RE-Tool Manual

Pre-requisites:

The user must have the following software installed:

      - Firefox (web browser)

      - Selenium IDE (plugin for Firefox)

      - Eclipse (the project can be run in other IDE's but use with Eclipse for maximum compability)

      - JAVA

To import the project into Eclipse:

1 - Choose File->Import.
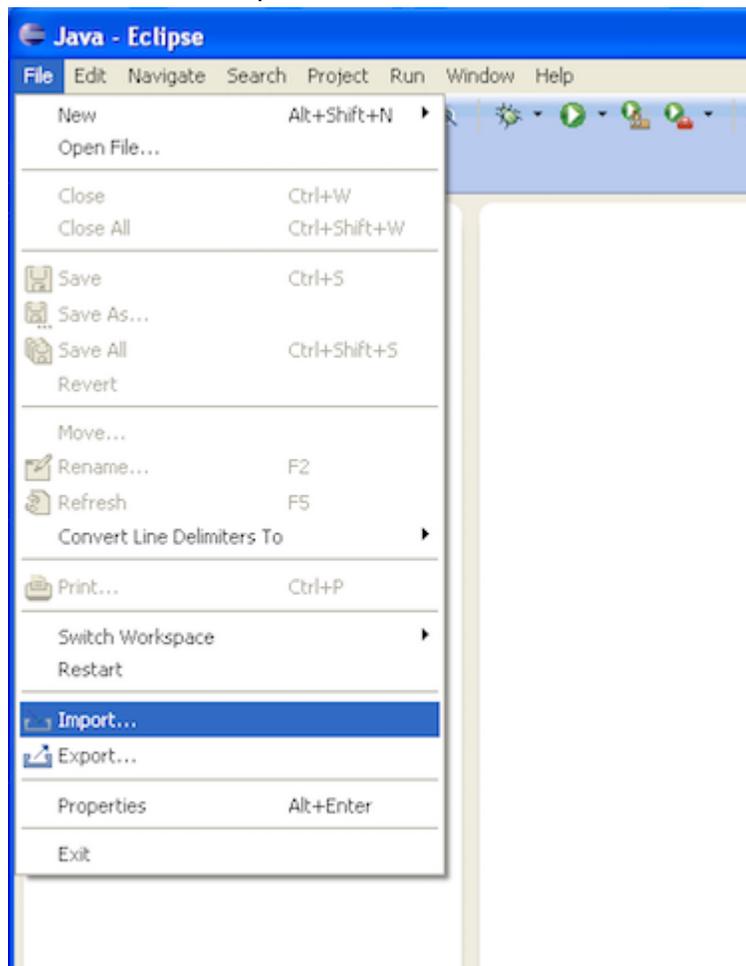
2 - Expand the Menu "General" and select "Existing Projects into Workspace"
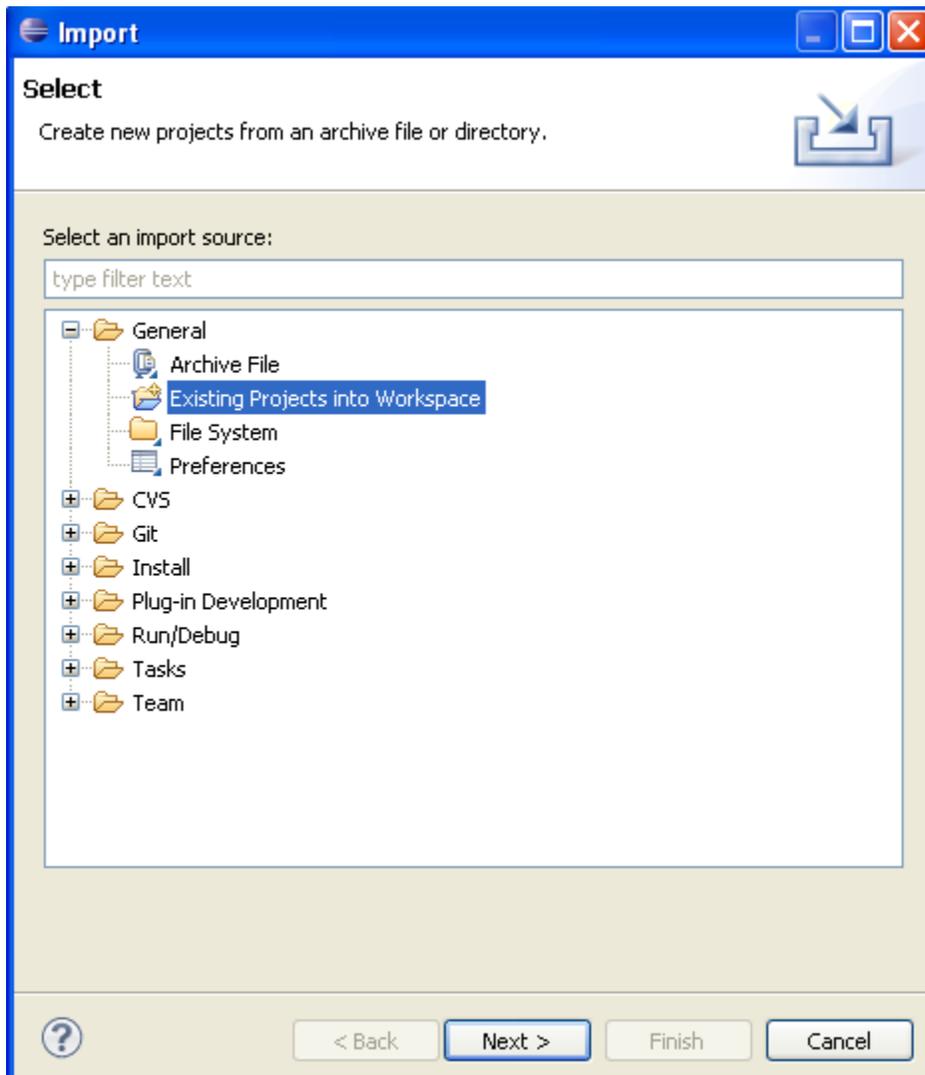
3 - Click on "Select archive file" - > "Browse" and choose the DiffHTMLCatcher.zip file
4 - Click on "Finish". The project is now imported.
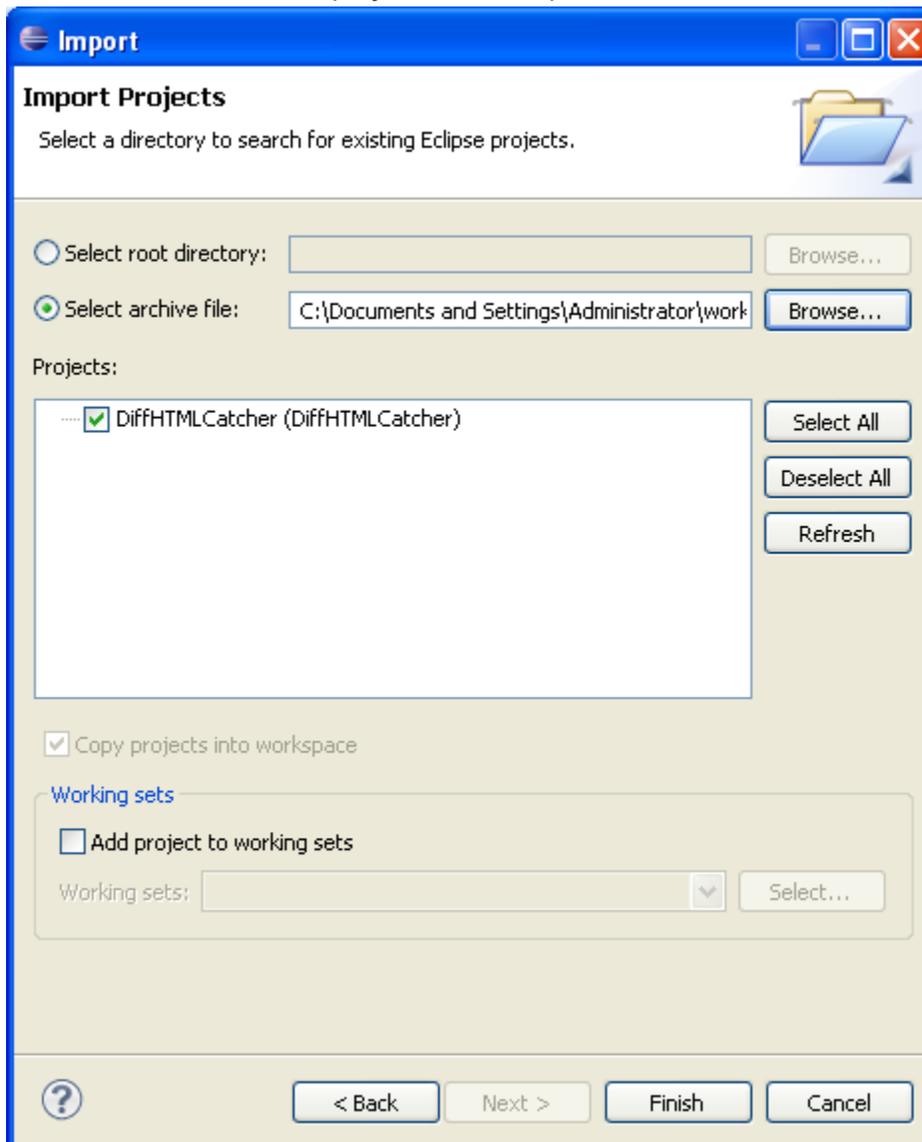
Run the program:

1 - To run the program, the user must run the Class "BrowserHandler". If the user is
running the .jar file, the url goes as the first argument (java -jar DiffHTMLCatcher http://
www.amazon.co.uk)
      a) If the user wants to change the URL of the webapplication to use, he must change the
variable *pageURL* at the beggining of the class.

2 - The program will then start Firefox, create the necessary folders and files. The user must
manually start Selenium IDE as soon as Firefox stops loading. To do this, the user must click on
"Tools ->Selenium IDE".
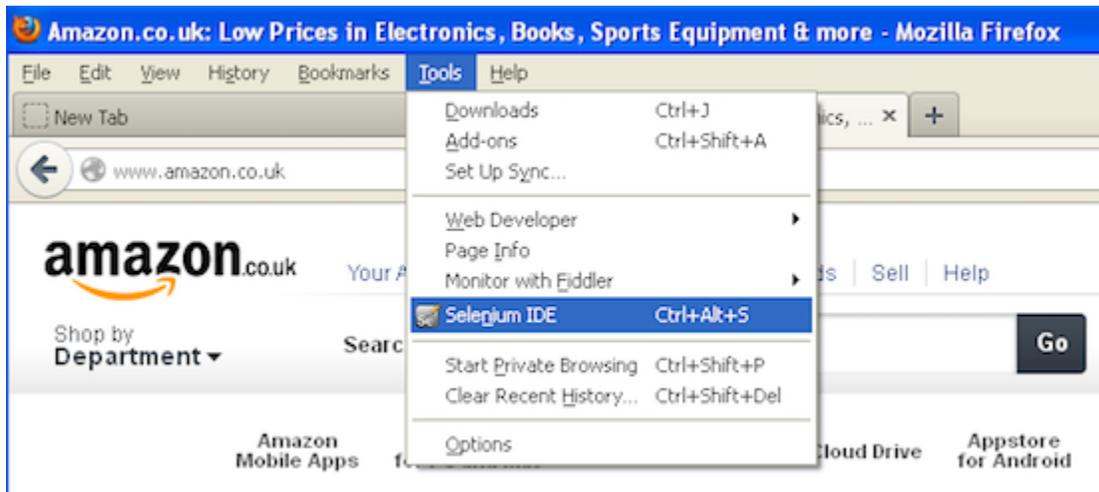
3 - From now on, the user can start interacting with the website.

Each time an action caused by the user is detected by Selenium IDE but DOES NOT change anything in the page (for example a simple click in the Amazon search bar causes a click action to appear but does not change the page), the user must press the VK_CLOSE_BRACKET key.

Each time an action caused by the user is detected by Selenium IDE and changes elements in the page, the user must press the VK_DEAD_ACUTE key.

Each time an action caused by the user is detected by Selenium IDE as TWO or more different actions, two possible actions can be performed:
        - If nothing in the page was changed, the user must press the VK_CLOSE_BRACKET key the number of times that Selenium IDE detected as actions
        - if something as changed in the page, the user must press the VK_CLOSE_BRACKET key the number of times that selenium IDE detected as actions MINUS ONE and then press the VK_DEAD_ACUTE key.
For example, if one action performed by the user reported as three actions in Selenium IDE:
        - if nothing was changed in the page, the user presses VK_CLOSE_BRACKET three times
        - if something changed, the user presses VK_DEAD_ACUTE two times and VK_DEAD_ACUTE one time

At the end of the execution, the user must save the Selenium execution trace as "selenium.html" in the HTMLFinal folder that the program created. After this, the user must press the VK_ESCAPE key (esc button) to terminate the program and start the inferrer process, that does not take more than a couple of seconds.

To know which keys are VK_DEAD_ACUTE and VK_CLOSE_BRACKET, the user can execute the KeyboardEvents class to see how the each keyboard key is registered, and he can change the code to accept different keys.

In the function NativeKeyPress in the BrowserHandler class:

```
if (e.getKeyCode() == NativeKeyEvent.VK_ESCAPE) {

else if (e.getKeyCode() == NativeKeyEvent.VK_DEAD_ACUTE){

else if (e.getKeyCode() == NativeKeyEvent.VK_CLOSE_BRACKET){
```

For more information, the user can consult the documentation of the JNativeHook Library.