| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | Class | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/security/package-use.html)    [**NO FRAMES**](http://docs.google.com/package-use.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Package**

**java.security**

| Packages that use [java.security](http://docs.google.com/java/security/package-summary.html) | |
| --- | --- |
| [**java.awt**](#3znysh7) | Contains all of the classes for creating user interfaces and for painting graphics and images. |
| [**java.io**](#2et92p0) | Provides for system input and output through data streams, serialization and the file system. |
| [**java.lang**](#tyjcwt) | Provides classes that are fundamental to the design of the Java programming language. |
| [**java.lang.instrument**](#3dy6vkm) | Provides services that allow Java programming language agents to instrument programs running on the JVM. |
| [**java.lang.management**](#1t3h5sf) | Provides the management interface for monitoring and management of the Java virtual machine as well as the operating system on which the Java virtual machine is running. |
| [**java.lang.reflect**](#4d34og8) | Provides classes and interfaces for obtaining reflective information about classes and objects. |
| [**java.net**](#2s8eyo1) | Provides the classes for implementing networking applications. |
| [**java.security**](#17dp8vu) | Provides the classes and interfaces for the security framework. |
| [**java.security.acl**](#3rdcrjn) | The classes and interfaces in this package have been superseded by classes in the java.security package. |
| [**java.security.cert**](#26in1rg) | Provides classes and interfaces for parsing and managing certificates, certificate revocation lists (CRLs), and certification paths. |
| [**java.security.interfaces**](#lnxbz9) | Provides interfaces for generating RSA (Rivest, Shamir and Adleman AsymmetricCipher algorithm) keys as defined in the RSA Laboratory Technical Note PKCS#1, and DSA (Digital Signature Algorithm) keys as defined in NIST's FIPS-186. |
| [**java.security.spec**](#35nkun2) | Provides classes and interfaces for key specifications and algorithm parameter specifications. |
| [**java.sql**](#1ksv4uv) | Provides the API for accessing and processing data stored in a data source (usually a relational database) using the JavaTM programming language. |
| [**java.util**](#44sinio) | Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array). |
| [**java.util.concurrent**](#2jxsxqh) | Utility classes commonly useful in concurrent programming. |
| [**java.util.jar**](#z337ya) | Provides classes for reading and writing the JAR (Java ARchive) file format, which is based on the standard ZIP file format with an optional manifest file. |
| [**java.util.logging**](#3j2qqm3) | Provides the classes and interfaces of the JavaTM 2 platform's core logging facilities. |
| [**javax.crypto**](#1y810tw) | Provides the classes and interfaces for cryptographic operations. |
| [**javax.crypto.interfaces**](#4i7ojhp) | Provides interfaces for Diffie-Hellman keys as defined in RSA Laboratories' PKCS #3. |
| [**javax.crypto.spec**](#2xcytpi) | Provides classes and interfaces for key specifications and algorithm parameter specifications. |
| [**javax.management**](#1ci93xb) | Provides the core classes for the Java Management Extensions. |
| [**javax.management.loading**](#3whwml4) | Provides the classes which implement advanced dynamic loading. |
| [**javax.management.remote**](#2bn6wsx) | Interfaces for remote access to JMX MBean servers. |
| [**javax.net.ssl**](#qsh70q) | Provides classes for the secure socket package. |
| [**javax.security.auth**](#3as4poj) | This package provides a framework for authentication and authorization. |
| [**javax.security.auth.kerberos**](#1pxezwc) | This package contains utility classes related to the Kerberos network authentication protocol. |
| [**javax.security.auth.login**](#49x2ik5) | This package provides a pluggable authentication framework. |
| [**javax.security.auth.x500**](#2p2csry) | This package contains the classes that should be used to store X500 Principal and X500 Private Crendentials in a *Subject*. |
| [**javax.sound.sampled**](#147n2zr) | Provides interfaces and classes for capture, processing, and playback of sampled audio data. |
| [**javax.xml.crypto**](#3o7alnk) | Common classes for XML cryptography. |
| [**javax.xml.crypto.dsig**](#23ckvvd) | Classes for generating and validating XML digital signatures. |
| [**javax.xml.crypto.dsig.dom**](#ihv636) | DOM-specific classes for the [javax.xml.crypto.dsig](http://docs.google.com/javax/xml/crypto/dsig/package-summary.html) package. |
| [**javax.xml.crypto.dsig.keyinfo**](#32hioqz) | Classes for parsing and processing [KeyInfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/KeyInfo.html) elements and structures. |
| [**javax.xml.ws**](#1hmsyys) | This package contains the core JAX-WS APIs. |
