| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/UnresolvedPermission.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/UnrecoverableKeyException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/URIParameter.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/UnresolvedPermission.html)    [**NO FRAMES**](http://docs.google.com/UnresolvedPermission.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **java.security**

Class UnresolvedPermission

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Guard](http://docs.google.com/java/security/Guard.html)

public final class **UnresolvedPermission**extends [Permission](http://docs.google.com/java/security/Permission.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

The UnresolvedPermission class is used to hold Permissions that were "unresolved" when the Policy was initialized. An unresolved permission is one whose actual Permission class does not yet exist at the time the Policy is initialized (see below).

The policy for a Java runtime (specifying which permissions are available for code from various principals) is represented by a Policy object. Whenever a Policy is initialized or refreshed, Permission objects of appropriate classes are created for all permissions allowed by the Policy.

Many permission class types referenced by the policy configuration are ones that exist locally (i.e., ones that can be found on CLASSPATH). Objects for such permissions can be instantiated during Policy initialization. For example, it is always possible to instantiate a java.io.FilePermission, since the FilePermission class is found on the CLASSPATH.

Other permission classes may not yet exist during Policy initialization. For example, a referenced permission class may be in a JAR file that will later be loaded. For each such class, an UnresolvedPermission is instantiated. Thus, an UnresolvedPermission is essentially a "placeholder" containing information about the permission.

Later, when code calls AccessController.checkPermission on a permission of a type that was previously unresolved, but whose class has since been loaded, previously-unresolved permissions of that type are "resolved". That is, for each such UnresolvedPermission, a new object of the appropriate class type is instantiated, based on the information in the UnresolvedPermission.

To instantiate the new class, UnresolvedPermission assumes the class provides a zero, one, and/or two-argument constructor. The zero-argument constructor would be used to instantiate a permission without a name and without actions. A one-arg constructor is assumed to take a String name as input, and a two-arg constructor is assumed to take a String name and String actions as input. UnresolvedPermission may invoke a constructor with a null name and/or actions. If an appropriate permission constructor is not available, the UnresolvedPermission is ignored and the relevant permission will not be granted to executing code.

The newly created permission object replaces the UnresolvedPermission, which is removed.

Note that the getName method for an UnresolvedPermission returns the type (class name) for the underlying permission that has not been resolved.

**See Also:**[Permission](http://docs.google.com/java/security/Permission.html), [Permissions](http://docs.google.com/java/security/Permissions.html), [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html), [Policy](http://docs.google.com/java/security/Policy.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.security.UnresolvedPermission)

| **Constructor Summary** | |
| --- | --- |
| [**UnresolvedPermission**](http://docs.google.com/java/security/UnresolvedPermission.html#UnresolvedPermission(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.security.cert.Certificate%5B%5D))([String](http://docs.google.com/java/lang/String.html) type, [String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) actions, [Certificate](http://docs.google.com/java/security/cert/Certificate.html)[] certs)            Creates a new UnresolvedPermission containing the permission information needed later to actually create a Permission of the specified class, when the permission is resolved. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/security/UnresolvedPermission.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Checks two UnresolvedPermission objects for equality. |
| [String](http://docs.google.com/java/lang/String.html) | [**getActions**](http://docs.google.com/java/security/UnresolvedPermission.html#getActions())()            Returns the canonical string representation of the actions, which currently is the empty string "", since there are no actions for an UnresolvedPermission. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUnresolvedActions**](http://docs.google.com/java/security/UnresolvedPermission.html#getUnresolvedActions())()            Get the actions for the underlying permission that has not been resolved. |
| [Certificate](http://docs.google.com/java/security/cert/Certificate.html)[] | [**getUnresolvedCerts**](http://docs.google.com/java/security/UnresolvedPermission.html#getUnresolvedCerts())()            Get the signer certificates (without any supporting chain) for the underlying permission that has not been resolved. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUnresolvedName**](http://docs.google.com/java/security/UnresolvedPermission.html#getUnresolvedName())()            Get the target name of the underlying permission that has not been resolved. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUnresolvedType**](http://docs.google.com/java/security/UnresolvedPermission.html#getUnresolvedType())()            Get the type (class name) of the underlying permission that has not been resolved. |
| int | [**hashCode**](http://docs.google.com/java/security/UnresolvedPermission.html#hashCode())()            Returns the hash code value for this object. |
| boolean | [**implies**](http://docs.google.com/java/security/UnresolvedPermission.html#implies(java.security.Permission))([Permission](http://docs.google.com/java/security/Permission.html) p)            This method always returns false for unresolved permissions. |
| [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html) | [**newPermissionCollection**](http://docs.google.com/java/security/UnresolvedPermission.html#newPermissionCollection())()            Returns a new PermissionCollection object for storing UnresolvedPermission objects. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/security/UnresolvedPermission.html#toString())()            Returns a string describing this UnresolvedPermission. |

| **Methods inherited from class java.security.**[**Permission**](http://docs.google.com/java/security/Permission.html) |
| --- |
| [checkGuard](http://docs.google.com/java/security/Permission.html#checkGuard(java.lang.Object)), [getName](http://docs.google.com/java/security/Permission.html#getName()) |

| **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()), [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** |
| --- |

### UnresolvedPermission

public **UnresolvedPermission**([String](http://docs.google.com/java/lang/String.html) type,  
 [String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) actions,  
 [Certificate](http://docs.google.com/java/security/cert/Certificate.html)[] certs)

Creates a new UnresolvedPermission containing the permission information needed later to actually create a Permission of the specified class, when the permission is resolved.

