# JUnit Workbook(2012)

*Links:*

- http://qtpselenium.com/selenium-tutorial/junit-selenium-training/

# JUNIT

## Annotations

@Rule
- Used to declare variables that JUnit can use internally for its trackind reporting of test progress/errors

@BeforeClass (Static)
- Ran before the first @Test in the class
- Ran before the @Before annotation

@AfterClass (Static)
- The last annotation ran by JUnit in the class

@Before
- Ran before each @Test

@After
- Ran after each @Test

@Ignore
- used with @Test to skip the test

@Test
- Declare a test that JUnit will run
- method name must have 'test'

@Parameters

## Handling Assertions

- Create class variable
- Create a Try/Catch block around evaluations
- Use 'Throwable' data type
- Use .addError() to accumulate 'Failures'

**Example**

```
@Rule //JUNIT Rule
public ErrorCollector errCollector = new ErrorCollector();
..
..
          @Test
            ..
```

```
          ..
          try {
                Assert.assertEquals(expected_total_friends,
actual_total_friends);
          } catch (Throwable t) { //Throwable data type
                System.out.println("Error encountered 100..101");
                // java code reports the error
                errCollector.addError(t);
          }
          System.out.println("B");
..
..
```

# Parameterizing

creating data driven tests
method of data storage doesn't matter, Parameterization is done via:
  - @Parameters uses Collection<Object[]>() ← this is a 2D Array
  - @Test uses objects row-by-row: [1][*], [2][*], [3][*]


**Example**

```
    @Parameters
  public static Collection<Object[]> data(){
    //sublass Collection is in superclass ArrayList is subclass
    //Collection a = new ArrayList();
    //a.add("aa");
    //rows - number of times to run the test
    //cols - number of parameters to pass to test
    Object[][] data = new Object[2][3];
    // 1st row
    data[0][0]="testuser1";
    data[0][1]="pass1";
    data[0][2]=7898;
    // 2st row
    data[1][0]="testuser2";
    data[1][1]="pass2";
    data[1][2]=78999;
    //returning a list instead of an array
    return Arrays.asList(data);
```
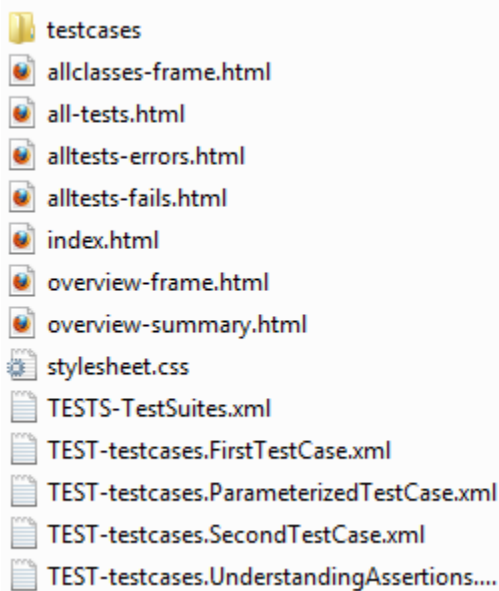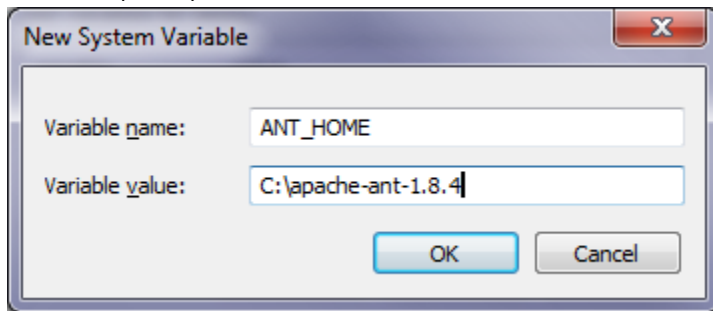
# Test Listeners

## Custom Runners

# Apache Ant

Ant is used for running a test suite via command line and outputting results in XML with HTML pages to review test data results

- testcases
- allclasses-frame.html
- all-tests.html
- alltests-errors.html
- alltests-fails.html
- index.html
- overview-frame.html
- overview-summary.html
- stylesheet.css
- TESTS-TestSuites.xml
- TEST-testcases.FirstTestCase.xml
- TEST-testcases.ParameterizedTestCase.xml
- TEST-testcases.SecondTestCase.xml
- TEST-testcases.UnderstandingAssertions....