| [**org.ietf.jgss**](#41mghml) | This package presents a framework that allows application developers to make use of security services like authentication, data integrity and data confidentiality from a variety of underlying security mechanisms like Kerberos, using a unified API. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.awt](http://docs.google.com/java/awt/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.awt)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.awt)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.awt)            Abstract class for representing access to a system resource. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.io](http://docs.google.com/java/io/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.io)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.io)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.io)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#java.io)            Abstract class representing a collection of Permission objects. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.lang](http://docs.google.com/java/lang/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.lang)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.lang)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.lang)            Abstract class for representing access to a system resource. |
| [**ProtectionDomain**](http://docs.google.com/java/security/class-use/ProtectionDomain.html#java.lang)             This ProtectionDomain class encapsulates the characteristics of a domain, which encloses a set of classes whose instances are granted a set of permissions when being executed on behalf of a given set of Principals. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.lang.instrument](http://docs.google.com/java/lang/instrument/package-summary.html) | |
| --- | --- |
| [**ProtectionDomain**](http://docs.google.com/java/security/class-use/ProtectionDomain.html#java.lang.instrument)             This ProtectionDomain class encapsulates the characteristics of a domain, which encloses a set of classes whose instances are granted a set of permissions when being executed on behalf of a given set of Principals. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.lang.management](http://docs.google.com/java/lang/management/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.lang.management)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.lang.management)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.lang.management)            Abstract class for representing access to a system resource. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.lang.reflect](http://docs.google.com/java/lang/reflect/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.lang.reflect)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.lang.reflect)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.lang.reflect)            Abstract class for representing access to a system resource. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.net](http://docs.google.com/java/net/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.net)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**CodeSource**](http://docs.google.com/java/security/class-use/CodeSource.html#java.net)            This class extends the concept of a codebase to encapsulate not only the location (URL) but also the certificate chains that were used to verify signed code originating from that location. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.net)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.net)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#java.net)            Abstract class representing a collection of Permission objects. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#java.net)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |
| [**SecureClassLoader**](http://docs.google.com/java/security/class-use/SecureClassLoader.html#java.net)            This class extends ClassLoader with additional support for defining classes with an associated code source and permissions which are retrieved by the system policy by default. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.security](http://docs.google.com/java/security/package-summary.html) | |
| --- | --- |
| [**AccessControlContext**](http://docs.google.com/java/security/class-use/AccessControlContext.html#java.security)            An AccessControlContext is used to make system resource access decisions based on the context it encapsulates. |
| [**AccessControlException**](http://docs.google.com/java/security/class-use/AccessControlException.html#java.security)             This exception is thrown by the AccessController to indicate that a requested access (to a critical system resource such as the file system or the network) is denied. |
| [**AlgorithmParameterGenerator**](http://docs.google.com/java/security/class-use/AlgorithmParameterGenerator.html#java.security)            The AlgorithmParameterGenerator class is used to generate a set of parameters to be used with a certain algorithm. |
| [**AlgorithmParameterGeneratorSpi**](http://docs.google.com/java/security/class-use/AlgorithmParameterGeneratorSpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the AlgorithmParameterGenerator class, which is used to generate a set of parameters to be used with a certain algorithm. |
| [**AlgorithmParameters**](http://docs.google.com/java/security/class-use/AlgorithmParameters.html#java.security)            This class is used as an opaque representation of cryptographic parameters. |
