| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AccessControlContext.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   [**NEXT CLASS**](http://docs.google.com/java/security/AccessControlException.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/AccessControlContext.html)    [**NO FRAMES**](http://docs.google.com/AccessControlContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **java.security**

Class AccessControlContext

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.security.AccessControlContext**

public final class **AccessControlContext**extends [Object](http://docs.google.com/java/lang/Object.html)

An AccessControlContext is used to make system resource access decisions based on the context it encapsulates.

More specifically, it encapsulates a context and has a single method, checkPermission, that is equivalent to the checkPermission method in the AccessController class, with one difference: The AccessControlContext checkPermission method makes access decisions based on the context it encapsulates, rather than that of the current execution thread.

Thus, the purpose of AccessControlContext is for those situations where a security check that should be made within a given context actually needs to be done from within a *different* context (for example, from within a worker thread).

An AccessControlContext is created by calling the AccessController.getContext method. The getContext method takes a "snapshot" of the current calling context, and places it in an AccessControlContext object, which it returns. A sample call is the following:

AccessControlContext acc = AccessController.getContext()

Code within a different context can subsequently call the checkPermission method on the previously-saved AccessControlContext object. A sample call is the following:

acc.checkPermission(permission)

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

| **Constructor Summary** | |
| --- | --- |
| [**AccessControlContext**](http://docs.google.com/java/security/AccessControlContext.html#AccessControlContext(java.security.AccessControlContext,%20java.security.DomainCombiner))([AccessControlContext](http://docs.google.com/java/security/AccessControlContext.html) acc, [DomainCombiner](http://docs.google.com/java/security/DomainCombiner.html) combiner)            Create a new AccessControlContext with the given AccessControlContext and DomainCombiner. |
| [**AccessControlContext**](http://docs.google.com/java/security/AccessControlContext.html#AccessControlContext(java.security.ProtectionDomain%5B%5D))([ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html)[] context)            Create an AccessControlContext with the given set of ProtectionDomains. |

| **Method Summary** | |
| --- | --- |
| void | [**checkPermission**](http://docs.google.com/java/security/AccessControlContext.html#checkPermission(java.security.Permission))([Permission](http://docs.google.com/java/security/Permission.html) perm)            Determines whether the access request indicated by the specified permission should be allowed or denied, based on the security policy currently in effect, and the context in this object. |
| boolean | [**equals**](http://docs.google.com/java/security/AccessControlContext.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Checks two AccessControlContext objects for equality. |
| [DomainCombiner](http://docs.google.com/java/security/DomainCombiner.html) | [**getDomainCombiner**](http://docs.google.com/java/security/AccessControlContext.html#getDomainCombiner())()            Get the DomainCombiner associated with this AccessControlContext. |
| int | [**hashCode**](http://docs.google.com/java/security/AccessControlContext.html#hashCode())()            Returns the hash code value for this context. |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [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** |
| --- |

### AccessControlContext

public **AccessControlContext**([ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html)[] context)

Create an AccessControlContext with the given set of ProtectionDomains. Context must not be null. Duplicate domains will be removed from the context.

**Parameters:**context - the ProtectionDomains associated with this context. The non-duplicate domains are copied from the array. Subsequent changes to the array will not affect this AccessControlContext.

### AccessControlContext

public **AccessControlContext**([AccessControlContext](http://docs.google.com/java/security/AccessControlContext.html) acc,  
 [DomainCombiner](http://docs.google.com/java/security/DomainCombiner.html) combiner)

Create a new AccessControlContext with the given AccessControlContext and DomainCombiner. This constructor associates the provided DomainCombiner with the provided AccessControlContext.

**Parameters:**acc - the AccessControlContext associated with the provided DomainCombiner.

combiner - the DomainCombiner to be associated with the provided AccessControlContext. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the provided context is null.

[SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the caller does not have permission to invoke this constructor.**Since:** 1.3

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

### getDomainCombiner

public [DomainCombiner](http://docs.google.com/java/security/DomainCombiner.html) **getDomainCombiner**()

Get the DomainCombiner associated with this AccessControlContext.

**Returns:**the DomainCombiner associated with this AccessControlContext, or null if there is none. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the caller does not have permission to get the DomainCombiner associated with this AccessControlContext.**Since:** 1.3

### checkPermission

public void **checkPermission**([Permission](http://docs.google.com/java/security/Permission.html) perm)  
 throws [AccessControlException](http://docs.google.com/java/security/AccessControlException.html)

Determines whether the access request indicated by the specified permission should be allowed or denied, based on the security policy currently in effect, and the context in this object. The request is allowed only if every ProtectionDomain in the context implies the permission. Otherwise the request is denied.

This method quietly returns if the access request is permitted, or throws a suitable AccessControlException otherwise.

**Parameters:**perm - the requested permission. **Throws:** [AccessControlException](http://docs.google.com/java/security/AccessControlException.html) - if the specified permission is not permitted, based on the current security policy and the context encapsulated by this object. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the permission to check for is null.

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Checks two AccessControlContext objects for equality. Checks that *obj* is an AccessControlContext and has the same set of ProtectionDomains as this context.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the object we are testing for equality with this object. **Returns:**true if *obj* is an AccessControlContext, and has the same set of ProtectionDomains as this context, false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns the hash code value for this context. The hash code is computed by exclusive or-ing the hash code of all the protection domains in the context together.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this context.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AccessControlContext.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   [**NEXT CLASS**](http://docs.google.com/java/security/AccessControlException.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/AccessControlContext.html)    [**NO FRAMES**](http://docs.google.com/AccessControlContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).