

org.junit

# Class Assert

java.lang.Object  
└─ **org.junit.Assert**

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

A set of assertion methods useful for writing tests. Only failed assertions are recorded. These methods can be used directly: `Assert.assertEquals(...)`, however, they read better if they are referenced through static import:

```
import static org.junit.Assert.*;  
...  
assertEquals(...);
```

**See Also:**  
`AssertionError`

## Constructor Summary

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

## Method Summary

static void	<a href="#"><b>assertArrayEquals</b></a> (byte[] expecteds, byte[] actuals) Asserts that two byte arrays are equal.
static void	<a href="#"><b>assertArrayEquals</b></a> (char[] expecteds, char[] actuals) Asserts that two char arrays are equal.
static void	<a href="#"><b>assertArrayEquals</b></a> (int[] expecteds, int[] actuals) Asserts that two int arrays are equal.
static void	<a href="#"><b>assertArrayEquals</b></a> (long[] expecteds, long[] actuals) Asserts that two long arrays are equal.
static void	<a href="#"><b>assertArrayEquals</b></a> (java.lang.Object[] expecteds, java.lang.Object[] actuals) Asserts that two object arrays are equal.
static void	<a href="#"><b>assertArrayEquals</b></a> (short[] expecteds, short[] actuals) Asserts that two short arrays are equal.
static void	<a href="#"><b>assertArrayEquals</b></a> (java.lang.String message, byte[] expecteds, byte[] actuals) Asserts that two byte arrays are equal.

static void	<u><b>assertArrayEquals</b></u> (java.lang.String message, char[] expecteds, char[] actuals) Asserts that two char arrays are equal.
static void	<u><b>assertArrayEquals</b></u> (java.lang.String message, int[] expecteds, int[] actuals) Asserts that two int arrays are equal.
static void	<u><b>assertArrayEquals</b></u> (java.lang.String message, long[] expecteds, long[] actuals) Asserts that two long arrays are equal.
static void	<u><b>assertArrayEquals</b></u> (java.lang.String message, java.lang.Object[] expecteds, java.lang.Object[] actuals) Asserts that two object arrays are equal.
static void	<u><b>assertArrayEquals</b></u> (java.lang.String message, short[] expecteds, short[] actuals) Asserts that two short arrays are equal.
static void	<u><b>assertEquals</b></u> (double expected, double actual) <b>Deprecated.</b> <i>Use assertEquals(double expected, double actual, double epsilon) instead</i>
static void	<u><b>assertEquals</b></u> (double expected, double actual, double delta) Asserts that two doubles or floats are equal to within a positive delta.
static void	<u><b>assertEquals</b></u> (long expected, long actual) Asserts that two longs are equal.
static void	<u><b>assertEquals</b></u> (java.lang.Object[] expecteds, java.lang.Object[] actuals) <b>Deprecated.</b> <i>use assertArrayEquals</i>
static void	<u><b>assertEquals</b></u> (java.lang.Object expected, java.lang.Object actual) Asserts that two objects are equal.
static void	<u><b>assertEquals</b></u> (java.lang.String message, double expected, double actual) <b>Deprecated.</b> <i>Use assertEquals(String message, double expected, double actual, double epsilon) instead</i>
static void	<u><b>assertEquals</b></u> (java.lang.String message, double expected, double actual, double delta) Asserts that two doubles or floats are equal to within a positive delta.
static void	<u><b>assertEquals</b></u> (java.lang.String message, long expected, long actual) Asserts that two longs are equal.
static void	<u><b>assertEquals</b></u> (java.lang.String message, java.lang.Object[] expecteds, java.lang.Object[] actuals) <b>Deprecated.</b> <i>use assertArrayEquals</i>
static void	<u><b>assertEquals</b></u> (java.lang.String message, java.lang.Object expected, java.lang.Object actual) Asserts that two objects are equal.
static void	<u><b>assertFalse</b></u> (boolean condition) Asserts that a condition is false.
static void	<u><b>assertFalse</b></u> (java.lang.String message, boolean condition) Asserts that a condition is false.
static void	<u><b>assertNotNull</b></u> (java.lang.Object object) Asserts that an object isn't null.
static void	<u><b>assertNotNull</b></u> (java.lang.String message, java.lang.Object object)