| [**AlgorithmParametersSpi**](http://docs.google.com/java/security/class-use/AlgorithmParametersSpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the AlgorithmParameters class, which is used to manage algorithm parameters. |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.security)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Certificate**](http://docs.google.com/java/security/class-use/Certificate.html#java.security)  **Deprecated.** *A new certificate handling package is created in the Java platform. This Certificate interface is entirely deprecated and is here to allow for a smooth transition to the new package.* |
| [**CodeSigner**](http://docs.google.com/java/security/class-use/CodeSigner.html#java.security)            This class encapsulates information about a code signer. |
| [**CodeSource**](http://docs.google.com/java/security/class-use/CodeSource.html#java.security)            This class extends the concept of a codebase to encapsulate not only the location (URL) but also the certificate chains that were used to verify signed code originating from that location. |
| [**DigestException**](http://docs.google.com/java/security/class-use/DigestException.html#java.security)            This is the generic Message Digest exception. |
| [**DomainCombiner**](http://docs.google.com/java/security/class-use/DomainCombiner.html#java.security)            A DomainCombiner provides a means to dynamically update the ProtectionDomains associated with the current AccessControlContext. |
| [**GeneralSecurityException**](http://docs.google.com/java/security/class-use/GeneralSecurityException.html#java.security)            The GeneralSecurityException class is a generic security exception class that provides type safety for all the security-related exception classes that extend from it. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.security)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Identity**](http://docs.google.com/java/security/class-use/Identity.html#java.security)  **Deprecated.** *This class is no longer used. Its functionality has been replaced by java.security.KeyStore, the java.security.cert package, and java.security.Principal.* |
| [**IdentityScope**](http://docs.google.com/java/security/class-use/IdentityScope.html#java.security)  **Deprecated.** *This class is no longer used. Its functionality has been replaced by java.security.KeyStore, the java.security.cert package, and java.security.Principal.* |
| [**InvalidAlgorithmParameterException**](http://docs.google.com/java/security/class-use/InvalidAlgorithmParameterException.html#java.security)            This is the exception for invalid or inappropriate algorithm parameters. |
| [**InvalidKeyException**](http://docs.google.com/java/security/class-use/InvalidKeyException.html#java.security)            This is the exception for invalid Keys (invalid encoding, wrong length, uninitialized, etc). |
| [**InvalidParameterException**](http://docs.google.com/java/security/class-use/InvalidParameterException.html#java.security)            This exception, designed for use by the JCA/JCE engine classes, is thrown when an invalid parameter is passed to a method. |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#java.security)            The Key interface is the top-level interface for all keys. |
| [**KeyException**](http://docs.google.com/java/security/class-use/KeyException.html#java.security)            This is the basic key exception. |
| [**KeyFactory**](http://docs.google.com/java/security/class-use/KeyFactory.html#java.security)            Key factories are used to convert *keys* (opaque cryptographic keys of type Key) into *key specifications* (transparent representations of the underlying key material), and vice versa. |
| [**KeyFactorySpi**](http://docs.google.com/java/security/class-use/KeyFactorySpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the KeyFactory class. |
| [**KeyManagementException**](http://docs.google.com/java/security/class-use/KeyManagementException.html#java.security)            This is the general key management exception for all operations dealing with key management. |
| [**KeyPair**](http://docs.google.com/java/security/class-use/KeyPair.html#java.security)            This class is a simple holder for a key pair (a public key and a private key). |
| [**KeyPairGenerator**](http://docs.google.com/java/security/class-use/KeyPairGenerator.html#java.security)            The KeyPairGenerator class is used to generate pairs of public and private keys. |
| [**KeyPairGeneratorSpi**](http://docs.google.com/java/security/class-use/KeyPairGeneratorSpi.html#java.security)             This class defines the *Service Provider Interface* (**SPI**) for the KeyPairGenerator class, which is used to generate pairs of public and private keys. |
| [**KeyRep.Type**](http://docs.google.com/java/security/class-use/KeyRep.Type.html#java.security)            Key type. |
| [**KeyStore**](http://docs.google.com/java/security/class-use/KeyStore.html#java.security)            This class represents a storage facility for cryptographic keys and certificates. |
| [**KeyStore.Builder**](http://docs.google.com/java/security/class-use/KeyStore.Builder.html#java.security)            A description of a to-be-instantiated KeyStore object. |
| [**KeyStore.Entry**](http://docs.google.com/java/security/class-use/KeyStore.Entry.html#java.security)            A marker interface for KeyStore entry types. |