## Installation & Config

- Download Links and Reference
  - http://ant.apache.org/bindownload.cgi
  - http://ant.apache.org/manual/
- Install
  - Copy to C:\apache-ant-1.8.4\
  - Environment
    - PATH: "C:\apache-ant-1.8.4\bin"
    - ANT_HOME var: C:\apache-ant-1.8.4\

- Run 'ant' and check for errors

**Error Message on tools.jar**

```
C:\Users\SiloSix>ant
Unable to locate tools.jar. Expected to find it in C:\Program
Files\Java\jre6\lib\tools.jar
Buildfile: build.xml does not exist!
Build failed
```
- Get Tools.jar
  - C:\Program Files\Java\jdk1.6.0_33\lib\tools.jar
  - C:\Program Files\Java\jre6\lib\tools.jar
```
C:\Users\SiloSix>ant
Buildfile: build.xml does not exist!
Build failed
```

# build.xml

- Rresides at project root
- Ant searches for in the java project
- Gives parameters for ant
- Environment vars are ${} not ()
- All Directories must be present prior to compile/run

*Example includes for test classes*

```
<!--include name="**/*TestCase*.class /for all files with TestCase-->
<include name="testcases/FirstTestCase.class" />
<include name="testcases/SecondTestCase.class" />
<include name="testcases/ParameterizedTestCase.class" />
<include name="testcases/UnderstandingAssertions.class" />
```

**Error on Ant Run (Compile OK)**

## Errors while applying transformations: Fatal error

My guess is that the problem is caused by the mismatching XSLT version supported by Ant JUnit plugin (which contains XSLT files that are XSTL 1.0, as you correctly noted) and XSLT processor

(supporting XSLT version 2.0). My guess #2 is that you have the newer XSLT processor (like Saxon) somewhere in classpath so that it's loaded before some other XSLT processor (like Xalan which implements XSLT v1.0). For further troubleshooting I would suggest [url=http://www.google.com/search? q=junitreport+"Errors+while+applying+transformations:+Fatal+error+during+transformation"&btnG=Search] googing it out[/url].

```
D:\Java\Workspace\Module8_Junit_Ant>ant run
Buildfile: D:\Java\Workspace\Module8_Junit_Ant\build.xml

run:
     [junit] Running testcases.FirstTestCase
     [junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.501 sec
     [junit] Running testcases.ParameterizedTestCase
     [junit] Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0.485 sec
     [junit] Running testcases.SecondTestCase
     [junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.434 sec
     [junit] Running testcases.UnderstandingAssertions
     [junit] Tests run: 2, Failures: 3, Errors: 0, Time elapsed: 0.488 sec
     [junit] Test testcases.UnderstandingAssertions FAILED
[junitreport] Processing D:\Java\Workspace\reports\TESTS-TestSuites.xml to
C:\Users\SiloSix\AppData\
Local\Temp\null987575801
[junitreport] Loading stylesheet jar:file:/C:/apache-ant-1.8.4/lib/ant-junit.jar!
/org/apache/tools/a
nt/taskdefs/optional/junit/xsl/junit-frames.xsl
[junitreport] jar:file:/C:/apache-ant-1.8.4/lib/ant-junit.jar!/org/apache/tools/
ant/taskdefs/optiona
l/junit/xsl/junit-frames.xsl:6: Warning! Running an XSLT 1.0 stylesheet with an
XSLT 2.0 processor
[junitreport] jar:file:/C:/apache-ant-1.8.4/lib/ant-junit.jar!/org/apache/tools/
ant/taskdefs/optiona
l/junit/xsl/junit-frames.xsl:38: Fatal Error! Unknown extension instruction
[junitreport] Failed to process D:\Java\Workspace\reports\TESTS-TestSuites.xml

BUILD FAILED
D:\Java\Workspace\Module8_Junit_Ant\build.xml:112: Errors while applying
transformations: Fatal error
r during transformation

Total time: 10 seconds
```

