| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RandomAccess.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/Random.html)   [**NEXT CLASS**](http://docs.google.com/java/util/ResourceBundle.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/RandomAccess.html)    [**NO FRAMES**](http://docs.google.com/RandomAccess.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

## **java.util**

Interface RandomAccess

**All Known Implementing Classes:** [ArrayList](http://docs.google.com/java/util/ArrayList.html), [AttributeList](http://docs.google.com/javax/management/AttributeList.html), [CopyOnWriteArrayList](http://docs.google.com/java/util/concurrent/CopyOnWriteArrayList.html), [RoleList](http://docs.google.com/javax/management/relation/RoleList.html), [RoleUnresolvedList](http://docs.google.com/javax/management/relation/RoleUnresolvedList.html), [Stack](http://docs.google.com/java/util/Stack.html), [Vector](http://docs.google.com/java/util/Vector.html)

public interface **RandomAccess**

Marker interface used by List implementations to indicate that they support fast (generally constant time) random access. The primary purpose of this interface is to allow generic algorithms to alter their behavior to provide good performance when applied to either random or sequential access lists.

The best algorithms for manipulating random access lists (such as ArrayList) can produce quadratic behavior when applied to sequential access lists (such as LinkedList). Generic list algorithms are encouraged to check whether the given list is an instanceof this interface before applying an algorithm that would provide poor performance if it were applied to a sequential access list, and to alter their behavior if necessary to guarantee acceptable performance.

It is recognized that the distinction between random and sequential access is often fuzzy. For example, some List implementations provide asymptotically linear access times if they get huge, but constant access times in practice. Such a List implementation should generally implement this interface. As a rule of thumb, a List implementation should implement this interface if, for typical instances of the class, this loop:

for (int i=0, n=list.size(); i < n; i++)  
 list.get(i);

runs faster than this loop:

for (Iterator i=list.iterator(); i.hasNext(); )  
 i.next();

This interface is a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html).

**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RandomAccess.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/Random.html)   [**NEXT CLASS**](http://docs.google.com/java/util/ResourceBundle.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/RandomAccess.html)    [**NO FRAMES**](http://docs.google.com/RandomAccess.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

[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).