| [**KeyStore.LoadStoreParameter**](http://docs.google.com/java/security/class-use/KeyStore.LoadStoreParameter.html#java.security)            A marker interface for KeyStore [load](http://docs.google.com/java/security/KeyStore.html#load(java.security.KeyStore.LoadStoreParameter)) and [store](http://docs.google.com/java/security/KeyStore.html#store(java.security.KeyStore.LoadStoreParameter)) parameters. |
| [**KeyStore.ProtectionParameter**](http://docs.google.com/java/security/class-use/KeyStore.ProtectionParameter.html#java.security)            A marker interface for keystore protection parameters. |
| [**KeyStoreException**](http://docs.google.com/java/security/class-use/KeyStoreException.html#java.security)            This is the generic KeyStore exception. |
| [**KeyStoreSpi**](http://docs.google.com/java/security/class-use/KeyStoreSpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the KeyStore class. |
| [**MessageDigest**](http://docs.google.com/java/security/class-use/MessageDigest.html#java.security)            This MessageDigest class provides applications the functionality of a message digest algorithm, such as MD5 or SHA. |
| [**MessageDigestSpi**](http://docs.google.com/java/security/class-use/MessageDigestSpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the MessageDigest class, which provides the functionality of a message digest algorithm, such as MD5 or SHA. |
| [**NoSuchAlgorithmException**](http://docs.google.com/java/security/class-use/NoSuchAlgorithmException.html#java.security)            This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#java.security)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.security)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#java.security)            Abstract class representing a collection of Permission objects. |
| [**Policy**](http://docs.google.com/java/security/class-use/Policy.html#java.security)            A Policy object is responsible for determining whether code executing in the Java runtime environment has permission to perform a security-sensitive operation. |
| [**Policy.Parameters**](http://docs.google.com/java/security/class-use/Policy.Parameters.html#java.security)            This represents a marker interface for Policy parameters. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#java.security)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |
| [**PrivateKey**](http://docs.google.com/java/security/class-use/PrivateKey.html#java.security)            A private key. |
| [**PrivilegedAction**](http://docs.google.com/java/security/class-use/PrivilegedAction.html#java.security)            A computation to be performed with privileges enabled. |
| [**PrivilegedActionException**](http://docs.google.com/java/security/class-use/PrivilegedActionException.html#java.security)            This exception is thrown by doPrivileged(PrivilegedExceptionAction) and doPrivileged(PrivilegedExceptionAction, AccessControlContext context) to indicate that the action being performed threw a checked exception. |
| [**PrivilegedExceptionAction**](http://docs.google.com/java/security/class-use/PrivilegedExceptionAction.html#java.security)            A computation to be performed with privileges enabled, that throws one or more checked exceptions. |
| [**ProtectionDomain**](http://docs.google.com/java/security/class-use/ProtectionDomain.html#java.security)             This ProtectionDomain class encapsulates the characteristics of a domain, which encloses a set of classes whose instances are granted a set of permissions when being executed on behalf of a given set of Principals. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#java.security)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |
| [**Provider.Service**](http://docs.google.com/java/security/class-use/Provider.Service.html#java.security)            The description of a security service. |
| [**PublicKey**](http://docs.google.com/java/security/class-use/PublicKey.html#java.security)            A public key. |
| [**SecureRandom**](http://docs.google.com/java/security/class-use/SecureRandom.html#java.security)            This class provides a cryptographically strong random number generator (RNG). |
| [**SecureRandomSpi**](http://docs.google.com/java/security/class-use/SecureRandomSpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the SecureRandom class. |
| [**Signature**](http://docs.google.com/java/security/class-use/Signature.html#java.security)            This Signature class is used to provide applications the functionality of a digital signature algorithm. |
| [**SignatureException**](http://docs.google.com/java/security/class-use/SignatureException.html#java.security)            This is the generic Signature exception. |
| [**SignatureSpi**](http://docs.google.com/java/security/class-use/SignatureSpi.html#java.security)            This class defines the *Service Provider Interface* (**SPI**) for the Signature class, which is used to provide the functionality of a digital signature algorithm. |
| [**Timestamp**](http://docs.google.com/java/security/class-use/Timestamp.html#java.security)            This class encapsulates information about a signed timestamp. |
| [**UnrecoverableEntryException**](http://docs.google.com/java/security/class-use/UnrecoverableEntryException.html#java.security)            This exception is thrown if an entry in the keystore cannot be recovered. |