**Parameters:**type - the class name of the Permission class that will be created when this unresolved permission is resolved.name - the name of the permission.actions - the actions of the permission.certs - the certificates the permission's class was signed with. This is a list of certificate chains, where each chain is composed of a signer certificate and optionally its supporting certificate chain. Each chain is ordered bottom-to-top (i.e., with the signer certificate first and the (root) certificate authority last). The signer certificates are copied from the array. Subsequent changes to the array will not affect this UnsolvedPermission.

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

### implies

public boolean **implies**([Permission](http://docs.google.com/java/security/Permission.html) p)

This method always returns false for unresolved permissions. That is, an UnresolvedPermission is never considered to imply another permission.

**Specified by:**[implies](http://docs.google.com/java/security/Permission.html#implies(java.security.Permission)) in class [Permission](http://docs.google.com/java/security/Permission.html) **Parameters:**p - the permission to check against. **Returns:**false.

### equals

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

Checks two UnresolvedPermission objects for equality. Checks that *obj* is an UnresolvedPermission, and has the same type (class) name, permission name, actions, and certificates as this object.

To determine certificate equality, this method only compares actual signer certificates. Supporting certificate chains are not taken into consideration by this method.

**Specified by:**[equals](http://docs.google.com/java/security/Permission.html#equals(java.lang.Object)) in class [Permission](http://docs.google.com/java/security/Permission.html) **Parameters:**obj - the object we are testing for equality with this object. **Returns:**true if obj is an UnresolvedPermission, and has the same type (class) name, permission name, actions, and certificates as this object.**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 object.

**Specified by:**[hashCode](http://docs.google.com/java/security/Permission.html#hashCode()) in class [Permission](http://docs.google.com/java/security/Permission.html) **Returns:**a hash code value for this object.**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)

### getActions

public [String](http://docs.google.com/java/lang/String.html) **getActions**()

Returns the canonical string representation of the actions, which currently is the empty string "", since there are no actions for an UnresolvedPermission. That is, the actions for the permission that will be created when this UnresolvedPermission is resolved may be non-null, but an UnresolvedPermission itself is never considered to have any actions.

**Specified by:**[getActions](http://docs.google.com/java/security/Permission.html#getActions()) in class [Permission](http://docs.google.com/java/security/Permission.html) **Returns:**the empty string "".

### getUnresolvedType

public [String](http://docs.google.com/java/lang/String.html) **getUnresolvedType**()

Get the type (class name) of the underlying permission that has not been resolved.

**Returns:**the type (class name) of the underlying permission that has not been resolved**Since:** 1.5

### getUnresolvedName

public [String](http://docs.google.com/java/lang/String.html) **getUnresolvedName**()

Get the target name of the underlying permission that has not been resolved.

**Returns:**the target name of the underlying permission that has not been resolved, or null, if there is no targe name**Since:** 1.5

### getUnresolvedActions

public [String](http://docs.google.com/java/lang/String.html) **getUnresolvedActions**()

Get the actions for the underlying permission that has not been resolved.

**Returns:**the actions for the underlying permission that has not been resolved, or null if there are no actions**Since:** 1.5

### getUnresolvedCerts

public [Certificate](http://docs.google.com/java/security/cert/Certificate.html)[] **getUnresolvedCerts**()

Get the signer certificates (without any supporting chain) for the underlying permission that has not been resolved.

**Returns:**the signer certificates for the underlying permission that has not been resolved, or null, if there are no signer certificates. Returns a new array each time this method is called.**Since:** 1.5

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string describing this UnresolvedPermission. The convention is to specify the class name, the permission name, and the actions, in the following format: '(unresolved "ClassName" "name" "actions")'.

**Overrides:**[toString](http://docs.google.com/java/security/Permission.html#toString()) in class [Permission](http://docs.google.com/java/security/Permission.html) **Returns:**information about this UnresolvedPermission.

### newPermissionCollection

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

Returns a new PermissionCollection object for storing UnresolvedPermission objects.

**Overrides:**[newPermissionCollection](http://docs.google.com/java/security/Permission.html#newPermissionCollection()) in class [Permission](http://docs.google.com/java/security/Permission.html) **Returns:**a new PermissionCollection object suitable for storing UnresolvedPermissions.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/UnresolvedPermission.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/UnrecoverableKeyException.html)   [**NEXT CLASS**](http://docs.google.com/java/security/URIParameter.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/UnresolvedPermission.html)    [**NO FRAMES**](http://docs.google.com/UnresolvedPermission.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [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).