	Asserts that an object isn't null.
static void	<b><u>assertNotSame</u></b> (java.lang.Object unexpected, java.lang.Object actual) Asserts that two objects do not refer to the same object.
static void	<b><u>assertNotSame</u></b> (java.lang.String message, java.lang.Object unexpected, java.lang.Object actual) Asserts that two objects do not refer to the same object.
static void	<b><u>assertNull</u></b> (java.lang.Object object) Asserts that an object is null.
static void	<b><u>assertNull</u></b> (java.lang.String message, java.lang.Object object) Asserts that an object is null.
static void	<b><u>assertSame</u></b> (java.lang.Object expected, java.lang.Object actual) Asserts that two objects refer to the same object.
static void	<b><u>assertSame</u></b> (java.lang.String message, java.lang.Object expected, java.lang.Object actual) Asserts that two objects refer to the same object.
static <T> void	<b><u>assertThat</u></b> (java.lang.String reason, T actual, org.hamcrest.Matcher<T> matcher) Asserts that actual satisfies the condition specified by matcher.
static <T> void	<b><u>assertThat</u></b> (T actual, org.hamcrest.Matcher<T> matcher) Asserts that actual satisfies the condition specified by matcher.
static void	<b><u>assertTrue</u></b> (boolean condition) Asserts that a condition is true.
static void	<b><u>assertTrue</u></b> (java.lang.String message, boolean condition) Asserts that a condition is true.
static void	<b><u>fail</u></b> () Fails a test with no message.
static void	<b><u>fail</u></b> (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 `AssertionError` with the given message.

**Parameters:**

- message - the identifying message for the `AssertionError` (null okay)
- condition - condition to be checked

---

## assertTrue

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

Asserts that a condition is true. If it isn't it throws an `AssertionError` without a message.

**Parameters:**

- condition - condition to be checked

---

## assertFalse

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

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

**Parameters:**

- message - the identifying message for the `AssertionError` (null okay)
- condition - condition to be checked

---

## assertFalse

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

Asserts that a condition is false. If it isn't it throws an `AssertionError` without a message.

**Parameters:**

- condition - condition to be checked

---

## fail

```
public static void fail(java.lang.String message)
```

Fails a test with the given message.

**Parameters:**

- message - the identifying message for the `AssertionError` (null okay)

**See Also:**

- `AssertionError`

---

## fail

```
public static void fail()
```

Fails a test with no message.

## assertEquals

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

Asserts that two objects are equal. If they are not, an `AssertionError` is thrown with the given message. If `expected` and `actual` are `null`, they are considered equal.

### Parameters:

message - the identifying message for the AssertionError (null okay)  
 expected - expected value  
 actual - actual value

## assertEquals

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

Asserts that two objects are equal. If they are not, an `AssertionError` without a message is thrown. If `expected` and `actual` are `null`, they are considered equal.

## Parameters:

expected - expected value  
actual - the value to check against expected

## assertArrayEquals

```
public static void assertArrayEquals(java.lang.String message,  
                                     java.lang.Object[] expecteds,  
                                     java.lang.Object[] actuals)  
    throws org.junit.internal.ArrayComparisonFailure
```

Asserts that two object arrays are equal. If they are not, an `AssertionError` is thrown with the given message. If `expecteds` and `actuals` are `null`, they are considered equal.

### Parameters:

message - the identifying message for the AssertionError (null okay)  
 expecteds - Object array or array of arrays (multi-dimensional array) with expected values.  
 actuals - Object array or array of arrays (multi-dimensional array) with actual values

## Throws:

org.junit.internal.ArrayComparisonFailure

## assertArrayEquals

[illegible]

Asserts that two object arrays are equal. If they are not, an `AssertionError` is thrown. If `expected` and `actual` are `null`, they are considered equal.

**Parameters:**

`expecteds` - Object array or array of arrays (multi-dimensional array) with expected values  
`actuals` - Object array or array of arrays (multi-dimensional array) with actual values

---

## **assertArrayEquals**

```
public static void assertArrayEquals(java.lang.String message,  
                                     byte[] expecteds,  
                                     byte[] actuals)  
    throws org.junit.internal.ArrayComparisonFailure
```

Asserts that two byte arrays are equal. If they are not, an `AssertionError` is thrown with the given message.

**Parameters:**

`message` - the identifying message for the `AssertionError` (null okay)  
`expecteds` - byte array with expected values.  
`actuals` - byte array with actual values

**Throws:**

`org.junit.internal.ArrayComparisonFailure`

---

## **assertArrayEquals**

```
public static void assertArrayEquals(byte[] expecteds,  
                                     byte[] actuals)
```

Asserts that two byte arrays are equal. If they are not, an `AssertionError` is thrown.