| [**UnrecoverableKeyException**](http://docs.google.com/java/security/class-use/UnrecoverableKeyException.html#java.security)            This exception is thrown if a key in the keystore cannot be recovered. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.security.acl](http://docs.google.com/java/security/acl/package-summary.html) | |
| --- | --- |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#java.security.acl)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.security.cert](http://docs.google.com/java/security/cert/package-summary.html) | |
| --- | --- |
| [**GeneralSecurityException**](http://docs.google.com/java/security/class-use/GeneralSecurityException.html#java.security.cert)            The GeneralSecurityException class is a generic security exception class that provides type safety for all the security-related exception classes that extend from it. |
| [**InvalidAlgorithmParameterException**](http://docs.google.com/java/security/class-use/InvalidAlgorithmParameterException.html#java.security.cert)            This is the exception for invalid or inappropriate algorithm parameters. |
| [**InvalidKeyException**](http://docs.google.com/java/security/class-use/InvalidKeyException.html#java.security.cert)            This is the exception for invalid Keys (invalid encoding, wrong length, uninitialized, etc). |
| [**KeyStore**](http://docs.google.com/java/security/class-use/KeyStore.html#java.security.cert)            This class represents a storage facility for cryptographic keys and certificates. |
| [**KeyStoreException**](http://docs.google.com/java/security/class-use/KeyStoreException.html#java.security.cert)            This is the generic KeyStore exception. |
| [**NoSuchAlgorithmException**](http://docs.google.com/java/security/class-use/NoSuchAlgorithmException.html#java.security.cert)            This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#java.security.cert)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#java.security.cert)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#java.security.cert)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |
| [**PublicKey**](http://docs.google.com/java/security/class-use/PublicKey.html#java.security.cert)            A public key. |
| [**SignatureException**](http://docs.google.com/java/security/class-use/SignatureException.html#java.security.cert)            This is the generic Signature exception. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.security.interfaces](http://docs.google.com/java/security/interfaces/package-summary.html) | |
| --- | --- |
| [**InvalidParameterException**](http://docs.google.com/java/security/class-use/InvalidParameterException.html#java.security.interfaces)            This exception, designed for use by the JCA/JCE engine classes, is thrown when an invalid parameter is passed to a method. |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#java.security.interfaces)            The Key interface is the top-level interface for all keys. |
| [**PrivateKey**](http://docs.google.com/java/security/class-use/PrivateKey.html#java.security.interfaces)            A private key. |
| [**PublicKey**](http://docs.google.com/java/security/class-use/PublicKey.html#java.security.interfaces)            A public key. |
| [**SecureRandom**](http://docs.google.com/java/security/class-use/SecureRandom.html#java.security.interfaces)            This class provides a cryptographically strong random number generator (RNG). |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.security.spec](http://docs.google.com/java/security/spec/package-summary.html) | |
| --- | --- |
| [**GeneralSecurityException**](http://docs.google.com/java/security/class-use/GeneralSecurityException.html#java.security.spec)            The GeneralSecurityException class is a generic security exception class that provides type safety for all the security-related exception classes that extend from it. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.sql](http://docs.google.com/java/sql/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.sql)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.sql)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.sql)            Abstract class for representing access to a system resource. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.util](http://docs.google.com/java/util/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.util)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.util)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.util)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#java.util)            Abstract class representing a collection of Permission objects. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) | |
| --- | --- |
| [**PrivilegedAction**](http://docs.google.com/java/security/class-use/PrivilegedAction.html#java.util.concurrent)            A computation to be performed with privileges enabled. |
| [**PrivilegedExceptionAction**](http://docs.google.com/java/security/class-use/PrivilegedExceptionAction.html#java.util.concurrent)            A computation to be performed with privileges enabled, that throws one or more checked exceptions. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.util.jar](http://docs.google.com/java/util/jar/package-summary.html) | |
