## **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"

🖨 Import	
Select Create new projects from an archive file or directory.	Ľ
Select an import source:	
<pre>type filter text General General</pre>	
(?) < Back Next > Finish (	Cancel

3 - Click on "Select archive file" - > "Browse" and choose the DiffHTMLCatcher.zip file 4 - Click on "Finish". The project is now imported.

🖨 Import		
Import Projects Select a directory to searc	h for existing Eclipse projects.	
<ul> <li>Select root directory:</li> <li>Select archive file:</li> <li>Projects:</li> </ul>	C:\Documents and Settings\Administrator\work	Browse Browse
	(DiffHTMLCatcher)	Select All Deselect All Refresh
Copy projects into working sets	rkspace ng sets	Select
?	<pre></pre>	Cancel

## 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".

la Amazon.co.uk: Low Prices in Electronics, Books, Sports Equipment & more - Mozilla Firefox			
Eile Edit View History Bookmarks	Tools Help		
New Tab	Downloads     Ctrl+J       Add-ons     Ctrl+Shift+A       Set Up Sync		
amazon.co.uk Your	Web Developer  Page Info Monitor with Eiddler  Is   Sell   Help		
Shop by Department - Searc	C Start Private Browsing Ctrl+Shift+P Go		
Amazon Mobile Apps	Clear Recent History Ctri+Shirt+Del		

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  $\mathsf{VK\_CLOSE\_BRACKET}$  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.