

### About Jd2chm...

The project files of this HTML Help  
Were generated with Jd2chm version 0.3  
Copyright © 2000-2003 [Andre Burgaud](http://www.burgaud.com)  
<http://www.burgaud.com>

# Packages

[junit.extensions](#)

[junit.framework](#)



# Hierarchy For All Packages

**Package Hierarchies:**

[junit.extensions](#), [junit.framework](#)

---

# Class Hierarchy

- class java.lang.Object
  - class junit.framework.[Assert](#)
    - class junit.framework.[TestCase](#) (implements junit.framework.[Test](#))
      - class junit.extensions.[ExceptionTestCase](#)
    - class junit.extensions.[TestDecorator](#) (implements junit.framework.[Test](#))
      - class junit.extensions.[RepeatedTest](#)
      - class junit.extensions.[TestSetup](#)
  - class junit.framework.[TestFailure](#)
  - class junit.framework.[TestResult](#)
  - class junit.framework.[TestSuite](#) (implements junit.framework.[Test](#))
    - class junit.extensions.[ActiveTestSuite](#)
  - class java.lang.Throwable (implements java.io.Serializable)
    - class java.lang.Error
      - class junit.framework.[AssertionFailedError](#)
        - class junit.framework.[ComparisonFailure](#)

# Interface Hierarchy

- interface junit.framework.[Protectable](#)
- interface junit.framework.[Test](#)
- interface junit.framework.[TestListener](#)

---

---

[Overview](#) Package Class  [Deprecated](#) [Index](#) [Help](#)

PREV NEXT

[FRAMES](#) [NO FRAMES](#)

---

---

**Overview**

**Class Tree Deprecated Index Help**

PREV PACKAGE [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

---

## Package junit.extensions

### Class Summary

<a href="#">ActiveTestSuite</a>	A TestSuite for active Tests.
<a href="#">ExceptionTestCase</a>	A TestCase that expects an Exception of class fExpected to be thrown.
<a href="#">RepeatedTest</a>	A Decorator that runs a test repeatedly.
<a href="#">TestDecorator</a>	A Decorator for Tests.
<a href="#">TestSetup</a>	A Decorator to set up and tear down additional fixture state.

---

[Overview](#)

Class [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

---





# Hierarchy For Package junit.extensions

Package Hierarchies:

[All Packages](#)

---

# Class Hierarchy

- class java.lang.Object
  - class junit.framework.[Assert](#)
    - class junit.framework.[TestCase](#) (implements junit.framework.[Test](#))
      - class junit.extensions.[ExceptionTestCase](#)
    - class junit.extensions.[TestDecorator](#) (implements junit.framework.[Test](#))
      - class junit.extensions.[RepeatedTest](#)
      - class junit.extensions.[TestSetup](#)
  - class junit.framework.[TestSuite](#) (implements junit.framework.[Test](#))
    - class junit.extensions.[ActiveTestSuite](#)

---

[Overview](#) [Package](#) [Class](#)  [Deprecated](#) [Index](#) [Help](#)

PREV [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.extensions **Class ActiveTestSuite**

```
java.lang.Object
|
+-- junit.framework.TestSuite
    |
    +-- junit.extensions.ActiveTestSuite
```

### All Implemented Interfaces:

[Test](#)

---

```
public class ActiveTestSuite
extends TestSuite
```

A TestSuite for active Tests. It runs each test in a separate thread and waits until all threads have terminated. -- Aarhus Radisson Scandinavian Center 11th floor

---

## Constructor Summary

[ActiveTestSuite](#)()

[ActiveTestSuite](#)(java.lang.Class theClass)

[ActiveTestSuite](#)(java.lang.Class theClass, java.lang.String name)

[ActiveTestSuite](#)(java.lang.String name)

## Method Summary

void	<a href="#">run</a> ( <a href="#">TestResult</a> result) Runs the tests and collects their result in a TestResult.
------	---

void	<a href="#">runFinished</a> ( <a href="#">Test</a> test)
------	--

```
void runTest(Test test, TestResult result)
```

### Methods inherited from class [junit.framework.TestSuite](#)

[addTest](#), [addTestSuite](#), [countTestCases](#), [createTest](#), [getName](#),  
[getTestConstructor](#), [setName](#), [testAt](#), [testCount](#), [tests](#), [toString](#)

### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),  
[wait](#), [wait](#), [wait](#)

## Constructor Detail

### ActiveTestSuite

```
public ActiveTestSuite()
```

---

### ActiveTestSuite

```
public ActiveTestSuite(java.lang.Class theClass)
```

---

### ActiveTestSuite

```
public ActiveTestSuite(java.lang.String name)
```

---

### ActiveTestSuite

```
public ActiveTestSuite(java.lang.Class theClass,  
                       java.lang.String name)
```

## Method Detail

## run

```
public void run(TestResult result)
```

**Description copied from class:** [TestSuite](#)

Runs the tests and collects their result in a [TestResult](#).

**Specified by:**

[run](#) in interface [Test](#)

**Overrides:**

[run](#) in class [TestSuite](#)

---

## runTest

```
public void runTest(Test test,  
                   TestResult result)
```

**Overrides:**

[runTest](#) in class [TestSuite](#)

---

## runFinished

```
public void runFinished(Test test)
```

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## junit.extensions **Class ExceptionTestCase**

```
java.lang.Object
|
+-- junit.framework.Assert
    |
    +-- junit.framework.TestCase
        |
        +-- junit.extensions.ExceptionTestCase
```

### All Implemented Interfaces:

[Test](#)

---

```
public class ExceptionTestCase
extends TestCase
```

A TestCase that expects an Exception of class fExpected to be thrown. The other way to check that an expected exception is thrown is:

```
try {
    shouldThrow();
}
catch (SpecialException e) {
    return;
}
fail("Expected SpecialException");
```

To use ExceptionTestCase, create a TestCase like:

```
new ExceptionTestCase("testShouldThrow", SpecialException.class);
```

---

## **Constructor Summary**

```
ExceptionTestCase(java.lang.String name,
java.lang.Class exception)
```

---

## Method Summary

protected void	<a href="#">runTest()</a> Execute the test method expecting that an Exception of class fExpected or one of its subclasses will be thrown
-------------------	---

### Methods inherited from class junit.framework.TestCase

[countTestCases](#), [createResult](#), [getName](#), [run](#), [run](#), [runBare](#), [setName](#), [setUp](#), [tearDown](#), [toString](#)

### Methods inherited from class junit.framework.Assert

[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertFalse](#), [assertFalse](#), [assertNotNull](#), [assertNotNull](#),  
[assertNotSame](#), [assertNotSame](#), [assertNull](#), [assertNull](#), [assertSame](#),  
[assertSame](#), [assertTrue](#), [assertTrue](#), [fail](#), [fail](#)

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),  
[wait](#), [wait](#), [wait](#)

## Constructor Detail

### ExceptionTestCase

```
public ExceptionTestCase(java.lang.String name,  
                          java.lang.Class exception)
```

## Method Detail

### runTest

```
protected void runTest()  
    throws java.lang.Throwable
```

Execute the test method expecting that an Exception of class `fExpected` or one of its subclasses will be thrown

**Overrides:**

[runTest](#) in class [TestCase](#)

**Throws:**

`java.lang.Throwable` - if any exception is thrown

---

---

[Overview](#) [Package](#)

[Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.extensions **Class RepeatedTest**

```
java.lang.Object
|
+-- junit.framework.Assert
    |
    +-- junit.extensions.TestDecorator
        |
        +-- junit.extensions.RepeatedTest
```

### All Implemented Interfaces:

[Test](#)

---

```
public class RepeatedTest
extends TestDecorator
```

A Decorator that runs a test repeatedly.