| --- | --- |
| [**CodeSigner**](http://docs.google.com/java/security/class-use/CodeSigner.html#java.util.jar)            This class encapsulates information about a code signer. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [java.util.logging](http://docs.google.com/java/util/logging/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#java.util.logging)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#java.util.logging)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#java.util.logging)            Abstract class for representing access to a system resource. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.crypto](http://docs.google.com/javax/crypto/package-summary.html) | |
| --- | --- |
| [**AlgorithmParameters**](http://docs.google.com/java/security/class-use/AlgorithmParameters.html#javax.crypto)            This class is used as an opaque representation of cryptographic parameters. |
| [**GeneralSecurityException**](http://docs.google.com/java/security/class-use/GeneralSecurityException.html#javax.crypto)            The GeneralSecurityException class is a generic security exception class that provides type safety for all the security-related exception classes that extend from it. |
| [**InvalidAlgorithmParameterException**](http://docs.google.com/java/security/class-use/InvalidAlgorithmParameterException.html#javax.crypto)            This is the exception for invalid or inappropriate algorithm parameters. |
| [**InvalidKeyException**](http://docs.google.com/java/security/class-use/InvalidKeyException.html#javax.crypto)            This is the exception for invalid Keys (invalid encoding, wrong length, uninitialized, etc). |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#javax.crypto)            The Key interface is the top-level interface for all keys. |
| [**NoSuchAlgorithmException**](http://docs.google.com/java/security/class-use/NoSuchAlgorithmException.html#javax.crypto)            This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#javax.crypto)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#javax.crypto)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |
| [**SecureRandom**](http://docs.google.com/java/security/class-use/SecureRandom.html#javax.crypto)            This class provides a cryptographically strong random number generator (RNG). |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.crypto.interfaces](http://docs.google.com/javax/crypto/interfaces/package-summary.html) | |
| --- | --- |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#javax.crypto.interfaces)            The Key interface is the top-level interface for all keys. |
| [**PrivateKey**](http://docs.google.com/java/security/class-use/PrivateKey.html#javax.crypto.interfaces)            A private key. |
| [**PublicKey**](http://docs.google.com/java/security/class-use/PublicKey.html#javax.crypto.interfaces)            A public key. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.crypto.spec](http://docs.google.com/javax/crypto/spec/package-summary.html) | |
| --- | --- |
| [**InvalidKeyException**](http://docs.google.com/java/security/class-use/InvalidKeyException.html#javax.crypto.spec)            This is the exception for invalid Keys (invalid encoding, wrong length, uninitialized, etc). |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#javax.crypto.spec)            The Key interface is the top-level interface for all keys. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.management](http://docs.google.com/javax/management/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.management)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.management)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.management)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#javax.management)            Abstract class representing a collection of Permission objects. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.management.loading](http://docs.google.com/javax/management/loading/package-summary.html) | |
| --- | --- |
| [**SecureClassLoader**](http://docs.google.com/java/security/class-use/SecureClassLoader.html#javax.management.loading)            This class extends ClassLoader with additional support for defining classes with an associated code source and permissions which are retrieved by the system policy by default. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.management.remote](http://docs.google.com/javax/management/remote/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.management.remote)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.management.remote)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.management.remote)            Abstract class for representing access to a system resource. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#javax.management.remote)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.net.ssl](http://docs.google.com/javax/net/ssl/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.net.ssl)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.net.ssl)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**InvalidAlgorithmParameterException**](http://docs.google.com/java/security/class-use/InvalidAlgorithmParameterException.html#javax.net.ssl)            This is the exception for invalid or inappropriate algorithm parameters. |
