| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Policy.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/Permissions.html)   [**NEXT CLASS**](http://docs.google.com/java/security/Policy.Parameters.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/Policy.html)    [**NO FRAMES**](http://docs.google.com/Policy.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **java.security**

Class Policy

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

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

A Policy object is responsible for determining whether code executing in the Java runtime environment has permission to perform a security-sensitive operation.

There is only one Policy object installed in the runtime at any given time. A Policy object can be installed by calling the setPolicy method. The installed Policy object can be obtained by calling the getPolicy method.

If no Policy object has been installed in the runtime, a call to getPolicy installs an instance of the default Policy implementation (a default subclass implementation of this abstract class). The default Policy implementation can be changed by setting the value of the "policy.provider" security property (in the Java security properties file) to the fully qualified name of the desired Policy subclass implementation. The Java security properties file is located in the file named <JAVA\_HOME>/lib/security/java.security. <JAVA\_HOME> refers to the value of the java.home system property, and specifies the directory where the JRE is installed.

Application code can directly subclass Policy to provide a custom implementation. In addition, an instance of a Policy object can be constructed by invoking one of the getInstance factory methods with a standard type. The default policy type is "JavaPolicy". See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for a list of standard Policy types.

Once a Policy instance has been installed (either by default, or by calling setPolicy), the Java runtime invokes its implies when it needs to determine whether executing code (encapsulated in a ProtectionDomain) can perform SecurityManager-protected operations. How a Policy object retrieves its policy data is up to the Policy implementation itself. The policy data may be stored, for example, in a flat ASCII file, in a serialized binary file of the Policy class, or in a database.

The refresh method causes the policy object to refresh/reload its data. This operation is implementation-dependent. For example, if the policy object stores its data in configuration files, calling refresh will cause it to re-read the configuration policy files. If a refresh operation is not supported, this method does nothing. Note that refreshed policy may not have an effect on classes in a particular ProtectionDomain. This is dependent on the Policy provider's implementation of the implies method and its PermissionCollection caching strategy.

**See Also:**[Provider](http://docs.google.com/java/security/Provider.html), [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html), [Permission](http://docs.google.com/java/security/Permission.html)

| **Nested Class Summary** | |
| --- | --- |
| static interface | [**Policy.Parameters**](http://docs.google.com/java/security/Policy.Parameters.html)            This represents a marker interface for Policy parameters. |

| **Field Summary** | |
| --- | --- |
| static [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html) | [**UNSUPPORTED\_EMPTY\_COLLECTION**](http://docs.google.com/java/security/Policy.html#UNSUPPORTED_EMPTY_COLLECTION)            A read-only empty PermissionCollection instance. |

| **Constructor Summary** | |
| --- | --- |
| [**Policy**](http://docs.google.com/java/security/Policy.html#Policy())() |

| **Method Summary** | |
| --- | --- |
| static [Policy](http://docs.google.com/java/security/Policy.html) | [**getInstance**](http://docs.google.com/java/security/Policy.html#getInstance(java.lang.String,%20java.security.Policy.Parameters))([String](http://docs.google.com/java/lang/String.html) type, [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) params)            Returns a Policy object of the specified type. |
| static [Policy](http://docs.google.com/java/security/Policy.html) | [**getInstance**](http://docs.google.com/java/security/Policy.html#getInstance(java.lang.String,%20java.security.Policy.Parameters,%20java.security.Provider))([String](http://docs.google.com/java/lang/String.html) type, [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) params, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Returns a Policy object of the specified type. |
| static [Policy](http://docs.google.com/java/security/Policy.html) | [**getInstance**](http://docs.google.com/java/security/Policy.html#getInstance(java.lang.String,%20java.security.Policy.Parameters,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) type, [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) params, [String](http://docs.google.com/java/lang/String.html) provider)            Returns a Policy object of the specified type. |
| [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) | [**getParameters**](http://docs.google.com/java/security/Policy.html#getParameters())()            Return Policy parameters. |
| [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html) | [**getPermissions**](http://docs.google.com/java/security/Policy.html#getPermissions(java.security.CodeSource))([CodeSource](http://docs.google.com/java/security/CodeSource.html) codesource)            Return a PermissionCollection object containing the set of permissions granted to the specified CodeSource. |
| [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html) | [**getPermissions**](http://docs.google.com/java/security/Policy.html#getPermissions(java.security.ProtectionDomain))([ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) domain)            Return a PermissionCollection object containing the set of permissions granted to the specified ProtectionDomain. |
| static [Policy](http://docs.google.com/java/security/Policy.html) | [**getPolicy**](http://docs.google.com/java/security/Policy.html#getPolicy())()            Returns the installed Policy object. |
| [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/java/security/Policy.html#getProvider())()            Return the Provider of this Policy. |
| [String](http://docs.google.com/java/lang/String.html) | [**getType**](http://docs.google.com/java/security/Policy.html#getType())()            Return the type of this Policy. |
| boolean | [**implies**](http://docs.google.com/java/security/Policy.html#implies(java.security.ProtectionDomain,%20java.security.Permission))([ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) domain, [Permission](http://docs.google.com/java/security/Permission.html) permission)            Evaluates the global policy for the permissions granted to the ProtectionDomain and tests whether the permission is granted. |
| void | [**refresh**](http://docs.google.com/java/security/Policy.html#refresh())()            Refreshes/reloads the policy configuration. |
| static void | [**setPolicy**](http://docs.google.com/java/security/Policy.html#setPolicy(java.security.Policy))([Policy](http://docs.google.com/java/security/Policy.html) p)            Sets the system-wide Policy object. |

