| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AtomicReferenceFieldUpdater.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceArray.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicStampedReference.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html)    [**NO FRAMES**](http://docs.google.com/AtomicReferenceFieldUpdater.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.util.concurrent.atomic**

Class AtomicReferenceFieldUpdater<T,V>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.util.concurrent.atomic.AtomicReferenceFieldUpdater<T,V>**

**Type Parameters:**T - The type of the object holding the updatable fieldV - The type of the field

public abstract class **AtomicReferenceFieldUpdater<T,V>**extends [Object](http://docs.google.com/java/lang/Object.html)

A reflection-based utility that enables atomic updates to designated volatile reference fields of designated classes. This class is designed for use in atomic data structures in which several reference fields of the same node are independently subject to atomic updates. For example, a tree node might be declared as

class Node {  
 private volatile Node left, right;  
  
 private static final AtomicReferenceFieldUpdater<Node, Node> leftUpdater =  
 AtomicReferenceFieldUpdater.newUpdater(Node.class, Node.class, "left");  
 private static AtomicReferenceFieldUpdater<Node, Node> rightUpdater =  
 AtomicReferenceFieldUpdater.newUpdater(Node.class, Node.class, "right");  
  
 Node getLeft() { return left; }  
 boolean compareAndSetLeft(Node expect, Node update) {  
 return leftUpdater.compareAndSet(this, expect, update);  
 }  
 // ... and so on  
 }

Note that the guarantees of the compareAndSet method in this class are weaker than in other atomic classes. Because this class cannot ensure that all uses of the field are appropriate for purposes of atomic access, it can guarantee atomicity only with respect to other invocations of compareAndSet and set on the same updater.

**Since:** 1.5

| **Constructor Summary** | |
| --- | --- |
| protected | [**AtomicReferenceFieldUpdater**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#AtomicReferenceFieldUpdater())()            Protected do-nothing constructor for use by subclasses. |

| **Method Summary** | |
| --- | --- |
| abstract  boolean | [**compareAndSet**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#compareAndSet(T,%20V,%20V))([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) expect, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) update)            Atomically sets the field of the given object managed by this updater to the given updated value if the current value == the expected value. |
| abstract  [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) | [**get**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#get(T))([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj)            Gets the current value held in the field of the given object managed by this updater. |
| [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) | [**getAndSet**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#getAndSet(T,%20V))([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) newValue)            Atomically sets the field of the given object managed by this updater to the given value and returns the old value. |
| abstract  void | [**lazySet**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#lazySet(T,%20V))([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) newValue)            Eventually sets the field of the given object managed by this updater to the given updated value. |
| static   | <U,W> [AtomicReferenceFieldUpdater](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html)<U,W> | | --- | | [**newUpdater**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#newUpdater(java.lang.Class,%20java.lang.Class,%20java.lang.String))([Class](http://docs.google.com/java/lang/Class.html)<U> tclass, [Class](http://docs.google.com/java/lang/Class.html)<W> vclass, [String](http://docs.google.com/java/lang/String.html) fieldName)            Creates and returns an updater for objects with the given field. |
| abstract  void | [**set**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#set(T,%20V))([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) newValue)            Sets the field of the given object managed by this updater to the given updated value. |
| abstract  boolean | [**weakCompareAndSet**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html#weakCompareAndSet(T,%20V,%20V))([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) expect, [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) update)            Atomically sets the field of the given object managed by this updater to the given updated value if the current value == the expected value. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### AtomicReferenceFieldUpdater

protected **AtomicReferenceFieldUpdater**()

Protected do-nothing constructor for use by subclasses.

| **Method Detail** |
| --- |

### newUpdater

public static <U,W> [AtomicReferenceFieldUpdater](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html)<U,W> **newUpdater**([Class](http://docs.google.com/java/lang/Class.html)<U> tclass,  
 [Class](http://docs.google.com/java/lang/Class.html)<W> vclass,  
 [String](http://docs.google.com/java/lang/String.html) fieldName)

Creates and returns an updater for objects with the given field. The Class arguments are needed to check that reflective types and generic types match.

**Parameters:**tclass - the class of the objects holding the field.vclass - the class of the fieldfieldName - the name of the field to be updated. **Returns:**the updater **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the field is not a volatile reference type. [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html) - with a nested reflection-based exception if the class does not hold field or is the wrong type.

### compareAndSet

public abstract boolean **compareAndSet**([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) expect,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) update)

Atomically sets the field of the given object managed by this updater to the given updated value if the current value == the expected value. This method is guaranteed to be atomic with respect to other calls to compareAndSet and set, but not necessarily with respect to other changes in the field.

**Parameters:**obj - An object whose field to conditionally setexpect - the expected valueupdate - the new value **Returns:**true if successful.

### weakCompareAndSet

public abstract boolean **weakCompareAndSet**([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) expect,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) update)

Atomically sets the field of the given object managed by this updater to the given updated value if the current value == the expected value. This method is guaranteed to be atomic with respect to other calls to compareAndSet and set, but not necessarily with respect to other changes in the field.

May [fail spuriously](http://docs.google.com/package-summary.html#Spurious) and does not provide ordering guarantees, so is only rarely an appropriate alternative to compareAndSet.

**Parameters:**obj - An object whose field to conditionally setexpect - the expected valueupdate - the new value **Returns:**true if successful.

### set

public abstract void **set**([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) newValue)

Sets the field of the given object managed by this updater to the given updated value. This operation is guaranteed to act as a volatile store with respect to subsequent invocations of compareAndSet.

**Parameters:**obj - An object whose field to setnewValue - the new value

### lazySet

public abstract void **lazySet**([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) newValue)

Eventually sets the field of the given object managed by this updater to the given updated value.

**Parameters:**obj - An object whose field to setnewValue - the new value**Since:** 1.6

### get

public abstract [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) **get**([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj)

Gets the current value held in the field of the given object managed by this updater.

**Parameters:**obj - An object whose field to get **Returns:**the current value

### getAndSet

public [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) **getAndSet**([T](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) obj,  
 [V](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html) newValue)

Atomically sets the field of the given object managed by this updater to the given value and returns the old value.

**Parameters:**obj - An object whose field to get and setnewValue - the new value **Returns:**the previous value

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AtomicReferenceFieldUpdater.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicReferenceArray.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/atomic/AtomicStampedReference.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/atomic/AtomicReferenceFieldUpdater.html)    [**NO FRAMES**](http://docs.google.com/AtomicReferenceFieldUpdater.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).