| [**KeyManagementException**](http://docs.google.com/java/security/class-use/KeyManagementException.html#javax.net.ssl)            This is the general key management exception for all operations dealing with key management. |
| [**KeyStore**](http://docs.google.com/java/security/class-use/KeyStore.html#javax.net.ssl)            This class represents a storage facility for cryptographic keys and certificates. |
| [**KeyStore.Builder**](http://docs.google.com/java/security/class-use/KeyStore.Builder.html#javax.net.ssl)            A description of a to-be-instantiated KeyStore object. |
| [**KeyStoreException**](http://docs.google.com/java/security/class-use/KeyStoreException.html#javax.net.ssl)            This is the generic KeyStore exception. |
| [**NoSuchAlgorithmException**](http://docs.google.com/java/security/class-use/NoSuchAlgorithmException.html#javax.net.ssl)            This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#javax.net.ssl)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.net.ssl)            Abstract class for representing access to a system resource. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#javax.net.ssl)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |
| [**PrivateKey**](http://docs.google.com/java/security/class-use/PrivateKey.html#javax.net.ssl)            A private key. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#javax.net.ssl)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |
| [**SecureRandom**](http://docs.google.com/java/security/class-use/SecureRandom.html#javax.net.ssl)            This class provides a cryptographically strong random number generator (RNG). |
| [**UnrecoverableKeyException**](http://docs.google.com/java/security/class-use/UnrecoverableKeyException.html#javax.net.ssl)            This exception is thrown if a key in the keystore cannot be recovered. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.security.auth](http://docs.google.com/javax/security/auth/package-summary.html) | |
| --- | --- |
| [**AccessControlContext**](http://docs.google.com/java/security/class-use/AccessControlContext.html#javax.security.auth)            An AccessControlContext is used to make system resource access decisions based on the context it encapsulates. |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.security.auth)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**DomainCombiner**](http://docs.google.com/java/security/class-use/DomainCombiner.html#javax.security.auth)            A DomainCombiner provides a means to dynamically update the ProtectionDomains associated with the current AccessControlContext. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.security.auth)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.security.auth)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#javax.security.auth)            Abstract class representing a collection of Permission objects. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#javax.security.auth)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |
| [**PrivilegedAction**](http://docs.google.com/java/security/class-use/PrivilegedAction.html#javax.security.auth)            A computation to be performed with privileges enabled. |
| [**PrivilegedActionException**](http://docs.google.com/java/security/class-use/PrivilegedActionException.html#javax.security.auth)            This exception is thrown by doPrivileged(PrivilegedExceptionAction) and doPrivileged(PrivilegedExceptionAction, AccessControlContext context) to indicate that the action being performed threw a checked exception. |
| [**PrivilegedExceptionAction**](http://docs.google.com/java/security/class-use/PrivilegedExceptionAction.html#javax.security.auth)            A computation to be performed with privileges enabled, that throws one or more checked exceptions. |
| [**ProtectionDomain**](http://docs.google.com/java/security/class-use/ProtectionDomain.html#javax.security.auth)             This ProtectionDomain class encapsulates the characteristics of a domain, which encloses a set of classes whose instances are granted a set of permissions when being executed on behalf of a given set of Principals. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.security.auth.kerberos](http://docs.google.com/javax/security/auth/kerberos/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.security.auth.kerberos)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.security.auth.kerberos)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#javax.security.auth.kerberos)            The Key interface is the top-level interface for all keys. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.security.auth.kerberos)            Abstract class for representing access to a system resource. |
| [**PermissionCollection**](http://docs.google.com/java/security/class-use/PermissionCollection.html#javax.security.auth.kerberos)            Abstract class representing a collection of Permission objects. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#javax.security.auth.kerberos)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.security.auth.login](http://docs.google.com/javax/security/auth/login/package-summary.html) | |
| --- | --- |
