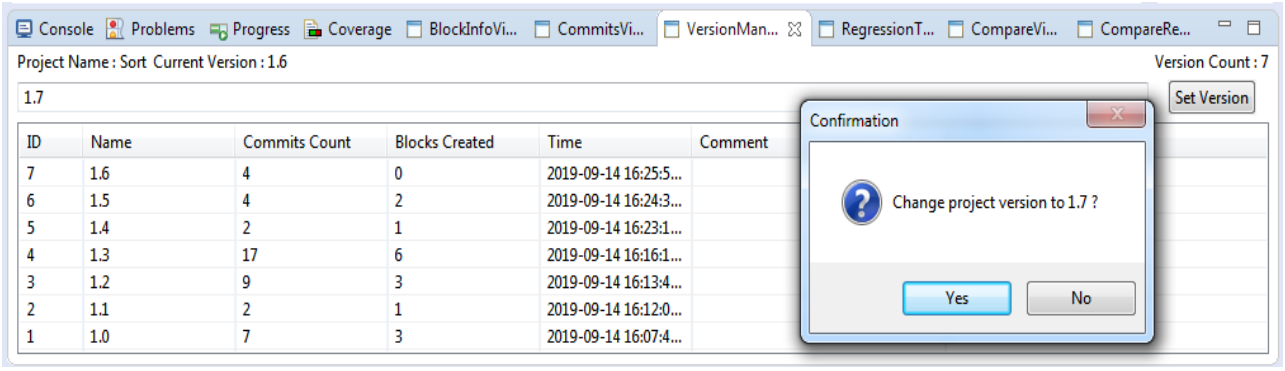

Fig. 15 Version Manager View

RegressinTestView:

It is possible to automatically select candidate test cases, and run them to show the time and results (Not Runned Yet is shown in blue/Fail in red/Pass in green), and export them to an Excel file format.

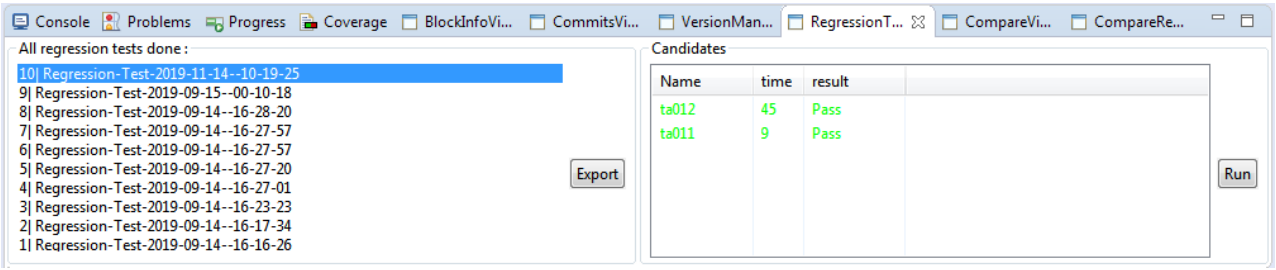

Fig. 16 Regression Test View

Fig. 16 illustrates Regression Test View. Two test case have been selected and run successfully (colored green), required time and test result have been shown.

CompareView:

It is possible to compare two different commits of each block. The code block will be shown in two situations (before/ after) and the differences will be colored and presented on ***CompareResultsView***.

Related figures are attached.

Fig.17 shows Compare View. Developer can select the desired code block to see the history of its changes and select two commit for comparison. Fig. 18 shows comparison result.

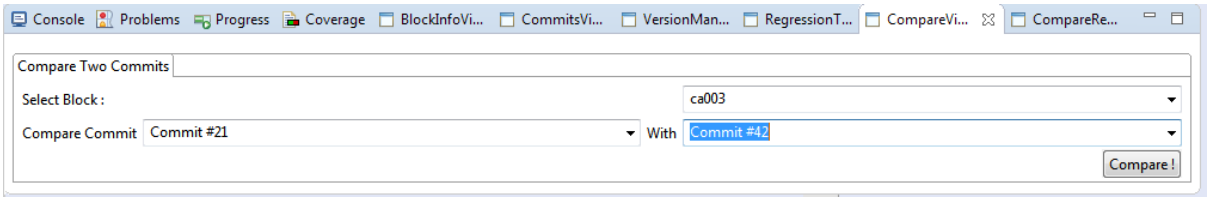

Fig. 17 Compare View

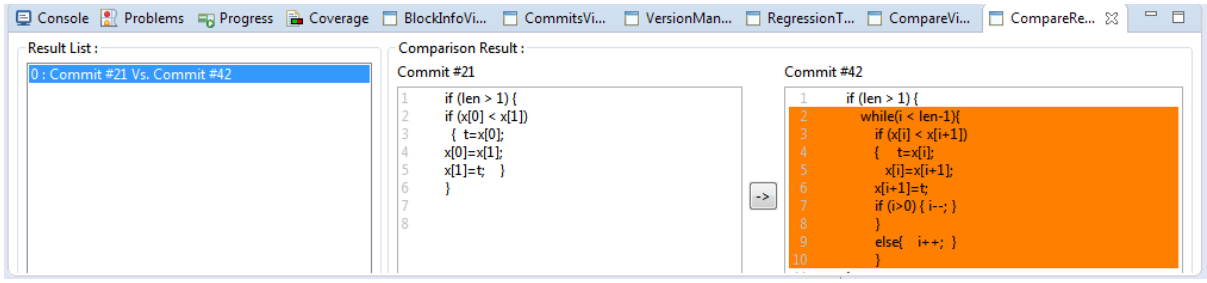

Fig. 18 Comare View Result