| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Owner.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/NotOwnerException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/acl/Permission.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/acl/Owner.html)    [**NO FRAMES**](http://docs.google.com/Owner.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **java.security.acl**

Interface Owner

**All Known Subinterfaces:** [Acl](http://docs.google.com/java/security/acl/Acl.html)

public interface **Owner**

Interface for managing owners of Access Control Lists (ACLs) or ACL configurations. (Note that the Acl interface in the java.security.acl package extends this Owner interface.) The initial owner Principal should be specified as an argument to the constructor of the class implementing this interface.

**See Also:**[Acl](http://docs.google.com/java/security/acl/Acl.html)

| **Method Summary** | |
| --- | --- |
| boolean | [**addOwner**](http://docs.google.com/java/security/acl/Owner.html#addOwner(java.security.Principal,%20java.security.Principal))([Principal](http://docs.google.com/java/security/Principal.html) caller, [Principal](http://docs.google.com/java/security/Principal.html) owner)            Adds an owner. |
| boolean | [**deleteOwner**](http://docs.google.com/java/security/acl/Owner.html#deleteOwner(java.security.Principal,%20java.security.Principal))([Principal](http://docs.google.com/java/security/Principal.html) caller, [Principal](http://docs.google.com/java/security/Principal.html) owner)            Deletes an owner. |
| boolean | [**isOwner**](http://docs.google.com/java/security/acl/Owner.html#isOwner(java.security.Principal))([Principal](http://docs.google.com/java/security/Principal.html) owner)            Returns true if the given principal is an owner of the ACL. |

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

### addOwner

boolean **addOwner**([Principal](http://docs.google.com/java/security/Principal.html) caller,  
 [Principal](http://docs.google.com/java/security/Principal.html) owner)  
 throws [NotOwnerException](http://docs.google.com/java/security/acl/NotOwnerException.html)

Adds an owner. Only owners can modify ACL contents. The caller principal must be an owner of the ACL in order to invoke this method. That is, only an owner can add another owner. The initial owner is configured at ACL construction time.

**Parameters:**caller - the principal invoking this method. It must be an owner of the ACL.owner - the owner that should be added to the list of owners. **Returns:**true if successful, false if owner is already an owner. **Throws:** [NotOwnerException](http://docs.google.com/java/security/acl/NotOwnerException.html) - if the caller principal is not an owner of the ACL.

### deleteOwner

boolean **deleteOwner**([Principal](http://docs.google.com/java/security/Principal.html) caller,  
 [Principal](http://docs.google.com/java/security/Principal.html) owner)  
 throws [NotOwnerException](http://docs.google.com/java/security/acl/NotOwnerException.html),  
 [LastOwnerException](http://docs.google.com/java/security/acl/LastOwnerException.html)

Deletes an owner. If this is the last owner in the ACL, an exception is raised.

The caller principal must be an owner of the ACL in order to invoke this method.

**Parameters:**caller - the principal invoking this method. It must be an owner of the ACL.owner - the owner to be removed from the list of owners. **Returns:**true if the owner is removed, false if the owner is not part of the list of owners. **Throws:** [NotOwnerException](http://docs.google.com/java/security/acl/NotOwnerException.html) - if the caller principal is not an owner of the ACL. [LastOwnerException](http://docs.google.com/java/security/acl/LastOwnerException.html) - if there is only one owner left, so that deleteOwner would leave the ACL owner-less.

### isOwner

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

Returns true if the given principal is an owner of the ACL.

**Parameters:**owner - the principal to be checked to determine whether or not it is an owner. **Returns:**true if the passed principal is in the list of owners, false if not.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Owner.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/NotOwnerException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/acl/Permission.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/acl/Owner.html)    [**NO FRAMES**](http://docs.google.com/Owner.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).