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

| **Field Detail** |
| --- |

### UNSUPPORTED\_EMPTY\_COLLECTION

public static final [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html) **UNSUPPORTED\_EMPTY\_COLLECTION**

A read-only empty PermissionCollection instance.

**Since:** 1.6

| **Constructor Detail** |
| --- |

### Policy

public **Policy**()

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

### getPolicy

public static [Policy](http://docs.google.com/java/security/Policy.html) **getPolicy**()

Returns the installed Policy object. This value should not be cached, as it may be changed by a call to setPolicy. This method first calls SecurityManager.checkPermission with a SecurityPermission("getPolicy") permission to ensure it's ok to get the Policy object..

**Returns:**the installed Policy. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method doesn't allow getting the Policy object.**See Also:**[SecurityManager.checkPermission(Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [setPolicy(java.security.Policy)](http://docs.google.com/java/security/Policy.html#setPolicy(java.security.Policy))

### setPolicy

public static void **setPolicy**([Policy](http://docs.google.com/java/security/Policy.html) p)

Sets the system-wide Policy object. This method first calls SecurityManager.checkPermission with a SecurityPermission("setPolicy") permission to ensure it's ok to set the Policy.

**Parameters:**p - the new system Policy object. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method doesn't allow setting the Policy.**See Also:**[SecurityManager.checkPermission(Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [getPolicy()](http://docs.google.com/java/security/Policy.html#getPolicy())

### getInstance

public static [Policy](http://docs.google.com/java/security/Policy.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) type,  
 [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) params)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns a Policy object of the specified type.

This method traverses the list of registered security providers, starting with the most preferred Provider. A new Policy object encapsulating the PolicySpi implementation from the first Provider that supports the specified type is returned.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**type - the specified Policy type. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for a list of standard Policy types.params - parameters for the Policy, which may be null. **Returns:**the new Policy object. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the caller does not have permission to get a Policy instance for the specified type. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified type is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified parameters are not understood by the PolicySpi implementation from the selected Provider. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if no Provider supports a PolicySpi implementation for the specified type.**Since:** 1.6 **See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [Policy](http://docs.google.com/java/security/Policy.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) type,  
 [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) params,  
 [String](http://docs.google.com/java/lang/String.html) provider)  
 throws [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html),  
 [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns a Policy object of the specified type.

A new Policy object encapsulating the PolicySpi implementation from the specified provider is returned. The specified provider must be registered in the provider list.

Note that the list of registered providers may be retrieved via the [Security.getProviders()](http://docs.google.com/java/security/Security.html#getProviders()) method.

**Parameters:**type - the specified Policy type. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for a list of standard Policy types.params - parameters for the Policy, which may be null.provider - the provider. **Returns:**the new Policy object. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the caller does not have permission to get a Policy instance for the specified type. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified type is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified provider is null or empty, or if the specified parameters are not understood by the PolicySpi implementation from the specified provider. [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - if the specified provider is not registered in the security provider list. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if the specified provider does not support a PolicySpi implementation for the specified type.**Since:** 1.6 **See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getInstance

public static [Policy](http://docs.google.com/java/security/Policy.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) type,  
 [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) params,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Returns a Policy object of the specified type.