## Field Summary

Fields inherited from class junit.extensions.[TestDecorator](#)

[fTest](#)

## Constructor Summary

[RepeatedTest](#)([Test](#) test, int repeat)

## Method Summary

int	<a href="#">countTestCases</a> () Counts the number of test cases that will be run by this test.
void	<a href="#">run</a> ( <a href="#">TestResult</a> result) Runs a test and collects its result in a TestResult

	instance.
java.lang.String	<a href="#">toString()</a>

**Methods inherited from class junit.extensions.[TestDecorator](#)**  
[basicRun](#), [getTest](#)

**Methods inherited from class junit.framework.[Assert](#)**  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertFalse](#), [assertFalse](#), [assertNotNull](#), [assertNotNull](#),  
[assertNotSame](#), [assertNotSame](#), [assertNull](#), [assertNull](#), [assertSame](#),  
[assertSame](#), [assertTrue](#), [assertTrue](#), [fail](#), [fail](#)

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
wait, wait, wait

## Constructor Detail

### RepeatedTest

```
public RepeatedTest(Test test,
                   int repeat)
```

## Method Detail

### countTestCases

```
public int countTestCases()
```

**Description copied from interface: [Test](#)**  
Counts the number of test cases that will be run by this test.

**Specified by:**

[countTestCases](#) in interface [Test](#)

**Overrides:**

[countTestCases](#) in class [TestDecorator](#)

---

## run

```
public void run(TestResult result)
```

**Description copied from interface:** [Test](#)

Runs a test and collects its result in a [TestResult](#) instance.

**Specified by:**

[run](#) in interface [Test](#)

**Overrides:**

[run](#) in class [TestDecorator](#)

---

## toString

```
public java.lang.String toString()
```

**Overrides:**

[toString](#) in class [TestDecorator](#)

---

[Overview](#) [Package](#)

[Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## junit.extensions **Class TestDecorator**

```
java.lang.Object
|
+-- junit.framework.Assert
    |
    +-- junit.extensions.TestDecorator
```

### All Implemented Interfaces:

[Test](#)

### Direct Known Subclasses:

[RepeatedTest](#), [TestSetup](#)

---

```
public class TestDecorator
extends Assert
implements Test
```

A Decorator for Tests. Use TestDecorator as the base class for defining new test decorators. Test decorator subclasses can be introduced to add behaviour before or after a test is run.

---

## Field Summary

protected <a href="#">Test</a>	<a href="#">fTest</a>
-----------------------------------	-----------------------

## Constructor Summary

<a href="#">TestDecorator</a> ( <a href="#">Test</a> test)
--

## Method Summary

void	<a href="#">basicRun</a> ( <a href="#">TestResult</a> result) The basic run behaviour.
------	---

int	<a href="#">countTestCases()</a> Counts the number of test cases that will be run by this test.
<a href="#">Test</a>	<a href="#">getTest()</a>
void	<a href="#">run(<a href="#">TestResult</a> result)</a> Runs a test and collects its result in a <a href="#">TestResult</a> instance.
java.lang.String	<a href="#">toString()</a>

<b>Methods inherited from class <a href="#">junit.framework.Assert</a></b>	
<a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertEquals</a> , <a href="#">assertFalse</a> , <a href="#">assertFalse</a> , <a href="#">assertNotNull</a> , <a href="#">assertNotNull</a> , <a href="#">assertNotSame</a> , <a href="#">assertNotSame</a> , <a href="#">assertNull</a> , <a href="#">assertNull</a> , <a href="#">assertSame</a> , <a href="#">assertSame</a> , <a href="#">assertTrue</a> , <a href="#">assertTrue</a> , <a href="#">fail</a> , <a href="#">fail</a>	

<b>Methods inherited from class <a href="#">java.lang.Object</a></b>	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait	

## Field Detail

### fTest

protected [Test](#) fTest

## Constructor Detail

### TestDecorator

public [TestDecorator](#)([Test](#) test)

# Method Detail

## basicRun

```
public void basicRun(TestResult result)
```

The basic run behaviour.

---

## countTestCases

```
public int countTestCases()
```

**Description copied from interface: [Test](#)**  
Counts the number of test cases that will be run by this test.

**Specified by:**  
[countTestCases](#) in interface [Test](#)

---

## run

```
public void run(TestResult result)
```

**Description copied from interface: [Test](#)**  
Runs a test and collects its result in a `TestResult` instance.

**Specified by:**  
[run](#) in interface [Test](#)

---

## toString

```
public java.lang.String toString()
```

**Overrides:**  
`toString` in class `java.lang.Object`

---

# getTest

```
public Test getTest()
```

---

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.extensions **Class TestSetup**

```
java.lang.Object
|
+-- junit.framework.Assert
    |
    +-- junit.extensions.TestDecorator
        |
        +-- junit.extensions.TestSetup
```

### All Implemented Interfaces:

[Test](#)

---

```
public class TestSetup
extends TestDecorator
```

A Decorator to set up and tear down additional fixture state. Subclass `TestSetup` and insert it into your tests when you want to set up additional state once before the tests are run.

---

## Field Summary

Fields inherited from class junit.extensions.[TestDecorator](#)

[fTest](#)

## Constructor Summary

[TestSetup](#)([Test](#) test)

## Method Summary

void	<a href="#">run</a> ( <a href="#">TestResult</a> result) Runs a test and collects its result in a <code>TestResult</code> instance.
------	--

protected void	<a href="#">setUp()</a> Sets up the fixture.
-------------------	---

protected void	<a href="#">tearDown()</a> Tears down the fixture.
-------------------	---

### Methods inherited from class junit.extensions.[TestDecorator](#)

[basicRun](#), [countTestCases](#), [getTest](#), [toString](#)

### Methods inherited from class junit.framework.[Assert](#)

[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertFalse](#), [assertFalse](#), [assertNotNull](#), [assertNotNull](#),  
[assertNotSame](#), [assertNotSame](#), [assertNull](#), [assertNull](#), [assertSame](#),  
[assertSame](#), [assertTrue](#), [assertTrue](#), [fail](#), [fail](#)

### Methods inherited from class java.lang.[Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),  
[wait](#), [wait](#), [wait](#)

## Constructor Detail

### TestSetup

```
public TestSetup(Test test)
```

## Method Detail

### run

```
public void run(TestResult result)
```

Description copied from interface: [Test](#)

Runs a test and collects its result in a [TestResult](#) instance.

**Specified by:**

[run](#) in interface [Test](#)

**Overrides:**

[run](#) in class [TestDecorator](#)

---

## setUp

```
protected void setUp()  
    throws java.lang.Exception
```

Sets up the fixture. Override to set up additional fixture state.

`java.lang.Exception`

---

## tearDown

```
protected void tearDown()  
    throws java.lang.Exception
```

Tears down the fixture. Override to tear down the additional fixture state.

`java.lang.Exception`

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



**Overview**

**Class Tree Deprecated Index Help**

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

---

# Package junit.framework

## Interface Summary

<a href="#">Protectable</a>	A <i>Protectable</i> can be run and can throw a Throwable.
<a href="#">Test</a>	A <i>Test</i> can be run and collect its results.
<a href="#">TestListener</a>	A Listener for test progress

## Class Summary

<a href="#">Assert</a>	A set of assert methods.
<a href="#">TestCase</a>	A test case defines the fixture to run multiple tests.
<a href="#">TestFailure</a>	A <i>TestFailure</i> collects a failed test together with the caught exception.
<a href="#">TestResult</a>	A <i>TestResult</i> collects the results of executing a test case.
<a href="#">TestSuite</a>	A <i>TestSuite</i> is a Composite of Tests.

## Error Summary

<a href="#">AssertionFailedError</a>	Thrown when an assertion failed.
<a href="#">ComparisonFailure</a>	Thrown when an assert equals for Strings failed.

---

[Overview](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

---



# Hierarchy For Package junit.framework

Package Hierarchies:

[All Packages](#)

---

# Class Hierarchy

- class java.lang.Object
  - class junit.framework.[Assert](#)
    - class junit.framework.[TestCase](#) (implements junit.framework.[Test](#))
  - class junit.framework.[TestFailure](#)
  - class junit.framework.[TestResult](#)
  - class junit.framework.[TestSuite](#) (implements junit.framework.[Test](#))
  - class java.lang.Throwable (implements java.io.Serializable)
    - class java.lang.Error
      - class junit.framework.[AssertionFailedError](#)
      - class junit.framework.[ComparisonFailure](#)

# Interface Hierarchy

- interface junit.framework.[Protectable](#)
- interface junit.framework.[Test](#)
- interface junit.framework.[TestListener](#)

---

---

<a href="#">Overview</a>	<a href="#">Package</a>	Class	<input type="checkbox"/>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>
--------------------------	-------------------------	-------	--------------------------	----------------------------	-----------------------	----------------------

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

---

## [Overview](#) [Package](#)

## [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Interface Protectable**

---

public interface **Protectable**

A *Protectable* can be run and can throw a Throwable.

