| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Group.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/security/acl/AclNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/acl/LastOwnerException.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/acl/Group.html)    [**NO FRAMES**](http://docs.google.com/Group.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.security.acl**

Interface Group

**All Superinterfaces:** [Principal](http://docs.google.com/java/security/Principal.html)

public interface **Group**extends [Principal](http://docs.google.com/java/security/Principal.html)

This interface is used to represent a group of principals. (A principal represents an entity such as an individual user or a company).

Note that Group extends Principal. Thus, either a Principal or a Group can be passed as an argument to methods containing a Principal parameter. For example, you can add either a Principal or a Group to a Group object by calling the object's addMember method, passing it the Principal or Group.

| **Method Summary** | |
| --- | --- |
| boolean | [**addMember**](http://docs.google.com/java/security/acl/Group.html#addMember(java.security.Principal))([Principal](http://docs.google.com/java/security/Principal.html) user)            Adds the specified member to the group. |
| boolean | [**isMember**](http://docs.google.com/java/security/acl/Group.html#isMember(java.security.Principal))([Principal](http://docs.google.com/java/security/Principal.html) member)            Returns true if the passed principal is a member of the group. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<? extends [Principal](http://docs.google.com/java/security/Principal.html)> | [**members**](http://docs.google.com/java/security/acl/Group.html#members())()            Returns an enumeration of the members in the group. |
| boolean | [**removeMember**](http://docs.google.com/java/security/acl/Group.html#removeMember(java.security.Principal))([Principal](http://docs.google.com/java/security/Principal.html) user)            Removes the specified member from the group. |

| **Methods inherited from interface java.security.**[**Principal**](http://docs.google.com/java/security/Principal.html) |
| --- |
| [equals](http://docs.google.com/java/security/Principal.html#equals(java.lang.Object)), [getName](http://docs.google.com/java/security/Principal.html#getName()), [hashCode](http://docs.google.com/java/security/Principal.html#hashCode()), [toString](http://docs.google.com/java/security/Principal.html#toString()) |

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

### addMember

boolean **addMember**([Principal](http://docs.google.com/java/security/Principal.html) user)

Adds the specified member to the group.

**Parameters:**user - the principal to add to this group. **Returns:**true if the member was successfully added, false if the principal was already a member.

### removeMember

boolean **removeMember**([Principal](http://docs.google.com/java/security/Principal.html) user)

Removes the specified member from the group.

**Parameters:**user - the principal to remove from this group. **Returns:**true if the principal was removed, or false if the principal was not a member.

### isMember

boolean **isMember**([Principal](http://docs.google.com/java/security/Principal.html) member)

Returns true if the passed principal is a member of the group. This method does a recursive search, so if a principal belongs to a group which is a member of this group, true is returned.

**Parameters:**member - the principal whose membership is to be checked. **Returns:**true if the principal is a member of this group, false otherwise.

### members

[Enumeration](http://docs.google.com/java/util/Enumeration.html)<? extends [Principal](http://docs.google.com/java/security/Principal.html)> **members**()

Returns an enumeration of the members in the group. The returned objects can be instances of either Principal or Group (which is a subclass of Principal).

**Returns:**an enumeration of the group members.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Group.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/security/acl/AclNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/acl/LastOwnerException.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/acl/Group.html)    [**NO FRAMES**](http://docs.google.com/Group.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).