

javax.security.auth

Class Subject

java.lang.Object
javax.security.auth.Subject

All Implemented Interfaces:

Serializable

```
public final class Subject
extends Object
implements Serializable
```

A Subject represents a grouping of related information for a single entity, such as a person. Such information includes the Subject's identities as well as its security-related attributes (passwords and cryptographic keys, for example).

Subjects may potentially have multiple identities. Each identity is represented as a Principal within the Subject. Principals simply bind names to a Subject. For example, a Subject that happens to be a person, Alice, might have two Principals: one which binds "Alice Bar", the name on her driver license, to the Subject, and another which binds, "999-99-9999", the number on her student identification card, to the Subject. Both Principals refer to the same Subject even though each has a different name.

A Subject may also own security-related attributes, which are referred to as credentials. Sensitive credentials that require special protection, such as private cryptographic keys, are stored within a private credential Set. Credentials intended to be shared, such as public key certificates or Kerberos server tickets are stored within a public credential Set. Different permissions are required to access and modify the different credential Sets.

To retrieve all the Principals associated with a Subject, invoke the `getPrincipals` method. To retrieve all the public or private credentials belonging to a Subject, invoke the `getPublicCredentials` method or `getPrivateCredentials` method, respectively. To modify the returned Set of Principals and credentials, use the methods defined in the Set class. For example:

```
Subject subject;
Principal principal;
Object credential;

// add a Principal and credential to the Subject
subject.getPrincipals().add(principal);
subject.getPublicCredentials().add(credential);
```

This Subject class implements `Serializable`. While the Principals associated with the Subject are serialized, the credentials associated with the Subject are not. Note that the `java.security.Principal` class does not implement `Serializable`. Therefore all concrete Principal implementations associated with Subjects must implement `Serializable`.

See Also:

Principal, DomainCombiner, Serialized Form

Constructor Summary

Constructors

Constructor and Description

Subject() Create an instance of a Subject with an empty Set of Principals and empty Sets of public and private credentials.
Subject(boolean readOnly, Set<? extends Principal> principals, Set<?> pubCredentials, Set<?>

```
> privCredentials)
Create an instance of a Subject with Principals and credentials.
```

Method Summary

Methods

Modifier and Type	Method and Description
static <T> T	doAs (Subject subject, PrivilegedAction <T> action) Perform work as a particular Subject.
static <T> T	doAs (Subject subject, PrivilegedExceptionAction <T> action) Perform work as a particular Subject.
static <T> T	doAsPrivileged (Subject subject, PrivilegedAction <T> action, AccessControlContext acc) Perform privileged work as a particular Subject.
static <T> T	doAsPrivileged (Subject subject, PrivilegedExceptionAction <T> action, AccessControlContext acc) Perform privileged work as a particular Subject.
boolean	equals (Object o) Compares the specified Object with this Subject for equality.
Set < Principal >	getPrincipals () Return the Set of Principals associated with this Subject.
<T extends Principal > Set <T>	getPrincipals (Class <T> c) Return a Set of Principals associated with this Subject that are instances or subclasses of the specified Class.
Set < Object >	getPrivateCredentials () Return the Set of private credentials held by this Subject.
<T> Set <T>	getPrivateCredentials (Class <T> c) Return a Set of private credentials associated with this Subject that are instances or subclasses of the specified Class.
Set < Object >	getPublicCredentials () Return the Set of public credentials held by this Subject.
<T> Set <T>	getPublicCredentials (Class <T> c) Return a Set of public credentials associated with this Subject that are instances or subclasses of the specified Class.
static Subject	getSubject (AccessControlContext acc) Get the Subject associated with the provided AccessControlContext.
int	hashCode () Returns a hashcode for this Subject.
boolean	isReadOnly () Query whether this Subject is read-only.
void	setReadOnly () Set this Subject to be read-only.
String	toString () Return the String representation of this Subject.

Methods inherited from class [java.lang.Object](#)

[clone](#), [finalize](#), [getClass](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Subject

```
public Subject()
```

Create an instance of a Subject with an empty Set of Principals and empty Sets of public and private credentials.

The newly constructed Sets check whether this Subject has been set read-only before permitting subsequent modifications. The newly created Sets also prevent illegal modifications by ensuring that callers have sufficient permissions.

To modify the Principals Set, the caller must have `AuthPermission("modifyPrincipals")`. To modify the public credential Set, the caller must have `AuthPermission("modifyPublicCredentials")`. To modify the private credential Set, the caller must have `AuthPermission("modifyPrivateCredentials")`.

Subject

```
public Subject(boolean readOnly,  
               Set<? extends Principal> principals,  
               Set<?> pubCredentials,  
               Set<?> privCredentials)
```

Create an instance of a Subject with Principals and credentials.