**See Also:**

[TestResult](#)

---

### Method Summary

void	<a href="#">protect</a> () Run the the following method protected.
------	---

### Method Detail

#### **protect**

```
public void protect()  
    throws java.lang.Throwable
```

Run the the following method protected.

```
java.lang.Throwable
```

---

[Overview](#) [Package](#)

[Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Interface Test**

All Known Implementing Classes:

[TestCase](#), [TestDecorator](#), [TestSuite](#)

---

public interface **Test**

A *Test* can be run and collect its results.

See Also:

[TestResult](#)

---

### Method Summary

int	<a href="#">countTestCases()</a> Counts the number of test cases that will be run by this test.
void	<a href="#">run(<a href="#">TestResult</a> result)</a> Runs a test and collects its result in a <a href="#">TestResult</a> instance.

### Method Detail

#### **countTestCases**

```
public int countTestCases()
```

Counts the number of test cases that will be run by this test.

---

#### **run**

```
public void run(TestResult result)
```

Runs a test and collects its result in a [TestResult](#) instance.

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Interface TestListener**

---

public interface **TestListener**

A Listener for test progress

---

### **Method Summary**

void	<a href="#">addError</a> ( <a href="#">Test</a> test, java.lang.Throwable t) An error occurred.
void	<a href="#">addFailure</a> ( <a href="#">Test</a> test, <a href="#">AssertionFailedError</a> t) A failure occurred.
void	<a href="#">endTest</a> ( <a href="#">Test</a> test) A test ended.
void	<a href="#">startTest</a> ( <a href="#">Test</a> test) A test started.

### **Method Detail**

#### **addError**

```
public void addError(Test test,  
                    java.lang.Throwable t)
```

An error occurred.

---

#### **addFailure**

```
public void addFailure(Test test,  
                      AssertionFailedError t)
```

A failure occurred.

---

## endTest

```
public void endTest(Test test)
```

A test ended.

---

## startTest

```
public void startTest(Test test)
```

A test started.

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Class Assert**

```
java.lang.Object
|
+-- junit.framework.Assert
```

### Direct Known Subclasses:

[TestCase](#), [TestDecorator](#)

---

```
public class Assert
extends java.lang.Object
```

A set of assert methods. Messages are only displayed when an assert fails.

---

## Constructor Summary

protected	<a href="#">Assert</a> () Protect constructor since it is a static only class
-----------	--

## Method Summary

static void	<a href="#">assertEquals</a> (boolean expected, boolean actual) Asserts that two booleans are equal.
static void	<a href="#">assertEquals</a> (byte expected, byte actual) Asserts that two bytes are equal.
static void	<a href="#">assertEquals</a> (char expected, char actual) Asserts that two chars are equal.
static void	<a href="#">assertEquals</a> (double expected, double actual, double delta) Asserts that two doubles are equal concerning a delta.
static void	<a href="#">assertEquals</a> (float expected, float actual, float delta) Asserts that two floats are equal concerning a delta.
static void	<a href="#">assertEquals</a> (int expected, int actual) Asserts that two ints are equal.



static void	<a href="#">assertEquals</a> (long expected, long actual) Asserts that two longs are equal.
static void	<a href="#">assertEquals</a> (java.lang.Object expected, java.lang.Object actual) Asserts that two objects are equal.
static void	<a href="#">assertEquals</a> (short expected, short actual) Asserts that two shorts are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, boolean expected, boolean actual) Asserts that two booleans are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, byte expected, byte actual) Asserts that two bytes are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, char expected, char actual) Asserts that two chars are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, double expected, double actual, double delta) Asserts that two doubles are equal concerning a delta.
static void	<a href="#">assertEquals</a> (java.lang.String message, float expected, float actual, float delta) Asserts that two floats are equal concerning a delta.
static void	<a href="#">assertEquals</a> (java.lang.String message, int expected, int actual) Asserts that two ints are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, long expected, long actual) Asserts that two longs are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, java.lang.Object expected, java.lang.Object actual) Asserts that two objects are equal.
static void	<a href="#">assertEquals</a> (java.lang.String message, short expected, short actual) Asserts that two shorts are equal.
static void	<a href="#">assertEquals</a> (java.lang.String expected, java.lang.String actual) Asserts that two Strings are equal.
	<a href="#">assertEquals</a> (java.lang.String message,

static void	java.lang.String expected, java.lang.String actual) Asserts that two Strings are equal.
static void	<a href="#">assertFalse</a> (boolean condition) Asserts that a condition is false.
static void	<a href="#">assertFalse</a> (java.lang.String message, boolean condition) Asserts that a condition is false.
static void	<a href="#">assertNotNull</a> (java.lang.Object object) Asserts that an object isn't null.
static void	<a href="#">assertNotNull</a> (java.lang.String message, java.lang.Object object) Asserts that an object isn't null.
static void	<a href="#">assertNotSame</a> (java.lang.Object expected, java.lang.Object actual) Asserts that two objects refer to the same object.
static void	<a href="#">assertNotSame</a> (java.lang.String message, java.lang.Object expected, java.lang.Object actual) Asserts that two objects refer to the same object.
static void	<a href="#">assertNull</a> (java.lang.Object object) Asserts that an object is null.
static void	<a href="#">assertNull</a> (java.lang.String message, java.lang.Object object) Asserts that an object is null.
static void	<a href="#">assertSame</a> (java.lang.Object expected, java.lang.Object actual) Asserts that two objects refer to the same object.
static void	<a href="#">assertSame</a> (java.lang.String message, java.lang.Object expected, java.lang.Object actual) Asserts that two objects refer to the same object.
static void	<a href="#">assertTrue</a> (boolean condition) Asserts that a condition is true.
static void	<a href="#">assertTrue</a> (java.lang.String message, boolean condition) Asserts that a condition is true.
static void	<a href="#">fail</a> () Fails a test with no message.
static void	<a href="#">fail</a> (java.lang.String message) Fails a test with the given message.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### Assert

protected **Assert**()

Protect constructor since it is a static only class

## Method Detail

### assertTrue

```
public static void assertTrue(java.lang.String message,  
                               boolean condition)
```

Asserts that a condition is true. If it isn't it throws an `AssertionFailedError` with the given message.

---

### assertTrue

```
public static void assertTrue(boolean condition)
```

Asserts that a condition is true. If it isn't it throws an `AssertionFailedError`.

---

### assertFalse

```
public static void assertFalse(java.lang.String message,  
                                boolean condition)
```



Asserts that two objects are equal. If they are not an `AssertionFailedError` is thrown.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                java.lang.String expected,  
                                java.lang.String actual)
```

Asserts that two Strings are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String expected,  
                                java.lang.String actual)
```

Asserts that two Strings are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                double expected,  
                                double actual,  
                                double delta)
```

Asserts that two doubles are equal concerning a delta. If they are not an `AssertionFailedError` is thrown with the given message. If the expected value is infinity then the delta value is ignored.

---

## **assertEquals**

```
public static void assertEquals(double expected,  
                                double actual,  
                                double delta)
```

Asserts that two doubles are equal concerning a delta. If the expected value is infinity then the delta value is ignored.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                float expected,  
                                float actual,  
                                float delta)
```

Asserts that two floats are equal concerning a delta. If they are not an `AssertionFailedError` is thrown with the given message. If the expected value is infinity then the delta value is ignored.

---

## **assertEquals**

```
public static void assertEquals(float expected,  
                                float actual,  
                                float delta)
```

Asserts that two floats are equal concerning a delta. If the expected value is infinity then the delta value is ignored.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                long expected,  
                                long actual)
```

Asserts that two longs are equal. If they are not an `AssertionFailedError` is thrown with the given message.

---

## **assertEquals**

```
public static void assertEquals(long expected,  
                                long actual)
```

Asserts that two longs are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                boolean expected,  
                                boolean actual)
```

Asserts that two booleans are equal. If they are not an `AssertionFailedError` is thrown with the given message.

