This is a short guide explaining how to install and use AutoBlackTest for testing interactive applications (at the current stage only the Java Swing-based GUI components are supported). In the following you can find the system requirements and instructions about how to install and run AutoBlackTest .

1. SW REQUIREMENTS:
The current version of AutoBlackTest runs on Microsoft Windows and requires IBM Rational Functional Tester. We do not tested the tool with all the version of Microsoft Windows and IBM Rational Functional Tester, so we recommend you to use the following versions:
- Microsoft Windows XP 32bit
- IBM Rational Functional Tester 8.1
- Java™ SE 6 Update 33

2. INSTALLATION INSTRUCTIONS:
We released AutoBlackTest as an Eclipse-based Java project. To import the project on your machine open IBM Rational Functional Tester and from the Java perspective use the import command to load the folder "...\workspaceDist\AutoBlackTest".
Since we also provide the workspace of AutoBlackTest you can directly use it. If you decide to use the workspace of AutoBlackTest, when you open IBM Rational Functional Tester chose the folder "...\workspaceDist " as current workspace. The workspace includes a case study.

3. SETUP INSTRUCTIONS:
To test a Swing-based interactive application and generate the regression test suite you have to setup the files "AutoBlackTest\abt-config\abt.properties" and "AutoBlackTest\abt-config\aut.properties". The first file, abt.properties, specifies the configuration options of AutoBlackTest. The second file, aut.properties, contains the configuration options of the application under test.
After you customize the above files (see subsections 4.1 and 4.2), you can execute AutoBlackTest from the Functional Test perspective of IBM Rational Functional Tester by running the Functional Test Script "abtRun\Tester.java".

The outputs produced by AutoBlackTest will be stored in a folder located in " \AutoBlackTest\outputs\experiment*_yymmdd_hhmm*" where " *yymmdd*" and " *hhmm*" are respectively the date and time when AutoBlackTest started its execution. The folder "experiment*_yymmdd_hhmm*" is organized into the following sub-folders and files:

- *coverage* folder: it contains data about code-coverage for each episode (a test case generated online) and for the entire execution. Into the log file, we already provide data about the code coverage rate, if you want to have more details you can open files stored into this folder by using the cobertura tool (http:\\cobertura.sourceforge.net).
- *stdOut* folder: it contains the standard output of the application under test.
- *stdErr* folder: it contains the standard error of the application under test.
- *errorStates.txt* file: this file contains the potential bugs of the application under test detected by AutoBlackTest.
- *exploration.log* file: this file is the log file of AutoBlackTest.
- *TestCases* folder: it contains all the test cases generated online by AutoBlackTest. It also includes a sub-folder *testSuite* containing the regression test suite.
- *State* folder: it contains a serialized object used for the generation of regression test oracles.

To execute the generated regression test suite you have to edit the file "AutoBlackTest\abt-

config\testsuite.properties". This properties file specifies the configuration options of the application to be tested for no-regression problems and the configuration options of the regression test suite to be executed.
After you customize this file (see subsection 4.3), you can run AutoBlackTest from the Fuctional Test perspective of IBM Functional Tester by running the Functiona Test Script "abtRun\TestSuiteExecutor".

4 PROPERTIES FILES

In the following we describe the content of each properties files by explaining the role of each parameters. For each parameter we show you a configuration example, we recommend you to strictly refer to them we you set up AutoBlackTest to test your application.

4.1 “abt.properties” configuration

The file is composed of the following parameters:

- *hours:* it represents the amount of hours AutoBlackTest is executed to test an application. Configuration examples:
  - hours = 12
  - hours = 0
- *minutes:* it represents the amount of minutes AutoBlackTest is executed to test an application. Configuration examples:
  - minutes = 0
  - minutes = 5
- *sleep_time:* it represents the execution delay between two consecutive actions used by AutoBlackTest. When an action is executed, the application is usually affected by different changes (i.e. a new windows is loaded after AutoBlackTest clicked on a button). Before AutoBlackTest executes the next action all the changes must be completed. The delay time guarantees that all the changes on the current GUI are terminated. The sleep time between actions is specified in seconds or the fraction of seconds. Configuration examples:
  - sleep_time = 1 (one second is the most common used value)
  - sleep_time = 2
  - sleep_time = 0.5

4.2 “aut.properties” configuration

The file is composed of the following parameters:

- *aut_classpath*: it represents the classpath of the application under test. All listed path must be separated by ";" and without space and carriage return. The special character "\" must be replaced by "\\". Configuration examples:
  - aut_classpath = C: \\buddi\\lib\\ qua-6.5.jar; C:\\buddi\\lib\\Browser2-1_3.jar;
- *aut_bin_directory*: it represents the path containing the compiled files of the application under test. The special character "\" must be replaced by "\\". Configuration example:
  - aut_bin_directory = C:\\buddi\\bin
- *aut_main_class*: it represents the main class of the application under test plus the arguments of the class. The class name and arguments must be separated by one space. The special character "\" must be replaced by "\\". Configuration examples:
  - aut_main_class = org.homeunix.thecave.buddi.Buddi

