This file describes basics of usage test automation suite package for WeTrust POC Demo app.   
  
The test framework comes as a zip folder, which contains following key components:   
 - JAR file (archived executable version of code package)  
 - xls folder with 3 minimal required DDT files used by test script: DDT\_Global.xls, GUI\_Map.xls and TC\_list.xls   
 - the ‘upload’ directory which contains webdriver driver for FF-3  
- log4j.properties file, used by java   
  
To run script pls make sure you have one of recent version of FF browser installed on your system. It’s a critical condition to run the test script. When you done with all this, open a console window and cd into your package dir. From the console verify that your directory contains all files described above. To run execute command: java -jar <Jar\_File\_Name> + Enter  
  
If you did everything correct you will see test log started in the console. Recomend - maximize the console window for better vision of log details. Ideally, if you run in from separate monitor then browser. In 15-30 sec you should see a browser windo get launched, starting with default Google page and then navigating to WeTrust test demo app. By default it should run all tests I got able to create during this time.   
  
You need to know reg role and use of those 3 key xls files. They containing DDT data, which is deeply integrated in every aspect of given framework. For now, you can think only about these two: TC\_list.xls and DDT\_Global.xls. In TCs you can define your own desired test scope from the list of available high-level test cases. To enable/disable any TC jsut set ‘1’ to enable and ‘0’ to disable every test case in the next column on the same row with TC name. As easy as that. You may want to narrow down a test scope to those TCs where you see failures. And there ARE failures, probably made there in purpose of view testing efficiency. If you look DDT\_Global tbl you will notice the same concept - having global value name in the left column and associated value in the column right, where you see ‘1’ in the 2nd (brown) row of col #3. THis makes that column being used to retrieve values from and assign in global hashMap obj with key values made after corresponding names in 1st column. Like you can make this script running against another URL, or decrease to (timOut) limit for not founf expected UI objs and for synching, and so on. You can run selenium on remote server too by simply enabling that feature by entering ‘1’ in the col #6 row 6, and running selenium standalone server on remote server, where you should also install FF browser. And so on.   
  
In the simple way - just unpack the file in your local dir, cd to it from console and execute that java -jar command as described above. The full set of tests currently available script runs about 20-25 mins. After that you will see a nicely formatted list of test results with test names and statuses. Most of those tests have been dynamically generated by ‘matrix’ of different users and permissions/relationships between then. If we change those uesrs (add more, create more groups, etc) the users parser class should be updated accordingly and scripts will recalculate all combinations to test. Each scenario tested is reported as individual dynamically generated test. But all tests are combined and running by high-level categories TCs described earlier, by TC\_List table.   
  
Pls be aware, that by observing test logs in the console sometime you will notice a blocks of similar lines by packets of 10 lines showing waiting for something. This is made in purpose. By test flow scripts expects status text ‘permission denied’, but instead, it sees unexpected ‘success’ msg, but it waits for expected UI obj to appear. 10 sec - is a timeOut limit set by DDT\_Global table (‘to’ value). feel free to change it and try. But that’s the concept of this test: if expected UI element didn’t appear, synching mechanism triggering an exception and test gets failed. You may enjoy to see how script reacts and handles such failure scenarios. They all are reflected in test status at the end.   
  
For any Qs or comments feel free to contact me at **any time** by:   
[linuxaa@yahoo.com](mailto:linuxaa@yahoo.com)+1-408-854-1958  
  
Enjoy running this framework, changing the scope and params, loop it for like overnight mul-cycle tests, etc., and observing the resultsresults. I am sure, more hidden bugs can be found there if write more tests.