---

## **assertEquals**

```
public static void assertEquals(boolean expected,  
                                boolean actual)
```

Asserts that two booleans are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                byte expected,  
                                byte actual)
```

Asserts that two bytes are equal. If they are not an `AssertionFailedError` is thrown with the given message.

---

## **assertEquals**

```
public static void assertEquals(byte expected,  
                                byte actual)
```

Asserts that two bytes are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                char expected,  
                                char actual)
```

Asserts that two chars are equal. If they are not an `AssertionFailedError` is thrown with the given message.

---

## **assertEquals**

```
public static void assertEquals(char expected,  
                                char actual)
```

Asserts that two chars are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                short expected,  
                                short actual)
```

Asserts that two shorts are equal. If they are not an `AssertionFailedError` is thrown with the given message.

---

## **assertEquals**

```
public static void assertEquals(short expected,  
                                short actual)
```

Asserts that two shorts are equal.

---

## **assertEquals**

```
public static void assertEquals(java.lang.String message,  
                                int expected,  
                                int actual)
```

Asserts that two ints are equal. If they are not an `AssertionFailedError` is thrown with the given message.

---



## **assertEquals**

```
public static void assertEquals(int expected,  
                                int actual)
```

Asserts that two ints are equal.

---

## **assertNotNull**

```
public static void assertNotNull(java.lang.Object object)
```

Asserts that an object isn't null.

---

## **assertNotNull**

```
public static void assertNotNull(java.lang.String message,  
                                java.lang.Object object)
```

Asserts that an object isn't null. If it is an `AssertionFailedError` is thrown with the given message.

---

## **assertNull**

```
public static void assertNull(java.lang.Object object)
```

Asserts that an object is null.

---

## **assertNull**

```
public static void assertNull(java.lang.String message,  
                                java.lang.Object object)
```

Asserts that an object is null. If it is not an `AssertionFailedError` is thrown with the given message.

---

## assertSame

```
public static void assertSame(java.lang.String message,  
                               java.lang.Object expected,  
                               java.lang.Object actual)
```

Asserts that two objects refer to the same object. If they are not an `AssertionFailedError` is thrown with the given message.

---

## assertSame

```
public static void assertSame(java.lang.Object expected,  
                               java.lang.Object actual)
```

Asserts that two objects refer to the same object. If they are not the same an `AssertionFailedError` is thrown.

---

## assertNotSame

```
public static void assertNotSame(java.lang.String message,  
                                   java.lang.Object expected,  
                                   java.lang.Object actual)
```

Asserts that two objects refer to the same object. If they are not an `AssertionFailedError` is thrown with the given message.

---

## assertNotSame

```
public static void assertNotSame(java.lang.Object expected,  
                                   java.lang.Object actual)
```

Asserts that two objects refer to the same object. If they are not the same an `AssertionFailedError` is thrown.

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Class TestCase**

```
java.lang.Object
|
+-- junit.framework.Assert
    |
    +-- junit.framework.TestCase
```

### All Implemented Interfaces:

[Test](#)

### Direct Known Subclasses:

[ExceptionTestCase](#)

---

```
public abstract class TestCase
extends Assert
implements Test
```

A test case defines the fixture to run multiple tests. To define a test case

- 1) implement a subclass of TestCase
- 2) define instance variables that store the state of the fixture
- 3) initialize the fixture state by overriding `setUp`
- 4) clean-up after a test by overriding `tearDown`.

Each test runs in its own fixture so there can be no side effects among test runs. Here is an example:

```
public class MathTest extends TestCase {
    protected double fValue1;
    protected double fValue2;

    protected void setUp() {
        fValue1= 2.0;
        fValue2= 3.0;
    }
}
```

For each test implement a method which interacts with the fixture. Verify the expected results with assertions specified by calling `assertTrue` with a boolean.

```
public void testAdd() {
    double result= fValue1 + fValue2;
    assertTrue(result == 5.0);
}
```

Once the methods are defined you can run them. The framework supports both a static type safe and more dynamic way to run a test. In the static way you override the `runTest` method and define the method to be invoked. A convenient way to do so is with an anonymous inner class.

```
TestCase test= new MathTest("add") {
    public void runTest() {
        testAdd();
    }
};
test.run();
```

The dynamic way uses reflection to implement `runTest`. It dynamically finds and invokes a method. In this case the name of the test case has to correspond to the test method to be run.

```
TestCase= new MathTest("testAdd");
test.run();
```

The tests to be run can be collected into a `TestSuite`. JUnit provides different *test runners* which can run a test suite and collect the results. A test runner either expects a static method `suite` as the entry point to get a test to run or it will extract the suite automatically.

```
public static Test suite() {
    suite.addTest(new MathTest("testAdd"));
    suite.addTest(new MathTest("testDivideByZero"));
    return suite;
}
```

**See Also:**

[TestResult](#), [TestSuite](#)

---

---

## Constructor Summary

---

---

[TestCase\(\)](#)

No-arg constructor to enable serialization.

[TestCase\(java.lang.String name\)](#)

Constructs a test case with the given name.

## Method Summary

int	<a href="#">countTestCases()</a> Counts the number of test cases executed by <code>run(TestResult result)</code> .
protected <a href="#">TestResult</a>	<a href="#">createResult()</a> Creates a default TestResult object
java.lang.String	<a href="#">getName()</a> Gets the name of a TestCase
<a href="#">TestResult</a>	<a href="#">run()</a> A convenience method to run this test, collecting the results with a default TestResult object.
void	<a href="#">run(TestResult result)</a> Runs the test case and collects the results in TestResult.
void	<a href="#">runBare()</a> Runs the bare test sequence.
protected void	<a href="#">runTest()</a> Override to run the test and assert its state.
void	<a href="#">setName(java.lang.String name)</a> Sets the name of a TestCase
protected void	<a href="#">setUp()</a> Sets up the fixture, for example, open a network connection.
protected void	<a href="#">tearDown()</a> Tears down the fixture, for example, close a network connection.
java.lang.String	<a href="#">toString()</a> Returns a string representation of the test case

## Methods inherited from class junit.framework.Assert

[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertEquals](#), [assertEquals](#), [assertEquals](#), [assertEquals](#),  
[assertFalse](#), [assertFalse](#), [assertNotNull](#), [assertNotNull](#),  
[assertNotSame](#), [assertNotSame](#), [assertNull](#), [assertNull](#), [assertSame](#),  
[assertSame](#), [assertTrue](#), [assertTrue](#), [fail](#), [fail](#)

## Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),  
[wait](#), [wait](#), [wait](#)

## Constructor Detail

### TestCase

```
public TestCase()
```

No-arg constructor to enable serialization. This method is not intended to be used by mere mortals without calling setName().

### TestCase

```
public TestCase(java.lang.String name)
```

Constructs a test case with the given name.

## Method Detail

### countTestCases

```
public int countTestCases()
```

Counts the number of test cases executed by run(TestResult result).

**Specified by:**

[countTestCases](#) in interface [Test](#)

---

## **createResult**

protected [TestResult](#) **createResult()**

Creates a default TestResult object

**See Also:**

[TestResult](#)

---

## **run**

public [TestResult](#) **run()**

A convenience method to run this test, collecting the results with a default TestResult object.

**See Also:**

[TestResult](#)

---

## **run**

public void **run**([TestResult](#) result)

Runs the test case and collects the results in TestResult.

**Specified by:**

[run](#) in interface [Test](#)

---

## **runBare**

public void **runBare()**  
throws java.lang.Throwable



Runs the bare test sequence.

**Throws:**

`java.lang.Throwable` - if any exception is thrown

---

## **runTest**

```
protected void runTest()  
    throws java.lang.Throwable
```

Override to run the test and assert its state.

**Throws:**

`java.lang.Throwable` - if any exception is thrown

---

## **setUp**

```
protected void setUp()  
    throws java.lang.Exception
```

Sets up the fixture, for example, open a network connection. This method is called before a test is executed.

`java.lang.Exception`

---

## **tearDown**

