

JUnit - Plug with ANT

We will have an example to demonstrate how to run JUnit using ANT. Follow the steps given below.

Step 1: Download Apache Ant

Download [Apache Ant](#) based on the operating system you are working on.

OS	Archive Name
Windows	apache-ant-1.8.4-bin.zip
Linux	apache-ant-1.8.4-bin.tar.gz
Mac	apache-ant-1.8.4-bin.tar.gz

Step 2: Set Ant Environment

Set the **ANT_HOME** environment variable to point to the base directory location, where the ANT libraries are stored on your machine. Let us assume the Ant libraries are stored in the folder apache-ant-1.8.4.

Sr.No.	OS & Description
1	Windows Set the environment variable ANT_HOME to C:\Program Files\Apache Software Foundation\apache-ant-1.8.4
2	Linux export ANT_HOME = /usr/local/apache-ant-1.8.4
3	Mac export ANT_HOME = /Library/apache-ant-1.8.4

Append Ant compiler location to the System Path as follows –

OS	Output
Windows	Append the string %ANT_HOME\bin at the end of the system variable, Path .
Linux	export PATH = \$PATH:\$ANT_HOME/bin/

Mac	not required
-----	--------------

Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

Step 3: Download JUnit Archive

Download a JUnit Archive that suits your operating system.

OS	Archive Name
Windows	junit4.10.jar
Linux	junit4.10.jar
Mac	junit4.10.jar

Step 4: Create Project Structure

Create a folder **TestJUnitWithAnt** in C:\>JUNIT_WORKSPACE.

Create a folder **src** in C:\>JUNIT_WORKSPACE>TestJUnitWithAnt.

Create a folder **test** in C:\>JUNIT_WORKSPACE>TestJUnitWithAnt.

Create a folder **lib** in C:\>JUNIT_WORKSPACE>TestJUnitWithAnt.

Create **MessageUtil** class in C:\>JUNIT_WORKSPACE>TestJUnitWithAnt>srcfolder.

```
/*  
 * This class prints the given message on console.  
 */
```

```
public class MessageUtil {  
  
    private String message;  
  
    //Constructor  
    //@param message to be printed  
    public MessageUtil(String message){  
        this.message = message;  
    }  
  
    // prints the message
```

```

public String printMessage(){
    System.out.println(message);
    return message;
}

// add "Hi!" to the message
public String salutationMessage(){
    message = "Hi!" + message;
    System.out.println(message);
    return message;
}
}

```

Create **TestMessageUtil** class in the folder C:
 \>JUNIT_WORKSPACE>TestJUnitWithAnt>src.

```

import org.junit.Test;
import org.junit.Ignore;
import static org.junit.Assert.assertEquals;

public class TestMessageUtil {

    String message = "Robert";
    MessageUtil messageUtil = new MessageUtil(message);

    @Test
    public void testPrintMessage() {
        System.out.println("Inside testPrintMessage()");
        assertEquals(message,messageUtil.printMessage());
    }

    @Test
    public void testSalutationMessage() {
        System.out.println("Inside testSalutationMessage()");
        message = "Hi!" + "Robert";
        assertEquals(message,messageUtil.salutationMessage());
    }
}

```

Copy junit-4.10.jar onto the folder C:\>JUNIT_WORKSPACE>TestJUnitWithAnt>lib.

Create ANT Build.xml

We'll be using **<junit>** task in Ant to execute our JUnit test cases.

```
<project name = "JUnitTest" default = "test" basedir = ".">
  <property name = "testdir" location = "test" />
  <property name = "srcdir" location = "src" />
  <property name = "full-compile" value = "true" />

  <path id = "classpath.base"/>

  <path id = "classpath.test">
    <pathelement location = "lib/junit-4.10.jar" />
    <pathelement location = "${testdir}" />
    <pathelement location = "${srcdir}" />
    <path refid = "classpath.base" />
  </path>

  <target name = "clean" >
    <delete verbose = "${full-compile}">
      <fileset dir = "${testdir}" includes = "**/*.class" />
    </delete>
  </target>

  <target name = "compile" depends = "clean">
    <javac srcdir = "${srcdir}" destdir = "${testdir}"
      verbose = "${full-compile}">
      <classpath refid = "classpath.test"/>
    </javac>
  </target>

  <target name = "test" depends = "compile">
    <junit>
      <classpath refid = "classpath.test" />
      <formatter type = "brief" usefile = "false" />
      <test name = "TestMessageUtil" />
    </junit>
  </target>

</project>
```