## Compile/Clean/Run

```
D:\Eclipse\Java\Module8_Junit_Ant>ant clean
D:\Eclipse\Java\Module8_Junit_Ant>ant compile
D:\Eclipse\Java\Module8_Junit_Ant>ant run
```

## Creating Ant Batchfile

```
D:\
cd D:\Eclipse\Java\Module8_Junit_Ant
ant clean compile run
```

**Home**

**Packages**

testcases

**Classes**

FirstTestCase
ParameterizedTest(
SecondTestCase
UnderstandingAsse

### Unit Test Results.

Designed for use with JUnit and Ant.

**Summary**

| Tests | Failures | Errors | Success rate | Time |
|-------|----------|--------|--------------|-------|
| 10    | 3        | 0      | 70.00%       | 2.656 |

Note: *failures* are anticipated and checked for with assertions while *errors* are unanticipated.

**Packages**

| Name | Tests | Errors | Failures | Time(s) | Time Stamp | Host |
|------|-------|--------|----------|---------|-----------|------|
| testcases | 10 | 0 | 3 | 2.656 | 2012-08-07T14:49:12 | Silo9 |

## TestNG

- http://testng.org/doc/download.html
- http://search.maven.org/#search|ga|1|a%3A%22testng%22

# Apendices

## JUnit Test Example 1(Simple)

**Test Suite Runner Class (Simple)**

```
package testcases;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
import org.junit.runners.Suite.SuiteClasses;
```

```
@RunWith(Suite.class)
@SuiteClasses({
    FirstTestCase.class,
    SecondTestCase.class
    })

public class MyTestSuiteRunner {

}
```

**Test Case Example 1 (Simple)**

```
package testcases;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

public class SecondTestCase {

    //@Test
    //@Ignore
    //@Before
    //@After
    //@BeforeClass
    //@AfterClass

    @BeforeClass // must be static
    public static void beforeClass(){
       System.out.println("*************Beginning**************");
    }
    @Before
    public void beforeTestCase(){
       //open browser - selenuim
       System.out.println("Opening the browser");
    }

    @Test
    public void sendEmailTest(){
       System.out.println("Testing Sending Email");
    }

    @Test
    public void sendMessageTest(){
```

```
        System.out.println("Testing Sending Message");
    }

    @After
    public void afterTestCase(){
        System.out.println("Closing the browser");
    }

    @AfterClass //must be static
    public static void afterClass(){
        System.out.println("**************Ending**************");
    }
}
```

## JUnit Test Example 2

**MyTestSuiteRunner.java**

```
package testcases;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
import org.junit.runners.Suite.SuiteClasses;

@RunWith(Suite.class)
@SuiteClasses({
    UnderstandingAssertions.class,
    FirstTestCase.class,
    SecondTestCase.class
    })

public class MyTestSuiteRunner {

}
```

**FirstTestCase.java**

```
package testcases;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

public class FirstTestCase {
```

```java
    @BeforeClass
    public static void beforeThisTest(){
       System.out.println("Test Class 01 Starting-------------");
    }

    @Test // test case
    public void loginTest(){
       // selenium code
       System.out.println("1-logging in user");
    }

    @Test
    public void registerTest(){
       System.out.println("2-Registering a user");
    }

    @Test
    public void dataBaseTest(){
       System.out.println("3-Testing the database");
    }

    @AfterClass
    public static void afterThisTest(){
       System.out.println("Test Class 01 Ending-------------");
    }

}
```