```
protected void tearDown()  
    throws java.lang.Exception
```

Tears down the fixture, for example, close a network connection. This method is called after a test is executed.

`java.lang.Exception`

---

## **toString**

```
public java.lang.String toString()
```

Returns a string representation of the test case

**Overrides:**

toString in class java.lang.Object

---

## getName

```
public java.lang.String getName()
```

Gets the name of a TestCase

**Returns:**

returns a String

---

## setName

```
public void setName(java.lang.String name)
```

Sets the name of a TestCase

**Parameters:**

name - The name to set

---

[Overview](#) [Package](#)

[Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# junit.framework **Class TestFailure**

java.lang.Object

|  
+-- junit.framework.TestFailure

---

```
public class TestFailure
extends java.lang.Object
```

A `TestFailure` collects a failed test together with the caught exception.

**See Also:**

[TestResult](#)

---

## Field Summary

protected <a href="#">Test</a>	<a href="#">fFailedTest</a>
protected java.lang.Throwable	<a href="#">fThrownException</a>

## Constructor Summary

<a href="#">TestFailure</a> ( <a href="#">Test</a> failedTest, java.lang.Throwable thrownException) Constructs a TestFailure with the given test and exception.
--

## Method Summary

java.lang.String	<a href="#">exceptionMessage</a> ()
<a href="#">Test</a>	<a href="#">failedTest</a> () Gets the failed test.
boolean	<a href="#">isFailure</a> ()
	<a href="#">thrownException</a> ()

java.lang.Throwable	Gets the thrown exception.
java.lang.String	<a href="#">toString()</a> Returns a short description of the failure.
java.lang.String	<a href="#">trace()</a>

<b>Methods inherited from class java.lang.Object</b>
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### fFailedTest

protected [Test](#) fFailedTest

### fThrownException

protected java.lang.Throwable fThrownException

## Constructor Detail

### TestFailure

```
public TestFailure(Test failedTest,
                  java.lang.Throwable thrownException)
```

Constructs a TestFailure with the given test and exception.

## Method Detail

### failedTest

```
public Test failedTest()
```

Gets the failed test.

---

## thrownException

```
public java.lang.Throwable thrownException()
```

Gets the thrown exception.

---

## toString

```
public java.lang.String toString()
```

Returns a short description of the failure.

### Overrides:

toString in class java.lang.Object

---

## trace

```
public java.lang.String trace()
```

---

## exceptionMessage

```
public java.lang.String exceptionMessage()
```

---

## isFailure

```
public boolean isFailure()
```

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Class TestResult**

```
java.lang.Object
|
+-- junit.framework.TestResult
```

---

```
public class TestResult
extends java.lang.Object
```

A `TestResult` collects the results of executing a test case. It is an instance of the Collecting Parameter pattern. The test framework distinguishes between *failures* and *errors*. A failure is anticipated and checked for with assertions. Errors are unanticipated problems like an `ArrayIndexOutOfBoundsException`.

### See Also:

[Test](#)

---

## Field Summary

protected java.util.Vector	<a href="#">fErrors</a>
protected java.util.Vector	<a href="#">fFailures</a>
protected java.util.Vector	<a href="#">fListeners</a>
protected int	<a href="#">fRunTests</a>

## Constructor Summary

[TestResult\(\)](#)

---



# Method Summary

void	<a href="#">addError</a> ( <a href="#">Test</a> test, java.lang.Throwable t) Adds an error to the list of errors.
void	<a href="#">addFailure</a> ( <a href="#">Test</a> test, <a href="#">AssertionFailedError</a> t) Adds a failure to the list of failures.
void	<a href="#">addListener</a> ( <a href="#">TestListener</a> listener) Registers a TestListener
void	<a href="#">endTest</a> ( <a href="#">Test</a> test) Informs the result that a test was completed.
int	<a href="#">errorCount</a> () Gets the number of detected errors.
java.util.Enumeration	<a href="#">errors</a> () Returns an Enumeration for the errors
int	<a href="#">failureCount</a> () Gets the number of detected failures.
java.util.Enumeration	<a href="#">failures</a> () Returns an Enumeration for the failures
void	<a href="#">removeListener</a> ( <a href="#">TestListener</a> listener) Unregisters a TestListener
protected void	<a href="#">run</a> ( <a href="#">TestCase</a> test) Runs a TestCase.
int	<a href="#">runCount</a> () Gets the number of run tests.
void	<a href="#">runProtected</a> ( <a href="#">Test</a> test, <a href="#">Protectable</a> p) Runs a TestCase.
boolean	<a href="#">shouldStop</a> () Checks whether the test run should stop
void	<a href="#">startTest</a> ( <a href="#">Test</a> test) Informs the result that a test will be started.
void	<a href="#">stop</a> () Marks that the test run should stop.
boolean	<a href="#">wasSuccessful</a> () Returns whether the entire test was successful or not.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### fFailures

protected java.util.Vector **fFailures**

---

### fErrors

protected java.util.Vector **fErrors**

---

### fListeners

protected java.util.Vector **fListeners**

---

### fRunTests

protected int **fRunTests**

## Constructor Detail

### TestResult

public TestResult()

## Method Detail

### addError

```
public void addError(Test test,  
                    java.lang.Throwable t)
```

Adds an error to the list of errors. The passed in exception caused the error.

---

## **addFailure**

```
public void addFailure(Test test,  
                      AssertionFailedError t)
```

Adds a failure to the list of failures. The passed in exception caused the failure.

---

## **addListener**

```
public void addListener(TestListener listener)
```

Registers a TestListener

---

## **removeListener**

```
public void removeListener(TestListener listener)
```

Unregisters a TestListener

---

## **endTest**

```
public void endTest(Test test)
```

Informs the result that a test was completed.

---

## **errorCount**

```
public int errorCount()
```

Gets the number of detected errors.

---

## **errors**

```
public java.util.Enumeration errors()
```

Returns an Enumeration for the errors

---

## **failureCount**

```
public int failureCount()
```

Gets the number of detected failures.

---

## **failures**

```
public java.util.Enumeration failures()
```

Returns an Enumeration for the failures

---

## **run**

```
protected void run(TestCase test)
```

Runs a TestCase.

---

## **runCount**

```
public int runCount()
```

Gets the number of run tests.

---

## **runProtected**

```
public void runProtected(Test test,  
                        Protectable p)
```

Runs a TestCase.

---

## shouldStop

```
public boolean shouldStop()
```

Checks whether the test run should stop

---

## startTest

```
public void startTest(Test test)
```

Informs the result that a test will be started.

---

## stop

```
public void stop()
```

Marks that the test run should stop.

---

## wasSuccessful

```
public boolean wasSuccessful()
```

Returns whether the entire test was successful or not.

---

[Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Class TestSuite**

```
java.lang.Object
|
+-- junit.framework.TestSuite
```

### All Implemented Interfaces:

[Test](#)

### Direct Known Subclasses:

[ActiveTestSuite](#)

---

```
public class TestSuite
extends java.lang.Object
implements Test
```

A `TestSuite` is a composite of `Tests`. It runs a collection of test cases. Here is an example using the dynamic test definition.

```
TestSuite suite= new TestSuite();
suite.addTest(new MathTest("testAdd"));
suite.addTest(new MathTest("testDivideByZero"));
```

Alternatively, a `TestSuite` can extract the tests to be run automatically. To do so you pass the class of your `TestCase` class to the `TestSuite` constructor.