Run the following Ant command.

```
C:\JUNIT_WORKSPACE\TestJUnitWithAnt>ant
```

Verify the output.

Buildfile: C:\JUNIT_WORKSPACE\TestJUnitWithAnt\build.xml

clean:

compile:

```
[javac] Compiling 2 source files to C:\JUNIT_WORKSPACE\TestJUnitWithAnt\test
[javac] [parsing started C:\JUNIT_WORKSPACE\TestJUnitWithAnt\src\
    MessageUtil.java]
[javac] [parsing completed 18ms]
[javac] [parsing started C:\JUNIT_WORKSPACE\TestJUnitWithAnt\src\
    TestMessageUtil.java]
[javac] [parsing completed 2ms]
[javac] [search path for source files: C:\JUNIT_WORKSPACE\
    TestJUnitWithAnt\src]
[javac] [loading java\lang\Object.class(java\lang:Object.class)]
[javac] [loading java\lang\String.class(java\lang:String.class)]
[javac] [loading org\junit\Test.class(org\junit:Test.class)]
[javac] [loading org\junit\Ignore.class(org\junit:Ignore.class)]
[javac] [loading org\junit\Assert.class(org\junit:Assert.class)]
[javac] [loading java\lang\annotation\Retention.class
    (java\lang\annotation:Retention.class)]
[javac] [loading java\lang\annotation\RetentionPolicy.class
    (java\lang\annotation:RetentionPolicy.class)]
[javac] [loading java\lang\annotation\Target.class
    (java\lang\annotation:Target.class)]
[javac] [loading java\lang\annotation\ElementType.class
    (java\lang\annotation:ElementType.class)]
[javac] [loading java\lang\annotation\Annotation.class
    (java\lang\annotation:Annotation.class)]
[javac] [checking MessageUtil]
[javac] [loading java\lang\System.class(java\lang:System.class)]
[javac] [loading java\io\PrintStream.class(java\io:PrintStream.class)]
[javac] [loading java\io\FilterOutputStream.class
    (java\io:FilterOutputStream.class)]
[javac] [loading java\io\OutputStream.class(java\io:OutputStream.class)]
[javac] [loading java\lang\StringBuilder.class
    (java\lang:StringBuilder.class)]
[javac] [loading java\lang\AbstractStringBuilder.class
    (java\lang:AbstractStringBuilder.class)]
[javac] [loading java\lang\CharSequence.class(java\lang:CharSequence.class)]
```

```
[javac] [loading java\io\Serializable.class(java\io:Serializable.class)]
[javac] [loading java\lang\Comparable.class(java\lang:Comparable.class)]
[javac] [loading java\lang\StringBuffer.class(java\lang:StringBuffer.class)]
[javac] [wrote C:\JUNIT_WORKSPACE\TestJUnitWithAnt\test\MessageUtil.class]
[javac] [checking TestMessageUtil]
[javac] [wrote C:\JUNIT_WORKSPACE\TestJUnitWithAnt\test\TestMessageUtil.class]
[javac] [total 281ms]
```

test:

```
[junit] Testsuite: TestMessageUtil
[junit] Tests run: 2, Failures: 0, Errors: 0, Time elapsed: 0.008 sec
[junit]
[junit] ----- Standard Output -----
[junit] Inside testPrintMessage()
[junit] Robert
[junit] Inside testSalutationMessage()
[junit] Hi!Robert
[junit] -----
```

BUILD SUCCESSFUL

Total time: 0 seconds