## SecondTestCase.java

```java
package testcases;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Assume;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

public class SecondTestCase {

    //@Test
    //@Ignore
    //@Before
    //@After
    //@BeforeClass
```

```java
    //@AfterClass

    // Function that JUnit won't run automatically
    // Must be called within @Tests
    public static boolean checkLogin(){
       return true;
       // true - if login is the success
    }

    @BeforeClass // must be static
    public static void beforeClass(){
       System.out.println("Test Class 02 Starting-------------");

       // false - skip all tests in this class
       // JUnit monitors this boolean for executing the rest of the tests.
       Assume.assumeTrue(checkLogin());
    }

    @AfterClass //must be static
    public static void afterClass(){
       System.out.println("Test Class 02 Ending-------------");
    }

    @Before
    public void beforeTestCase(){
       //open browser - selenuim
       System.out.println("Opening the browser");
    }

    @After
    public void afterTestCase(){
       System.out.println("Closing the browser");
    }

    // TEST CASES****************************************

    @Test
    public void sendEmailTest(){
       System.out.println("1-Testing Sending Email");
    }

    @Test
    public void getMessageTest(){
       // Assume.assumeTrue(false); use to break out of a @Test
       System.out.println("2-Testing Sending Message");

    }

    @Test
```

```
    public void sendMessageTest(){
       System.out.println("3-Testing Receiving Message");
    }


}
```

**ParameterizedTestCase.java**

package testcases;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.Parameterized.Parameters;

//1 step
@RunWith(Parameterized.class)
public class ParameterizedTestCase {

//2 step
   //declare parameters at global level
   public String username;
   public String password;
   public int pin;

//3 step
   //constructor
   public ParameterizedTestCase(
             String username,
             String password,
             int pin
             ){
      this.username = username;
      this.password = password;
      this.pin = pin;
   }
//4 step
   @Parameters
   public static Collection<Object[]> data(){
       //sublass Collection is in superclass ArrayList is subclass
       //Collection a = new ArrayList();
       //a.add("aa");
```

```
        //rows - number of times to run the test
        //cols - number of parameters to pass to test
        Object[][] data = new Object[2][3];
        // 1st row
        data[0][0]="testuser1";
        data[0][1]="pass1";
        data[0][2]=7898;
        // 2st row
        data[1][0]="testuser2";
        data[1][1]="pass2";
        data[1][2]=78999;
        //returning a list instead of an array
        return Arrays.asList(data);
    }



    @Test
    public void testRegister(){
        //selenium
        System.out.println("testing Registration -- " + username + " -- " + password + " -- " + pin);
    }


}
```

**UnderstandingAssertions.java**

```
package testcases;

import junit.framework.Assert;

import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ErrorCollector;

public class UnderstandingAssertions {

    @Rule //JUNIT Rule
    public ErrorCollector errCollector = new ErrorCollector();


    @Test
    public void testFriendlistFacebook(){
        int actual_total_friends =100; //selenium picks up this value from page
        int expected_total_friends=101; //from XLS file
/*      if (actual_total_friends == expected_total_friends) {
```

```
                    System.out.println("Pass");
            } else {
                    System.out.println("Fail");
                    // report error, but need assert for JUnit to pick up failure
            }
    */
            System.out.println("A");
            Assert.assertEquals(expected_total_friends, actual_total_friends);
            //JUnit will pick up @Test failure
            // not executed due to Assert...
            System.out.println("B");
    }