```
TestSuite suite= new TestSuite(MathTest.class);
```

This constructor creates a suite with all the methods starting with "test" that take no arguments.

### See Also:

[Test](#)

---

## Constructor Summary

<a href="#">TestSuite()</a>	Constructs an empty TestSuite.
<a href="#">TestSuite(java.lang.Class theClass)</a>	Constructs a TestSuite from the given class.
<a href="#">TestSuite(java.lang.Class theClass, java.lang.String name)</a>	Constructs a TestSuite from the given class with the given name.
<a href="#">TestSuite(java.lang.String name)</a>	Constructs an empty TestSuite.

## Method Summary

void	<a href="#">addTest</a> ( <a href="#">Test</a> test) Adds a test to the suite.
void	<a href="#">addTestSuite</a> (java.lang.Class testClass) Adds the tests from the given class to suite
int	<a href="#">countTestCases</a> () Counts the number of test cases that be run by this test.
static <a href="#">Test</a>	<a href="#">createTest</a> (java.lang.Class theClass, java.lang.String name) ...as the moon sets over the early morning Merlin, Oregon mountains, our intr adventurers type...
java.lang.String	<a href="#">getName</a> () Returns the name of the suite.
static java.lang.reflect.Constructor	<a href="#">getTestConstructor</a> (java.lang.Class theClass) Gets a constructor which takes a single String as its argument or a no arg constructor
void	<a href="#">run</a> ( <a href="#">TestResult</a> result) Runs the tests and collects their results in a TestResult.
void	<a href="#">runTest</a> ( <a href="#">Test</a> test, <a href="#">TestResult</a> result)
void	<a href="#">setName</a> (java.lang.String name) Sets the name of the suite.



	<a href="#">Test</a>	<a href="#">testAt</a> (int index) Returns the test at the given index
	int	<a href="#">testCount</a> () Returns the number of tests in this suite
	java.util.Enumeration	<a href="#">tests</a> () Returns the tests as an enumeration
	java.lang.String	<a href="#">toString</a> ()

<b>Methods inherited from class java.lang.Object</b>
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### TestSuite

```
public TestSuite()
```

Constructs an empty TestSuite.

### TestSuite

```
public TestSuite(java.lang.Class theClass,  
                 java.lang.String name)
```

Constructs a TestSuite from the given class with the given name.

### See Also:

[TestSuite\(Class\)](#)

### TestSuite

```
public TestSuite(java.lang.Class theClass)
```

Constructs a TestSuite from the given class. Adds all the methods starting with "test" as test cases to the suite. Parts of this method was written at 2337 meters in the Hüffihütte, Kanton Uri

---

## TestSuite

```
public TestSuite(java.lang.String name)
```

Constructs an empty TestSuite.

## Method Detail

### addTest

```
public void addTest(Test test)
```

Adds a test to the suite.

---

### addTestSuite

```
public void addTestSuite(java.lang.Class testClass)
```

Adds the tests from the given class to the suite

---

### createTest

```
public static Test createTest(java.lang.Class theClass,  
                               java.lang.String name)
```

...as the moon sets over the early morning Merlin, Oregon mountains, our intrepid adventurers type...

---

### countTestCases

```
public int countTestCases()
```

Counts the number of test cases that will be run by this test.

**Specified by:**

[countTestCases](#) in interface [Test](#)

---

## getTestConstructor

```
public static java.lang.reflect.Constructor getTestConstructor(java.  
                                                                    throws java.
```

Gets a constructor which takes a single String as its argument or a no arg constructor.

```
java.lang.NoSuchMethodException
```

---

## run

```
public void run(TestResult result)
```

Runs the tests and collects their result in a TestResult.

**Specified by:**

[run](#) in interface [Test](#)

---

## runTest

```
public void runTest(Test test,  
                    TestResult result)
```

---

## testAt

```
public Test testAt(int index)
```

Returns the test at the given index

---

## testCount

```
public int testCount()
```

Returns the number of tests in this suite

---

## tests

```
public java.util.Enumeration tests()
```

Returns the tests as an enumeration

---

## toString

```
public java.lang.String toString()
```

### Overrides:

toString in class java.lang.Object

---

## setName

```
public void setName(java.lang.String name)
```

Sets the name of the suite.

### Parameters:

name - The name to set


---

## getName

```
public java.lang.String getName()
```

Returns the name of the suite. Not all test suites have a name and this method can return null.

---



## [Overview](#) **Package**

## [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Class AssertionFailedError**

```
java.lang.Object
|
+-- java.lang.Throwable
    |
    +-- java.lang.Error
        |
        +-- junit.framework.AssertionFailedError
```

### All Implemented Interfaces:

java.io.Serializable

### Direct Known Subclasses:

[ComparisonFailure](#)

---

```
public class AssertionFailedError
extends java.lang.Error
```

Thrown when an assertion failed.

### See Also:

[Serialized Form](#)

---

## Constructor Summary

[AssertionFailedError](#)()

[AssertionFailedError](#)(java.lang.String message)

## Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

---

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### AssertionFailedError

```
public AssertionFailedError()
```

### AssertionFailedError

```
public AssertionFailedError(java.lang.String message)
```

[Overview](#) [Package](#)

[Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | METHOD



# [Overview](#) [Package](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## junit.framework **Class ComparisonFailure**

```
java.lang.Object
|
+-- java.lang.Throwable
    |
    +-- java.lang.Error
        |
        +-- junit.framework.AssertionFailedError
            |
            +-- junit.framework.ComparisonFailure
```

### All Implemented Interfaces:

[java.io.Serializable](#)

---

public class **ComparisonFailure**  
extends [AssertionFailedError](#)

Thrown when an assert equals for Strings failed. Inspired by a patch from Alex Chaffee <mailto:alex@purpletech.com>

### See Also:

[Serialized Form](#)

---

## Constructor Summary

<a href="#">ComparisonFailure</a> (java.lang.String message, java.lang.String expected, java.lang.String actual)
Constructs a comparison failure.

## Method Summary

java.lang.String	<a href="#">getMessage</a> ()
	Returns "... " in place of common prefix and "... " in place of common suffix between expected and actual.

## Methods inherited from class java.lang.Throwable

---

```
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace,
initCause, printStackTrace, printStackTrace, printStackTrace,
setStackTrace, toString
```

## Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait
```

# Constructor Detail

## ComparisonFailure

```
public ComparisonFailure(java.lang.String message,
                          java.lang.String expected,
                          java.lang.String actual)
```

Constructs a comparison failure.

### Parameters:

message - the identifying message or null  
expected - the expected string value  
actual - the actual string value

# Method Detail

## getMessage

```
public java.lang.String getMessage()
```

Returns "... " in place of common prefix and "... " in place of common suffix between expected and actual.

### Overrides:

getMessage in class java.lang.Throwable

### See Also:

Throwable.getMessage()

---

## [Overview](#) **Package**

## [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



# Deprecated API

---

---

[Overview](#) [Package](#) [Class](#) [Tree](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

---



## A

[\*\*ActiveTestSuite\*\*](#) - class junit.extensions.[ActiveTestSuite](#).

A TestSuite for active Tests.

[\*\*ActiveTestSuite\(\)\*\*](#) - Constructor for class junit.extensions.[ActiveTestSuite](#)

[\*\*ActiveTestSuite\(Class\)\*\*](#) - Constructor for class junit.extensions.[ActiveTestSuite](#)

[\*\*ActiveTestSuite\(Class, String\)\*\*](#) - Constructor for class junit.extensions.[ActiveTestSuite](#)

[\*\*ActiveTestSuite\(String\)\*\*](#) - Constructor for class junit.extensions.[ActiveTestSuite](#)

[\*\*addError\(Test, Throwable\)\*\*](#) - Method in class junit.framework.[TestResult](#)

Adds an error to the list of errors.

[\*\*addError\(Test, Throwable\)\*\*](#) - Method in interface junit.framework.[TestListener](#)

An error occurred.

[\*\*addFailure\(Test, AssertionError\)\*\*](#) - Method in class junit.framework.[TestResult](#)

Adds a failure to the list of failures.

[\*\*addFailure\(Test, AssertionError\)\*\*](#) - Method in interface junit.framework.[TestListener](#)

A failure occurred.

[\*\*addListener\(TestListener\)\*\*](#) - Method in class junit.framework.[TestResult](#)

Registers a TestListener

[\*\*addTest\(Test\)\*\*](#) - Method in class junit.framework.[TestSuite](#)

Adds a test to the suite.

[\*\*addTestSuite\(Class\)\*\*](#) - Method in class junit.framework.[TestSuite](#)

Adds the tests from the given class to the suite

[\*\*Assert\*\*](#) - class junit.framework.[Assert](#).

A set of assert methods.

[\*\*Assert\(\)\*\*](#) - Constructor for class junit.framework.[Assert](#)

Protect constructor since it is a static only class

[\*\*assertEquals\(boolean, boolean\)\*\*](#) - Static method in class



junit.framework.[Assert](#)

Asserts that two booleans are equal.

