# Taking Screenshots of JUnit Test Failures

FEST-Swing can take a screenshot of the desktop when a JUnit GUI test fails, either when running tests using Ant or inside an IDE (e.g. Eclipse.)

To take screenshots of failed GUI tests, regardless of how they are executed, please follow these steps:

1. Add the JUnit Extension to your classpath (file fest-swing-junit-{VERSION}.jar, included when you download FEST-Swing)
2. Mark GUI tests (class or method level) with the annotation `org.fest.swing.annotation.GUITest`

FEST-Swing's JUnit extension requires **JUnit 4.3.1**.

> **Changes in FEST-Swing 1.2**
>
> Starting with FEST-Swing 1.2, we have added support for JUnit 4.5. Due to JUnit's limited support for extensibility, we have changed the structure of the JUnit extension:
>
> 1. *fest-swing-junit-{VERSION}.jar*. Contains support for Ant and utility methods to support FEST's JUnit custom `Runner`. This jar should **always** be present when using FEST's JUnit extension.
> 2. *fest-swing-junit-4.3.1-{VERSION}.jar*. Contains FEST's `Runner` for JUnit 4.3.1, `org.fest.swing.junit.v4_3_1.runner.GUITestRunner`.
> 3. *fest-swing-junit-4.5-{VERSION}.jar*. Contains FEST's `Runner` for JUnit 4.5, `org.fest.swing.junit.v4_5.runner.GUITestRunner`.
>
> For more details, please read Jar Files Explained.

### Running GUI tests with Ant

In order to take screenshots of failed GUI tests with Ant please follow these steps:

1. Add a definition of the Ant task `festreport`
2. Use the formatter `org.fest.swing.junit.ant.ScreenshotOnFailureResultFormatter` inside the `junit` Ant task
3. Use the Ant task `festreport` instead of `junitreport`, and specify in its classpath where the fest-swing-junit-{VERSION}.jar file is.

#### Example

```
<target name="test" depends="compile">
  <taskdef resource="festjunittasks"
classpathref="lib.classpath" />

  <junit forkmode="perBatch"
printsummary="yes" haltonfailure="no"
haltonerror="no">
```

```
    <classpath refid="lib.classpath" />
    <classpath
location="${target.test.classes.dir}" />
    <classpath
location="${target.classes.dir}" />
    <formatter
classname="org.fest.swing.junit.ant.Screensh
otOnFailureResultFormatter" extension=".xml"
/>
    <batchtest fork="yes"
todir="${target.junit.results.dir}">
      <fileset
dir="${target.test.classes.dir}"
includes="**/*Test*.class" />
    </batchtest>
  </junit>

  <festreport
todir="${target.junit.report.dir}">
    <classpath refid="lib.classpath" />
    <fileset
dir="${target.junit.results.dir}">
      <include name="TEST-*.xml" />
    </fileset>
    <report format="frames"
```

```
todir="${target.junit.report.dir}/html" />
  </festreport>
</target>
```

The following is a screenshot of a JUnit HTML report:

The Ant task `festreport` works exactly as `junitreport`: it can generate HTML reports with or without frames. It has been tested with **Ant 1.7** and requires Apache-Commons Codec 1.3.

### Running GUI tests in an IDE

FEST-Swing also provides a custom JUnit `Runner`, `org.fest.swing.junit.GUITestRunner`, which takes screenshots of failed GUI tests. To use it, just annotate your test class with `@RunWith(GUITestRunner.class)`. Screenshots of failed tests will be saved in the directory "failed-gui-tests" (relative to the directory where tests are executed.)

`GUITestRunner` has been tested with Eclipse 3.4.1. It may not work with certain versions of IntelliJ IDEA due to bug IDEA-13389.