            @Test
            public void testFriendlistFacebook2(){
                    int actual_total_friends =100; //selenium picks up this value from page
                    int expected_total_friends=101; //from XLS file
    /*      if (actual_total_friends == expected_total_friends) {
                            System.out.println("Pass");
                    } else {
                            System.out.println("Fail");
                            // report error, but need assert for JUnit to pick up failure
                    }
    */
                    System.out.println("A");
                    // try catch will catch error
                    // But JUnit won't see the failure
                    try {
                            Assert.assertEquals(expected_total_friends, actual_total_friends);
                    } catch (Throwable t) { //Throwable data type
                            System.out.println("Error encountered 100..101");
                            // java code reports the error
                            errCollector.addError(t);
                    }
                    System.out.println("B");
                    try {
                            Assert.assertEquals("A", "B");
                    } catch (Throwable t) {
                            System.out.println("Error encountered AB");
                            errCollector.addError(t);
                    }
                    try {
                            Assert.assertEquals("A", "A");
                    } catch (Throwable t) {
                            System.out.println("Error encountered AA");
                            errCollector.addError(t);
                    }
                    //assert won't trigger on true
```

```
            Assert.assertTrue("error msg", true);
            Assert.assertTrue("error msg", 6>3);
    }
}
```

---

## Apache ANT Buildfile (build.xml)

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE project [
]>
<project name="Module_Junit_Ant" default="useage" basedir=".">
  <!--  =========== Initialize Properties =============== -->
  <description>
      Apache Ant build.xml file for Selenium tests
  </description>
  <!-- set global properties for this build -->
  <property name="ws.home" value="${basedir}" />
  <property name="ws.jars" value="D:/jars" />
  <!-- when compile the programs.. send .classes to /build -->
  <property name="test.dest" value="${ws.home}/build" />
  <!-- where .java files reside -->
  <property name="test.src" value="${ws.home}/src" />
  <!-- Reports directory -->
  <property name="test.reportsDir" value="D:/Eclipse/Java/reports" />
  <!--  property name="env.ANT_HOME" value="C:/apache-ant-1.8.4/bin"/-->

  <path id="testcase.path">
      <pathelement location="${test.dest}" />
      <fileset dir="${ws.jars}">
      <include name="*.jar" />
      </fileset>
  </path>

  <!-- target name="start-selenium-server">
       <java jar="${ws.home}/lib/selenium-server.jar"/>
  </target-->

  <target name="setClassPath" unless="test.classpath">
      <path id="classpath_jars">
      <fileset dir="${ws.jars}" includes="*.jar" />
      </path>
      <pathconvert pathsep=":" property="test.classpath" refid="classpath_jars" />
  </target>

  <target name="init" depends="setClassPath">
      <tstamp>
```

```
        <format property="start.time" pattern="MM/dd/yyyy hh:mm aa" />
        </tstamp>
        <condition property="ANT" value="${env.ANT_HOME}/bin/ant.bat"
else="${env.ANT_HOME}/bin/ant">
        <os family="windows" />
        </condition>
        <taskdef name="testng" classpath="${test.classpath}"
classname="org.testng.TestNGAntTask" />
    </target>
    <!--  all  -->
    <target name="all"></target>

    <!-- clean -->
    <target name="clean">
        <delete dir="${test.dest}" />
    </target>

    <!-- compile -->
    <target name="compile" depends="init, clean">
        <!--  delete dir"", even empty directories -->
        <delete includeemptydirs="true" quiet="true">
        <!-- include="**/*" means all files and subdirs -->
        <fileset dir="${test.dest}" includes="**/*" />
        </delete>
        <echo message="making directory..." />
        <!--  where to put the .class compiled files  -->
        <mkdir dir="${test.dest}" />
        <echo message="classpath-----: ${test.classpath}" />
        <echo message="compiling..." />
        <javac debug="true" destdir="${test.dest}" srcdir="${test.src}"
target="1.5" classpath="${test.classpath}"></javac>
    </target>

    <!--  build -->
    <target name="build" depends="init"></target>
    <target name="usage">
        <echo>
              ant run will execute the test
        </echo>
    </target>

    <path id="test.c">
        <fileset dir="${ws.jars}" includes="*.jar" />
    </path>
<!--  Executes all the tests "run" -->
    <target name="run">
        <!--  delete dir"", even empty directories -->
        <delete includeemptydirs="true" quiet="true">
        <!-- include="**/*" means all files and subdirs -->
```

```xml
        <fileset dir="${test.reportsDir}" includes="**/*" />
    </delete>
    <java jar="${ws.jars}" fork="true" spawn="true" />
    <junit fork="yes" haltonfailure="no" printsummary="yes">
    <classpath refid="testcase.path" />
    <!-- classpath ="${test.classpath}"/> -->


<!-- 1-Run test in the fileset -->


    <batchtest todir="${test.reportsDir}" fork="true">
    <fileset dir="${test.dest}">
        <!--include name="testcases/FirstTest.class" /-->
        <!--include name="testcases/SecondTest.class"/-->
        <include name="testcases/FirstTestCase.class" />
        <include name="testcases/SecondTestCase.class" />
        <include name="testcases/ParameterizedTestCase.class" />
        <include name="testcases/UnderstandingAssertions.class" />
        <!--  exclude name="**/AbstractSeleneseTestCase.class"/ -->
    </fileset>
    </batchtest>
    <formatter type="xml" />
    <classpath refid="testcase.path" />
    </junit>
<!-- Send JUNIT reports to this directory -->
    <junitreport todir="${test.reportsDir}">
    <fileset dir="${test.reportsDir}">
    <include name="TEST-*.xml" />
    </fileset>
    <report todir="${test.reportsDir}" />
    </junitreport>
  </target>
</project>
```