[assertEquals\(byte, byte\)](#) - Static method in class junit.framework.[Assert](#)

Asserts that two bytes are equal.

[assertEquals\(char, char\)](#) - Static method in class junit.framework.[Assert](#)

Asserts that two chars are equal.

[assertEquals\(double, double, double\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two doubles are equal concerning a delta.

[assertEquals\(float, float, float\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two floats are equal concerning a delta.

[assertEquals\(int, int\)](#) - Static method in class junit.framework.[Assert](#)

Asserts that two ints are equal.

[assertEquals\(long, long\)](#) - Static method in class junit.framework.[Assert](#)

Asserts that two longs are equal.

[assertEquals\(Object, Object\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two objects are equal.

[assertEquals\(short, short\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two shorts are equal.

[assertEquals\(String, boolean, boolean\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two booleans are equal.

[assertEquals\(String, byte, byte\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two bytes are equal.

[assertEquals\(String, char, char\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two chars are equal.

[assertEquals\(String, double, double, double\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two doubles are equal concerning a delta.

[assertEquals\(String, float, float, float\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two floats are equal concerning a delta.

[assertEquals\(String, int, int\)](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two ints are equal.

[\*\*assertEquals\(String, long, long\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two longs are equal.

[\*\*assertEquals\(String, Object, Object\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two objects are equal.

[\*\*assertEquals\(String, short, short\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two shorts are equal.

[\*\*assertEquals\(String, String\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two Strings are equal.

[\*\*assertEquals\(String, String, String\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two Strings are equal.

[\*\*assertFalse\(boolean\)\*\*](#) - Static method in class junit.framework.[Assert](#)

Asserts that a condition is false.

[\*\*assertFalse\(String, boolean\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that a condition is false.

[\*\*AssertionFailedError\*\*](#) - error junit.framework.[AssertionFailedError](#).

Thrown when an assertion failed.

[\*\*AssertionFailedError\(\)\*\*](#) - Constructor for class  
junit.framework.[AssertionFailedError](#)

[\*\*AssertionFailedError\(String\)\*\*](#) - Constructor for class  
junit.framework.[AssertionFailedError](#)

[\*\*assertNotNull\(Object\)\*\*](#) - Static method in class junit.framework.[Assert](#)

Asserts that an object isn't null.

[\*\*assertNotNull\(String, Object\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that an object isn't null.

[\*\*assertNotSame\(Object, Object\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two objects refer to the same object.

[\*\*assertNotSame\(String, Object, Object\)\*\*](#) - Static method in class  
junit.framework.[Assert](#)

Asserts that two objects refer to the same object.

[\*\*assertNull\(Object\)\*\*](#) - Static method in class [junit.framework.Assert](#)

Asserts that an object is null.

[\*\*assertNull\(String, Object\)\*\*](#) - Static method in class [junit.framework.Assert](#)

Asserts that an object is null.

[\*\*assertSame\(Object, Object\)\*\*](#) - Static method in class [junit.framework.Assert](#)

Asserts that two objects refer to the same object.

[\*\*assertSame\(String, Object, Object\)\*\*](#) - Static method in class [junit.framework.Assert](#)

Asserts that two objects refer to the same object.

[\*\*assertTrue\(boolean\)\*\*](#) - Static method in class [junit.framework.Assert](#)

Asserts that a condition is true.

[\*\*assertTrue\(String, boolean\)\*\*](#) - Static method in class [junit.framework.Assert](#)

Asserts that a condition is true.

---

## B

[basicRun\(TestResult\)](#) - Method in class junit.extensions.[TestDecorator](#)  
The basic run behaviour.

---

## C

**[ComparisonFailure](#)** - error junit.framework.[ComparisonFailure](#).

Thrown when an assert equals for Strings failed.

**[ComparisonFailure\(String, String, String\)](#)** - Constructor for class junit.framework.[ComparisonFailure](#)

Constructs a comparison failure.

**[countTestCases\(\)](#)** - Method in class junit.extensions.[TestDecorator](#)

**[countTestCases\(\)](#)** - Method in class junit.extensions.[RepeatedTest](#)

**[countTestCases\(\)](#)** - Method in class junit.framework.[TestSuite](#)

Counts the number of test cases that will be run by this test.

**[countTestCases\(\)](#)** - Method in class junit.framework.[TestCase](#)

Counts the number of test cases executed by run([TestResult](#) result).

**[countTestCases\(\)](#)** - Method in interface junit.framework.[Test](#)

Counts the number of test cases that will be run by this test.

**[createResult\(\)](#)** - Method in class junit.framework.[TestCase](#)

Creates a default [TestResult](#) object

**[createTest\(Class, String\)](#)** - Static method in class junit.framework.[TestSuite](#)

...as the moon sets over the early morning Merlin, Oregon mountains, our intrepid adventurers type...

---

## E

**[endTest\(Test\)](#)** - Method in class junit.framework.[TestResult](#)

Informs the result that a test was completed.

**[endTest\(Test\)](#)** - Method in interface junit.framework.[TestListener](#)

A test ended.

**[errorCount\(\)](#)** - Method in class junit.framework.[TestResult](#)

Gets the number of detected errors.

**[errors\(\)](#)** - Method in class junit.framework.[TestResult](#)

Returns an Enumeration for the errors

**[exceptionMessage\(\)](#)** - Method in class junit.framework.[TestFailure](#)

**[ExceptionTestCase](#)** - class junit.extensions.[ExceptionTestCase](#).

A TestCase that expects an Exception of class fExpected to be thrown.

**[ExceptionTestCase\(String, Class\)](#)** - Constructor for class junit.extensions.[ExceptionTestCase](#)

---

## F

**[fail\(\)](#)** - Static method in class [junit.framework.Assert](#)

    Fails a test with no message.

**[fail\(String\)](#)** - Static method in class [junit.framework.Assert](#)

    Fails a test with the given message.

**[failedTest\(\)](#)** - Method in class [junit.framework.TestFailure](#)

    Gets the failed test.

**[failureCount\(\)](#)** - Method in class [junit.framework.TestResult](#)

    Gets the number of detected failures.

**[failures\(\)](#)** - Method in class [junit.framework.TestResult](#)

    Returns an Enumeration for the failures

**[fErrors](#)** - Variable in class [junit.framework.TestResult](#)

**[fFailedTest](#)** - Variable in class [junit.framework.TestFailure](#)

**[fFailures](#)** - Variable in class [junit.framework.TestResult](#)

**[fListeners](#)** - Variable in class [junit.framework.TestResult](#)

**[fRunTests](#)** - Variable in class [junit.framework.TestResult](#)

**[fTest](#)** - Variable in class [junit.extensions.TestDecorator](#)

**[fThrownException](#)** - Variable in class [junit.framework.TestFailure](#)

---

## G

**[getMessage\(\)](#)** - Method in class junit.framework.[ComparisonFailure](#)

Returns "... " in place of common prefix and "... " in place of common suffix between expected and actual.

**[getName\(\)](#)** - Method in class junit.framework.[TestSuite](#)

Returns the name of the suite.

**[getName\(\)](#)** - Method in class junit.framework.[TestCase](#)

Gets the name of a TestCase

**[getTest\(\)](#)** - Method in class junit.extensions.[TestDecorator](#)

**[getTestConstructor\(Class\)](#)** - Static method in class junit.framework.[TestSuite](#)

Gets a constructor which takes a single String as its argument or a no arg constructor.

---



|

[isFailure\(\)](#) - Method in class `junit.framework.TestFailure`

---

## J

[junit.extensions](#) - package junit.extensions

[junit.framework](#) - package junit.framework

---

## P

[protect\(\)](#) - Method in interface `junit.framework.Protectable`

Run the the following method protected.

[Protectable](#) - interface `junit.framework.Protectable`.

*A `Protectable` can be run and can throw a `Throwable`.*

---

## R

[removeListener\(TestListener\)](#) - Method in class junit.framework.[TestResult](#)

Unregisters a TestListener

[RepeatedTest](#) - class junit.extensions.[RepeatedTest](#).

A Decorator that runs a test repeatedly.