**Parameters:**

`expecteds` - byte array with expected values.  
`actuals` - byte array with actual values

---

## **assertArrayEquals**

```
public static void assertArrayEquals(java.lang.String message,  
                                     char[] expecteds,  
                                     char[] actuals)  
    throws org.junit.internal.ArrayComparisonFailure
```

Asserts that two char arrays are equal. If they are not, an `AssertionError` is thrown with the given message.

**Parameters:**

`message` - the identifying message for the `AssertionError` (null okay)  
`expecteds` - char array with expected values.  
`actuals` - char array with actual values

**Throws:**

`org.junit.internal.ArrayComparisonFailure`

---

# assertArrayEquals

```
public static void assertArrayEquals(char[] expecteds,  
                                     char[] actuals)
```

Asserts that two char arrays are equal. If they are not, an AssertionError is thrown.

**Parameters:**

- expecteds - char array with expected values.
- actuals - char array with actual values

---

# assertArrayEquals

```
public static void assertArrayEquals(java.lang.String message,  
                                     short[] expecteds,  
                                     short[] actuals)  
    throws org.junit.internal.ArrayComparisonFailure
```

Asserts that two short arrays are equal. If they are not, an AssertionError is thrown with the given message.

**Parameters:**

- message - the identifying message for the AssertionError (null okay)
- expecteds - short array with expected values.
- actuals - short array with actual values

**Throws:**

- org.junit.internal.ArrayComparisonFailure

---

# assertArrayEquals

```
public static void assertArrayEquals(short[] expecteds,  
                                     short[] actuals)
```

Asserts that two short arrays are equal. If they are not, an AssertionError is thrown.

**Parameters:**

- expecteds - short array with expected values.
- actuals - short array with actual values

---

# assertArrayEquals

```
public static void assertArrayEquals(java.lang.String message,  
                                     int[] expecteds,  
                                     int[] actuals)  
    throws org.junit.internal.ArrayComparisonFailure
```

Asserts that two int arrays are equal. If they are not, an AssertionError is thrown with the given message.

**Parameters:**

- message - the identifying message for the AssertionError (null okay)
- expecteds - int array with expected values.
- actuals - int array with actual values

**Throws:**

`org.junit.internal.ArrayComparisonFailure`

---

## **assertArrayEquals**

```
public static void assertArrayEquals(int[] expecteds,  
                                     int[] actuals)
```

Asserts that two int arrays are equal. If they are not, an `AssertionError` is thrown.

**Parameters:**

`expecteds` - int array with expected values.

`actuals` - int array with actual values

---

## **assertArrayEquals**

```
public static void assertArrayEquals(java.lang.String message,  
                                     long[] expecteds,  
                                     long[] actuals)  
    throws org.junit.internal.ArrayComparisonFailure
```

Asserts that two long arrays are equal. If they are not, an `AssertionError` is thrown with the given message.

**Parameters:**

`message` - the identifying message for the `AssertionError` (null okay)

`expecteds` - long array with expected values.

`actuals` - long array with actual values

**Throws:**

`org.junit.internal.ArrayComparisonFailure`

---

## **assertArrayEquals**

```
public static void assertArrayEquals(long[] expecteds,  
                                     long[] actuals)
```

Asserts that two long arrays are equal. If they are not, an `AssertionError` is thrown.

**Parameters:**

`expecteds` - long array with expected values.

`actuals` - long array with actual values

---

## **assertEquals**

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

Asserts that two doubles or floats are equal to within a positive delta. If they are not, an `AssertionError` is thrown with the given message. If the expected value is infinity then the delta value is ignored. NaNs are considered equal: `assertEquals(Double.NaN, Double.NaN, *)` passes



### Parameters:

message - the identifying message for the AssertionError (null okay)

expected - expected value

actual - the value to check against expected

delta - the maximum delta between expected and actual for which both numbers are still considered equal.

---

## assertEquals

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

Asserts that two longs are equal. If they are not, an AssertionError is thrown.

### Parameters:

expected - expected long value.

actual - actual long value

---

## assertEquals

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

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

### Parameters:

message - the identifying message for the AssertionError (null okay)

expected - long expected value.

actual - long actual value

---

## assertEquals

@Deprecated

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

**Deprecated.** *Use `assertEquals(double expected, double actual, double epsilon)` instead*

---

## assertEquals

@Deprecated

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

**Deprecated.** *Use `assertEquals(String message, double expected, double actual, double epsilon)` instead*

---

# assertEquals

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

Asserts that two doubles or floats are equal to within a positive delta. If they are not, an `AssertionError` is thrown. If the expected value is infinity then the delta value is ignored. NaNs are considered equal: `assertEquals(Double.NaN, Double.NaN, *)` passes

- Parameters:**
- expected - expected value
  - actual - the value to check against expected
  - delta - the maximum delta between expected and actual for which both numbers are still considered equal.
- 

# assertNotNull

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

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

- Parameters:**
- message - the identifying message for the `AssertionError` (null okay)
  - object - Object to check or null
- 

# assertNotNull

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

Asserts that an object isn't null. If it is an `AssertionError` is thrown.