A new Policy object encapsulating the PolicySpi implementation from the specified Provider object is returned. Note that the specified Provider object does not have to be registered in the provider list.

**Parameters:**type - the specified Policy type. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for a list of standard Policy types.params - parameters for the Policy, which may be null.provider - the Provider. **Returns:**the new Policy object. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the caller does not have permission to get a Policy instance for the specified type. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified type is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified Provider is null, or if the specified parameters are not understood by the PolicySpi implementation from the specified Provider. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if the specified Provider does not support a PolicySpi implementation for the specified type.**Since:** 1.6 **See Also:**[Provider](http://docs.google.com/java/security/Provider.html)

### getProvider

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

Return the Provider of this Policy.

This Policy instance will only have a Provider if it was obtained via a call to Policy.getInstance. Otherwise this method returns null.

**Returns:**the Provider of this Policy, or null.**Since:** 1.6

### getType

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

Return the type of this Policy.

This Policy instance will only have a type if it was obtained via a call to Policy.getInstance. Otherwise this method returns null.

**Returns:**the type of this Policy, or null.**Since:** 1.6

### getParameters

public [Policy.Parameters](http://docs.google.com/java/security/Policy.Parameters.html) **getParameters**()

Return Policy parameters.

This Policy instance will only have parameters if it was obtained via a call to Policy.getInstance. Otherwise this method returns null.

**Returns:**Policy parameters, or null.**Since:** 1.6

### getPermissions

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

Return a PermissionCollection object containing the set of permissions granted to the specified CodeSource.

Applications are discouraged from calling this method since this operation may not be supported by all policy implementations. Applications should solely rely on the implies method to perform policy checks. If an application absolutely must call a getPermissions method, it should call getPermissions(ProtectionDomain).

The default implementation of this method returns Policy.UNSUPPORTED\_EMPTY\_COLLECTION. This method can be overridden if the policy implementation can return a set of permissions granted to a CodeSource.

**Parameters:**codesource - the CodeSource to which the returned PermissionCollection has been granted. **Returns:**a set of permissions granted to the specified CodeSource. If this operation is supported, the returned set of permissions must be a new mutable instance and it must support heterogeneous Permission types. If this operation is not supported, Policy.UNSUPPORTED\_EMPTY\_COLLECTION is returned.

### getPermissions

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

Return a PermissionCollection object containing the set of permissions granted to the specified ProtectionDomain.

Applications are discouraged from calling this method since this operation may not be supported by all policy implementations. Applications should rely on the implies method to perform policy checks.

The default implementation of this method first retrieves the permissions returned via getPermissions(CodeSource) (the CodeSource is taken from the specified ProtectionDomain), as well as the permissions located inside the specified ProtectionDomain. All of these permissions are then combined and returned in a new PermissionCollection object. If getPermissions(CodeSource) returns Policy.UNSUPPORTED\_EMPTY\_COLLECTION, then this method returns the permissions contained inside the specified ProtectionDomain in a new PermissionCollection object.

This method can be overridden if the policy implementation supports returning a set of permissions granted to a ProtectionDomain.

**Parameters:**domain - the ProtectionDomain to which the returned PermissionCollection has been granted. **Returns:**a set of permissions granted to the specified ProtectionDomain. If this operation is supported, the returned set of permissions must be a new mutable instance and it must support heterogeneous Permission types. If this operation is not supported, Policy.UNSUPPORTED\_EMPTY\_COLLECTION is returned.**Since:** 1.4

### implies

public boolean **implies**([ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) domain,  
 [Permission](http://docs.google.com/java/security/Permission.html) permission)

Evaluates the global policy for the permissions granted to the ProtectionDomain and tests whether the permission is granted.

**Parameters:**domain - the ProtectionDomain to testpermission - the Permission object to be tested for implication. **Returns:**true if "permission" is a proper subset of a permission granted to this ProtectionDomain.**Since:** 1.4 **See Also:**[ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html)

### refresh

public void **refresh**()

Refreshes/reloads the policy configuration. The behavior of this method depends on the implementation. For example, calling refresh on a file-based policy will cause the file to be re-read.

The default implementation of this method does nothing. This method should be overridden if a refresh operation is supported by the policy implementation.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Policy.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/Permissions.html)   [**NEXT CLASS**](http://docs.google.com/java/security/Policy.Parameters.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/Policy.html)    [**NO FRAMES**](http://docs.google.com/Policy.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).