    - aut_main_class = org.homeunix.thecave.buddi.Buddi 100
- *aut_configuation_file_path*: it represents the configuration files of the application under test. AutoBlackTest will save a copy of these files in order to reinitialize the application before every test. All listed path must be separated by ";" and without space and carriage return. The special character "\" must be replaced by "\\". Configuration example:
    - aut_configuation_file_path = C:\\Documents and Settings\\lta\\Application Data\\Buddi\\Buddi3_Prefs.xml
- *aut_working_file_path*: it represents the files used and modified by the application under test. AutoBlackTest will create a folder "abtMyDocuments" in the user home and the listed files will be copied in that folder. File choosers of the applications under test will point at the folder "abtMyDocuments". All listed path must be separated by ";" and without space and carriage return. The special character "\" must be replaced by "\\". Configuration example:
    - aut_working_file_path = C:\\Documents and Settings\\lta\\wile.buddi3

4.3 "testsuite.properties" configuration

The file is composed of the following parameters:

- *testSuite_folder:* it represents the path to the test suite or to the test case that must be rexecuted. AutoBlackTest stores the generated test cases in the following path: "...\workspaceDist\AutoBlackTest\outputs\experiment_*yymmdd_hhmm*\TestCases". The special character "\" must be replaced by "\\". Configuration examples:
    - testSuite_folder=C:\\workspace\\AutoBlackTest\\outputs\\experiment_20120727_1103\\TestCases\\TestCase_1.tc
    - testSuite_folder=C:\\workspace\\AutoBlackTest\\outputs\\experiment_20120727_1103\\TestCases\\testSuite
- *propertiesSet_file*: it represents the path to the file containing information to generate the oracles of the selected test suite or test case. AutoBlackTest stores that file in the following path:
  "...\workspaceDist\AutoBlackTest\outputs\experiment_*yymmdd_hhmm*\state".
  The special character "\" must be replaced by "\\". Configuration example:
    - propertiesSet_file = C:\\workspaceDist\\AutoBlackTest\\outputs\\experiment_20120727_1103\\state\\propertiesSet.ser
- *aut_classpath*: it represents the classpath of the application under test. All listed path must be separated by ";" and without space and carriage return. The special character "\" must be replaced by "\\". Configuration example:
    - aut_classpath = C: \\buddi\\lib\\ qua-6.5.jar; C:\\buddi\\lib\\Browser2-1_3.jar;
- *aut_bin_directory*: it represents the path containing the compiled files of the application under test. The special character "\" must be replaced by "\\". Configuration example:
    - aut_bin_directory = C:\\buddi\\bin
- *aut_main_class*: it represents the main class of the application under test plus the arguments of the class. The class name and arguments must be separated by one space. The special character "\" must be replaced by "\\". Configuration example:
    - aut_main_class = org.homeunix.thecave.buddi.Buddi
- *aut_configuation_file_path*: it represents the configuration files of the application under test. AutoBlackTest will save a copy of these files in order to reinitialize the application before every test. All listed path must be separated by ";" and without space and carriage return. The special character "\" must be replaced by "\\". Configuration

example:
    - aut_configuation_file_path = C:\\Documents and Settings\\lta\\Application Data\\Buddi\\Buddi3_Prefs.xml
- *aut_working_file_path*: it represents the files used and modified by the application under test. AutoBlackTest will create a folder “abtMyDocuments” in the user home and the listed files will be copied in that folder. File choosers of the applications under test will point at the folder “abtMyDocuments”. All listed path must be separated by ";" and without space and carriage return. The special character "\" must be replaced by "\\". Configuration example:
    - aut_working_file_path = C:\\Documents and Settings\\lta\\wile.buddi3

5 TECHNICAL NOTES

In order to work properly, AutoBlackTest needs to reset the application under test to an initial state before each test begins. To do that, AutoBlackTest reinitializes the configuration files specified in the properties file and mocks the static method *currentTimeMillis* of the class *System* in order to reset the date/time on the application under test. However, this might not be enough for you in some cases. More appropriate reset actions for your needs can be coded in the batch file “AutoBlackTest\AppScript\userAdditionalReset.bat”. This script is executed before launching the application under test.

CONTACTS

If you have any problem when you install or use the tool, please ask for help to developers:

- santoro@disco.unimib.it
- riganelli@disco.unimib.it
- mariani@disco.unimib.it