## Apache Ant Compile/Clean

```
D:\Eclipse\Java\Module8_Junit_Ant>ant clean
Buildfile: D:\Eclipse\Java\Module8_Junit_Ant\build.xml

clean:
   [delete] Deleting directory D:\Eclipse\Java\Module8_Junit_Ant\build

BUILD SUCCESSFUL
Total time: 1 second
```

```
D:\Eclipse\Java\Module8_Junit_Ant>ant compile
Buildfile: D:\Eclipse\Java\Module8_Junit_Ant\build.xml
```

```
setClassPath:

init:

clean:

compile:
     [echo] making directory...
     [mkdir] Created dir: D:\Eclipse\Java\Module8_Junit_Ant\build
     [echo] classpath-----: D:\jars\commons-io-2.4.jar:D:\jars\jdom-2.0.2.jar:D:
\jars\junit-4.11.jar:D:\jars\jxl.jar:D:\jars\selenium-server-standalone-2.25.0.j
ar:D:\jars\testng-6.7.jar:D:\jars\tools.jar
     [echo] compiling...
     [javac] D:\Eclipse\Java\Module8_Junit_Ant\build.xml:55: warning: 'includeant
runtime' was not set, defaulting to build.sysclasspath=last; set to false for re
peatable builds
     [javac] Compiling 5 source files to D:\Eclipse\Java\Module8_Junit_Ant\build

BUILD SUCCESSFUL
Total time: 6 seconds
```

## Run

- C:\Eclipse\Project>ant run

```
D:\Eclipse\Java\Module8_Junit_Ant>ant run
Buildfile: D:\Eclipse\Java\Module8_Junit_Ant\build.xml

run:
     [junit] Running testcases.FirstTestCase
     [junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.659 sec
     [junit] Running testcases.ParameterizedTestCase
     [junit] Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0.684 sec
     [junit] Running testcases.SecondTestCase
     [junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.637 sec
     [junit] Running testcases.UnderstandingAssertions
     [junit] Tests run: 2, Failures: 3, Errors: 0, Time elapsed: 0.676 sec
     [junit] Test testcases.UnderstandingAssertions FAILED
[junitreport] Processing D:\Eclipse\Java\reports\TESTS-TestSuites.xml to C:\User
s\SiloSix\AppData\Local\Temp\null276535517
[junitreport] Loading stylesheet jar:file:/C:/apache-ant-1.8.4/lib/ant-junit.jar
!/org/apache/tools/ant/taskdefs/optional/junit/xsl/junit-frames.xsl
[junitreport] Transform time: 2927ms
[junitreport] Deleting: C:\Users\SiloSix\AppData\Local\Temp\null276535517

BUILD SUCCESSFUL
Total time: 15 seconds
```

**Home**

**Packages**

testcases

**Classes**

FirstTestCase
ParameterizedTest
SecondTestCase
UnderstandingAsse

## Unit Test Results.

Designed for use with JUnit and Ant.