- Parameters:**
- object - Object to check or 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 `AssertionError` is thrown with the given message.

- Parameters:**
- message - the identifying message for the `AssertionError` (null okay)
  - object - Object to check or null
- 

# assertNull

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

Asserts that an object is null. If it isn't an `AssertionError` is thrown.

**Parameters:**

`object` - Object to check or null

---

## **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 `AssertionError` is thrown with the given message.

**Parameters:**

`message` - the identifying message for the `AssertionError` (null okay)

`expected` - the expected object

`actual` - the object to compare to expected

---

## **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 `AssertionError` without a message is thrown.

**Parameters:**

`expected` - the expected object

`actual` - the object to compare to expected

---

## **assertNotSame**

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

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

**Parameters:**

`message` - the identifying message for the `AssertionError` (null okay)

`unexpected` - the object you don't expect

`actual` - the object to compare to unexpected

---

## **assertNotSame**

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

Asserts that two objects do not refer to the same object. If they do refer to the same object, an

AssertionError without a message is thrown.

### Parameters:

unexpected - the object you don't expect  
actual - the object to compare to unexpected

---

## assertEquals

@Deprecated

```
public static void assertEquals(java.lang.String message,  
                                java.lang.Object[] expecteds,  
                                java.lang.Object[] actuals)
```

**Deprecated.** *use `assertArrayEquals`*

Asserts that two object arrays are equal. If they are not, an `AssertionError` is thrown with the given message. If `expecteds` and `actuals` are `null`, they are considered equal.

### Parameters:

message - the identifying message for the `AssertionError` (null okay)  
expecteds - Object array or array of arrays (multi-dimensional array) with expected values.  
actuals - Object array or array of arrays (multi-dimensional array) with actual values

---

## assertEquals

@Deprecated

```
public static void assertEquals(java.lang.Object[] expecteds,  
                                java.lang.Object[] actuals)
```

**Deprecated.** *use `assertArrayEquals`*

Asserts that two object arrays are equal. If they are not, an `AssertionError` is thrown. If `expected` and `actual` are `null`, they are considered equal.

### Parameters:

expecteds - Object array or array of arrays (multi-dimensional array) with expected values  
actuals - Object array or array of arrays (multi-dimensional array) with actual values

---

## assertThat

```
public static <T> void assertThat(T actual,  
                                  org.hamcrest.Matcher<T> matcher)
```

Asserts that `actual` satisfies the condition specified by `matcher`. If not, an `AssertionError` is thrown with information about the matcher and failing value. Example:

```
assertThat(0, is(1)); // fails:  
// failure message:  
// expected: is <1>  
// got value: <0>  
assertThat(0, is(not(1))) // passes
```

### Type Parameters:

T - the static type accepted by the matcher (this can flag obvious compile-time problems such as `assertThat(1, is("a"))`)

### Parameters:

actual - the computed value being compared

matcher - an expression, built of `Matchers`, specifying allowed values

### See Also:

`CoreMatchers`, [JUnitMatchers](#)

---

## assertThat

```
public static <T> void assertThat(java.lang.String reason,
                                   T actual,
                                   org.hamcrest.Matcher<T> matcher)
```

Asserts that `actual` satisfies the condition specified by `matcher`. If not, an `AssertionError` is thrown with the reason and information about the matcher and failing value. Example:

```
:
  assertThat("Help! Integers don't work", 0, is(1)); // fails:
    // failure message:
    // Help! Integers don't work
    // expected: is <1>
    // got value: <0>
  assertThat("Zero is one", 0, is(not(1))) // passes
```

### Type Parameters:

T - the static type accepted by the matcher (this can flag obvious compile-time problems such as `assertThat(1, is("a"))`)

### Parameters:

reason - additional information about the error

actual - the computed value being compared

matcher - an expression, built of `Matchers`, specifying allowed values

### See Also:

`CoreMatchers`, [JUnitMatchers](#)

---

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

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

[FRAMES](#) [NO FRAMES](#) [All Classes](#) [All Classes](#)

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

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

---