[RepeatedTest\(Test, int\)](#) - Constructor for class junit.extensions.[RepeatedTest](#)

[run\(\)](#) - Method in class junit.framework.[TestCase](#)

A convenience method to run this test, collecting the results with a default TestResult object.

[run\(TestCase\)](#) - Method in class junit.framework.[TestResult](#)

Runs a TestCase.

[run\(TestResult\)](#) - Method in class junit.extensions.[TestSetup](#)

[run\(TestResult\)](#) - Method in class junit.extensions.[TestDecorator](#)

[run\(TestResult\)](#) - Method in class junit.extensions.[RepeatedTest](#)

[run\(TestResult\)](#) - Method in class junit.extensions.[ActiveTestSuite](#)

[run\(TestResult\)](#) - Method in class junit.framework.[TestSuite](#)

Runs the tests and collects their result in a TestResult.

[run\(TestResult\)](#) - Method in class junit.framework.[TestCase](#)

Runs the test case and collects the results in TestResult.

[run\(TestResult\)](#) - Method in interface junit.framework.[Test](#)

Runs a test and collects its result in a TestResult instance.

[runBare\(\)](#) - Method in class junit.framework.[TestCase](#)

Runs the bare test sequence.

[runCount\(\)](#) - Method in class junit.framework.[TestResult](#)

Gets the number of run tests.

[runFinished\(Test\)](#) - Method in class junit.extensions.[ActiveTestSuite](#)

[runProtected\(Test, Protectable\)](#) - Method in class junit.framework.[TestResult](#)

Runs a TestCase.

**[runTest\(\)](#)** - Method in class junit.extensions.[ExceptionTestCase](#)

Execute the test method expecting that an Exception of class  
fExpected or one of its subclasses will be thrown

**[runTest\(\)](#)** - Method in class junit.framework.[TestCase](#)

Override to run the test and assert its state.

**[runTest\(Test, TestResult\)](#)** - Method in class  
junit.extensions.[ActiveTestSuite](#)

**[runTest\(Test, TestResult\)](#)** - Method in class junit.framework.[TestSuite](#)

---

## S

**[setName\(String\)](#)** - Method in class junit.framework.[TestSuite](#)

Sets the name of the suite.

**[setName\(String\)](#)** - Method in class junit.framework.[TestCase](#)

Sets the name of a TestCase

**[setUp\(\)](#)** - Method in class junit.extensions.[TestSetup](#)

Sets up the fixture.

**[setUp\(\)](#)** - Method in class junit.framework.[TestCase](#)

Sets up the fixture, for example, open a network connection.

**[shouldStop\(\)](#)** - Method in class junit.framework.[TestResult](#)

Checks whether the test run should stop

**[startTest\(Test\)](#)** - Method in class junit.framework.[TestResult](#)

Informs the result that a test will be started.

**[startTest\(Test\)](#)** - Method in interface junit.framework.[TestListener](#)

A test started.

**[stop\(\)](#)** - Method in class junit.framework.[TestResult](#)

Marks that the test run should stop.

---

## T

**[tearDown\(\)](#)** - Method in class `junit.extensions`. [TestSetup](#)

Tears down the fixture.

**[tearDown\(\)](#)** - Method in class `junit.framework`. [TestCase](#)

Tears down the fixture, for example, close a network connection.

**[Test](#)** - interface `junit.framework`. [Test](#).

A *Test* can be run and collect its results.

**[testAt\(int\)](#)** - Method in class `junit.framework`. [TestSuite](#)

Returns the test at the given index

**[TestCase](#)** - class `junit.framework`. [TestCase](#).

A test case defines the fixture to run multiple tests.

**[TestCase\(\)](#)** - Constructor for class `junit.framework`. [TestCase](#)

No-arg constructor to enable serialization.

**[TestCase\(String\)](#)** - Constructor for class `junit.framework`. [TestCase](#)

Constructs a test case with the given name.

**[testCount\(\)](#)** - Method in class `junit.framework`. [TestSuite](#)

Returns the number of tests in this suite

**[TestDecorator](#)** - class `junit.extensions`. [TestDecorator](#).

A Decorator for Tests.

**[TestDecorator\(Test\)](#)** - Constructor for class

`junit.extensions`. [TestDecorator](#)

**[TestFailure](#)** - class `junit.framework`. [TestFailure](#).

A `TestFailure` collects a failed test together with the caught exception.

**[TestFailure\(Test, Throwable\)](#)** - Constructor for class

`junit.framework`. [TestFailure](#)

Constructs a `TestFailure` with the given test and exception.

**[TestListener](#)** - interface `junit.framework`. [TestListener](#).

A Listener for test progress

**[TestResult](#)** - class `junit.framework`. [TestResult](#).

A `TestResult` collects the results of executing a test case.

**[TestResult\(\)](#)** - Constructor for class `junit.framework`. [TestResult](#)

**[tests\(\)](#)** - Method in class `junit.framework`. [TestSuite](#)

Returns the tests as an enumeration

**[TestSetup](#)** - class junit.extensions.[TestSetup](#).

A Decorator to set up and tear down additional fixture state.

**[TestSetup\(Test\)](#)** - Constructor for class junit.extensions.[TestSetup](#)

**[TestSuite](#)** - class junit.framework.[TestSuite](#).

A TestSuite is a composite of Tests.

**[TestSuite\(\)](#)** - Constructor for class junit.framework.[TestSuite](#)

Constructs an empty TestSuite.

**[TestSuite\(Class\)](#)** - Constructor for class junit.framework.[TestSuite](#)

Constructs a TestSuite from the given class.

**[TestSuite\(Class, String\)](#)** - Constructor for class  
junit.framework.[TestSuite](#)

Constructs a TestSuite from the given class with the given name.

**[TestSuite\(String\)](#)** - Constructor for class junit.framework.[TestSuite](#)

Constructs an empty TestSuite.

**[thrownException\(\)](#)** - Method in class junit.framework.[TestFailure](#)

Gets the thrown exception.

**[toString\(\)](#)** - Method in class junit.extensions.[TestDecorator](#)

**[toString\(\)](#)** - Method in class junit.extensions.[RepeatedTest](#)

**[toString\(\)](#)** - Method in class junit.framework.[TestSuite](#)

**[toString\(\)](#)** - Method in class junit.framework.[TestFailure](#)

Returns a short description of the failure.

**[toString\(\)](#)** - Method in class junit.framework.[TestCase](#)

Returns a string representation of the test case

**[trace\(\)](#)** - Method in class junit.framework.[TestFailure](#)

---



# W

[wasSuccessful\(\)](#) - Method in class junit.framework.[TestResult](#)

Returns whether the entire test was successful or not.

---

[A](#) [B](#) [C](#) [E](#) [F](#) [G](#) [I](#) [J](#) [P](#) [R](#) [S](#) [T](#) [W](#)

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#)  [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

# How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

## Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

## Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (*italic*)
- Classes
- Exceptions
- Errors

## Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description

- Nested Class Summary
  - Field Summary
  - Constructor Summary
  - Method Summary
- 
- Field Detail
  - Constructor Detail
  - Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

## Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

## Deprecated API

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

## Index

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

## Prev/Next

These links take you to the next or previous class, interface, package, or related page.

## Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

## Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

*This help file applies to API documentation generated using the standard doclet.*

---

---

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

---

## **All Classes** [ActiveTestSuite](#)

[Assert](#)

[AssertionFailedError](#)

[ComparisonFailure](#)

[ExceptionTestCase](#)

[\*Protectable\*](#)

[RepeatedTest](#)

[\*Test\*](#)

[TestCase](#)

[TestDecorator](#)

[TestFailure](#)

[\*TestListener\*](#)

[TestResult](#)

[TestSetup](#)

[TestSuite](#)

**[Overview](#)** [Package](#) [Class](#) **[Tree](#)** [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

# Serialized Form

---

---

**Package junit.framework**

---

---

**Class**

[junit.framework.AssertionFailedError](#)  
implements **Serializable**

---

---

**Class**

[junit.framework.ComparisonFailure](#)  
implements **Serializable**

---

---

**Serialized Fields**

---

---

**fExpected**

java.lang.String **fExpected**

---

---

**fActual**

java.lang.String **fActual**

---

---

**[Overview](#)** [Package](#) [Class](#) **[Tree](#)** [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

---