| [**GeneralSecurityException**](http://docs.google.com/java/security/class-use/GeneralSecurityException.html#javax.security.auth.login)            The GeneralSecurityException class is a generic security exception class that provides type safety for all the security-related exception classes that extend from it. |
| [**NoSuchAlgorithmException**](http://docs.google.com/java/security/class-use/NoSuchAlgorithmException.html#javax.security.auth.login)            This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#javax.security.auth.login)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#javax.security.auth.login)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.security.auth.x500](http://docs.google.com/javax/security/auth/x500/package-summary.html) | |
| --- | --- |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#javax.security.auth.x500)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |
| [**PrivateKey**](http://docs.google.com/java/security/class-use/PrivateKey.html#javax.security.auth.x500)            A private key. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.sound.sampled](http://docs.google.com/javax/sound/sampled/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.sound.sampled)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.sound.sampled)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.sound.sampled)            Abstract class for representing access to a system resource. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.xml.crypto](http://docs.google.com/javax/xml/crypto/package-summary.html) | |
| --- | --- |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#javax.xml.crypto)            The Key interface is the top-level interface for all keys. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.xml.crypto.dsig](http://docs.google.com/javax/xml/crypto/dsig/package-summary.html) | |
| --- | --- |
| [**InvalidAlgorithmParameterException**](http://docs.google.com/java/security/class-use/InvalidAlgorithmParameterException.html#javax.xml.crypto.dsig)            This is the exception for invalid or inappropriate algorithm parameters. |
| [**NoSuchAlgorithmException**](http://docs.google.com/java/security/class-use/NoSuchAlgorithmException.html#javax.xml.crypto.dsig)            This exception is thrown when a particular cryptographic algorithm is requested but is not available in the environment. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#javax.xml.crypto.dsig)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#javax.xml.crypto.dsig)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.xml.crypto.dsig.dom](http://docs.google.com/javax/xml/crypto/dsig/dom/package-summary.html) | |
| --- | --- |
| [**Key**](http://docs.google.com/java/security/class-use/Key.html#javax.xml.crypto.dsig.dom)            The Key interface is the top-level interface for all keys. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.xml.crypto.dsig.keyinfo](http://docs.google.com/javax/xml/crypto/dsig/keyinfo/package-summary.html) | |
| --- | --- |
| [**KeyException**](http://docs.google.com/java/security/class-use/KeyException.html#javax.xml.crypto.dsig.keyinfo)            This is the basic key exception. |
| [**NoSuchProviderException**](http://docs.google.com/java/security/class-use/NoSuchProviderException.html#javax.xml.crypto.dsig.keyinfo)            This exception is thrown when a particular security provider is requested but is not available in the environment. |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#javax.xml.crypto.dsig.keyinfo)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |
| [**PublicKey**](http://docs.google.com/java/security/class-use/PublicKey.html#javax.xml.crypto.dsig.keyinfo)            A public key. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [javax.xml.ws](http://docs.google.com/javax/xml/ws/package-summary.html) | |
| --- | --- |
| [**BasicPermission**](http://docs.google.com/java/security/class-use/BasicPermission.html#javax.xml.ws)            The BasicPermission class extends the Permission class, and can be used as the base class for permissions that want to follow the same naming convention as BasicPermission. |
| [**Guard**](http://docs.google.com/java/security/class-use/Guard.html#javax.xml.ws)             This interface represents a guard, which is an object that is used to protect access to another object. |
| [**Permission**](http://docs.google.com/java/security/class-use/Permission.html#javax.xml.ws)            Abstract class for representing access to a system resource. |
| [**Principal**](http://docs.google.com/java/security/class-use/Principal.html#javax.xml.ws)            This interface represents the abstract notion of a principal, which can be used to represent any entity, such as an individual, a corporation, and a login id. |

| Classes in [java.security](http://docs.google.com/java/security/package-summary.html) used by [org.ietf.jgss](http://docs.google.com/org/ietf/jgss/package-summary.html) | |
| --- | --- |
| [**Provider**](http://docs.google.com/java/security/class-use/Provider.html#org.ietf.jgss)            This class represents a "provider" for the Java Security API, where a provider implements some or all parts of Java Security. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | Class | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/security/package-use.html)    [**NO FRAMES**](http://docs.google.com/package-use.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).