### Summary

| Tests | Failures | Errors | Success rate | Time |
|-------|----------|--------|--------------|------|
| **10** | **3** | **0** | 70.00% | 2.656 |

Note: *failures* are anticipated and checked for with assertions while *errors* are unanticipated.

### Packages

| Name | Tests | Errors | Failures | Time(s) | Time Stamp | Host |
|------|-------|--------|----------|---------|------------|------|
| **testcases** | 10 | 0 | 3 | 2.656 | 2012-08-07T14:49:12 | Silo9 |

# Properties of testcases.FirstTestCase

Close

| Name | Value |
|------|-------|
| ant.core.lib | C:\apache-ant-1.8.4\lib\ant.jar |
| ant.file | D:\Eclipse\Java\Module8_Junit_Ant\build.xml |
| ant.file.Module_Junit_Ant | D:\Eclipse\Java\Module8_Junit_Ant\build.xml |
| ant.file.type | file |
| ant.file.type.Module_Junit_Ant | file |
| ant.home | C:\apache-ant-1.8.4\bin\.. |
| ant.java.version | 1.6 |
| ant.library.dir | C:\apache-ant-1.8.4\lib |
| ant.project.default-target | useage |
| ant.project.invoked-targets | run |
| ant.project.name | Module_Junit_Ant |
| ant.version | Apache Ant(TM) version 1.8.4 compiled on May 22 2012 |
| awt.toolkit | sun.awt.windows.WToolkit |
| basedir | D:\Eclipse\Java\Module8_Junit_Ant |
| file.encoding | Cp1252 |
| file.encoding.pkg | sun.io |
| file.separator | \ |
| java.awt.graphicsenv | sun.awt.Win32GraphicsEnvironment |
| java.awt.printerjob | sun.awt.windows.WPrinterJob |

| | |
|---|---|
| java.class.path | D:\Eclipse\Java\Module8_Junit_Ant\build;D:\jars\commons-io-2.4.jar;D:\jars\jdom-2.0.2.jar;D:\jars\junit-4.11.jar;D:\jars\jxl.jar;D:\jars\selenium-server-standalone-2.25.0.jar;D:\jars\testng-6.7.jar;D:\jars\tools.jar;D:\Eclipse\Java\Module8_Junit_Ant;C:\Program Files (x86)\QuickTime\QTSystem\QTJava.zip;C:\apache-ant-1.8.4\lib\ant-launcher.jar;C:\apache-ant-1.8.4\lib\ant.jar;C:\apache-ant-1.8.4\lib\ant-junit.jar;C:\apache-ant-1.8.4\lib\ant-junit4.jar |
| java.class.version | 50.0 |
| java.endorsed.dirs | C:\Program Files\Java\jre6\lib\endorsed |
| java.ext.dirs | C:\Program Files\Java\jre6\lib\ext;C:\Windows\Sun\Java\lib\ext |
| java.home | C:\Program Files\Java\jre6 |
| java.io.tmpdir | C:\Users\SiloSix\AppData\Local\Temp\ |
| java.library.path | C:\Program Files\Java\jre6\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\ATI Technologies\ATI.ACE\Core-Static;C:\Program Files (x86)\Common Files\Acronis\SnapAPI\;C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\DTS\Binn\;C:\Program Files (x86)\QuickTime\QTSystem\;C:\Program Files (x86)\Android\android-sdk\tools;C:\Program Files (x86)\Android\android-sdk\platform-tools;C:\Program Files (x86)\Java\jre6\bin;C:\Program Files (x86)\Java\jre6\bin\client;D:\Python27;C:\Program Files\TortoiseSVN\bin;C:\Program Files (x86)\Mozilla Firefox;C:\Program Files\Java\jdk1.6.0_33\bin;C:\apache-ant-1.8.4\bin;;. |
| java.runtime.name | Java(TM) SE Runtime Environment |
| java.runtime.version | 1.6.0_33-b03 |
| java.specification.name | Java Platform API Specification |
| java.specification.vendor | Sun Microsystems Inc. |
| java.specification.version | 1.6 |
| java.vendor | Sun Microsystems Inc. |
| java.vendor.url | http://java.sun.com/ |
| java.vendor.url.bug | http://java.sun.com/cgi-bin/bugreport.cgi |
| java.version | 1.6.0_33 |
| java.vm.info | mixed mode |
| java.vm.name | Java HotSpot(TM) 64-Bit Server VM |
| java.vm.specification.name | Java Virtual Machine Specification |
| java.vm.specification.vendor | Sun Microsystems Inc. |
| java.vm.specification.version | 1.0 |
| java.vm.vendor | Sun Microsystems Inc. |
| java.vm.version | 20.8-b03 |
| line.separator | |
| os.arch | amd64 |

| os.name | Windows 7 |
|---|---|
| os.version | 6.1 |
| path.separator | ; |
| sun.arch.data.model | 64 |
| sun.boot.class.path | C:\Program Files\Java\jre6\lib\resources.jar;C:\Program Files\Java\jre6\lib\rt.jar;C:\Program Files\Java\jre6\lib\sunrsasign.jar;C:\Program Files\Java\jre6\lib\jsse.jar;C:\Program Files\Java\jre6\lib\jce.jar;C:\Program Files\Java\jre6\lib\charsets.jar;C:\Program Files\Java\jre6\lib\modules\jdk.boot.jar;C:\Program Files\Java\jre6\classes |
| sun.boot.library.path | C:\Program Files\Java\jre6\bin |
| sun.cpu.endian | little |
| sun.cpu.isalist | amd64 |
| sun.desktop | windows |
| sun.io.unicode.encoding | UnicodeLittle |
| sun.java.command | org.apache.tools.ant.taskdefs.optional.junit.JUnitTestRunner testcases.FirstTestCase filtertrace=true haltOnError=false haltOnFailure=false formatter=org.apache.tools.ant.taskdefs.optional.junit.SummaryJUnitResultFormatter showoutput=false outputtoformatters=true logfailedtests=true logtestlistenerevents=false formatter=org.apache.tools.ant.taskdefs.optional.junit.XMLJUnitResultFormatter,D:\Eclipse\Java\reports\TEST-testcases.FirstTestCase.xml crashfile=D:\Eclipse\Java\Module8_Junit_Ant\junitvmwatcher1328204574902432774.properties propsfile=D:\Eclipse\Java\Module8_Junit_Ant\junit7241907459533418942.properties |
| sun.java.launcher | SUN_STANDARD |
| sun.jnu.encoding | Cp1252 |
| sun.management.compiler | HotSpot 64-Bit Tiered Compilers |
| sun.os.patch.level | Service Pack 1 |
| test.dest | D:\Eclipse\Java\Module8_Junit_Ant/build |
| test.reportsDir | D:/Eclipse/Java/reports |
| test.src | D:\Eclipse\Java\Module8_Junit_Ant/src |
| user.country | US |
| user.dir | D:\Eclipse\Java\Module8_Junit_Ant |
| user.home | C:\Users\SiloSix |
| user.language | en |
| user.name | SiloSix |
| user.timezone |  |
| user.variant |  |
| ws.home | D:\Eclipse\Java\Module8_Junit_Ant |
| ws.jars | D:/jars |