The Principals and credentials from the specified Sets are copied into newly constructed Sets. These newly created Sets check whether this Subject has been set read-only before permitting subsequent modifications. The newly created Sets also prevent illegal modifications by ensuring that callers have sufficient permissions.

To modify the Principals Set, the caller must have `AuthPermission("modifyPrincipals")`. To modify the public credential Set, the caller must have `AuthPermission("modifyPublicCredentials")`. To modify the private credential Set, the caller must have `AuthPermission("modifyPrivateCredentials")`.

Parameters:

`readOnly` - true if the Subject is to be read-only, and false otherwise.

`principals` - the Set of Principals to be associated with this Subject.

`pubCredentials` - the Set of public credentials to be associated with this Subject.

`privCredentials` - the Set of private credentials to be associated with this Subject.

Throws:

`NullPointerException` - if the specified principals, pubCredentials, or privCredentials are null.

Method Detail

setReadOnly

```
public void setReadOnly()
```

Set this Subject to be read-only.

Modifications (additions and removals) to this Subject's Principal Set and credential Sets will be disallowed. The destroy operation on this Subject's credentials will still be permitted.

Subsequent attempts to modify the Subject's Principal and credential Sets will result in an `IllegalStateException` being thrown. Also, once a Subject is read-only, it can not be reset to being writable again.

Throws:

[SecurityException](#) - if the caller does not have permission to set this Subject to be read-only.

isReadOnly

```
public boolean isReadOnly()
```

Query whether this Subject is read-only.

Returns:

true if this Subject is read-only, false otherwise.

getSubject

```
public static Subject getSubject(AccessControlContext acc)
```

Get the Subject associated with the provided AccessControlContext.

The AccessControlContext may contain many Subjects (from nested doAs calls). In this situation, the most recent Subject associated with the AccessControlContext is returned.

Parameters:

acc - the AccessControlContext from which to retrieve the Subject.

Returns:

the Subject associated with the provided AccessControlContext, or null if no Subject is associated with the provided AccessControlContext.

Throws:

[SecurityException](#) - if the caller does not have permission to get the Subject.

[NullPointerException](#) - if the provided AccessControlContext is null.

doAs

```
public static <T> T doAs(Subject subject,  
    PrivilegedAction<T> action)
```

Perform work as a particular Subject.

This method first retrieves the current Thread's AccessControlContext via AccessController.getContext, and then instantiates a new AccessControlContext using the retrieved context along with a new SubjectDomainCombiner (constructed using the provided Subject). Finally, this method invokes AccessController.doPrivileged, passing it the provided PrivilegedAction, as well as the newly constructed AccessControlContext.

Parameters:

subject - the Subject that the specified action will run as. This parameter may be null.

action - the code to be run as the specified Subject.

Returns:

the value returned by the PrivilegedAction's run method.

Throws:

[NullPointerException](#) - if the PrivilegedAction is null.

[SecurityException](#) - if the caller does not have permission to invoke this method.

doAs

```
public static <T> T doAs(Subject subject,  
    PrivilegedExceptionAction<T> action)  
    throws PrivilegedActionException
```

Perform work as a particular Subject.

This method first retrieves the current Thread's AccessControlContext via AccessController.getContext, and then instantiates a new AccessControlContext using the retrieved context along with a new SubjectDomainCombiner (constructed using the provided Subject). Finally, this method invokes AccessController.doPrivileged, passing it the provided PrivilegedExceptionAction, as well as the newly constructed AccessControlContext.

Parameters:

`subject` - the Subject that the specified action will run as. This parameter may be null.

`action` - the code to be run as the specified Subject.

Returns:

the value returned by the PrivilegedExceptionAction's run method.

Throws:

`PrivilegedActionException` - if the PrivilegedExceptionAction.run method throws a checked exception.

`NullPointerException` - if the specified PrivilegedExceptionAction is null.

`SecurityException` - if the caller does not have permission to invoke this method.

doAsPrivileged

```
public static <T> T doAsPrivileged(Subject subject,  
    PrivilegedAction<T> action,  
    AccessControlContext acc)
```

Perform privileged work as a particular Subject.

This method behaves exactly as Subject.doAs, except that instead of retrieving the current Thread's AccessControlContext, it uses the provided AccessControlContext. If the provided AccessControlContext is null, this method instantiates a new AccessControlContext with an empty collection of ProtectionDomains.

Parameters:

`subject` - the Subject that the specified action will run as. This parameter may be null.

`action` - the code to be run as the specified Subject.

`acc` - the AccessControlContext to be tied to the specified *subject* and *action*.

Returns:

the value returned by the PrivilegedAction's run method.

Throws:

`NullPointerException` - if the PrivilegedAction is null.

`SecurityException` - if the caller does not have permission to invoke this method.

doAsPrivileged

```
public static <T> T doAsPrivileged(Subject subject,  
    PrivilegedExceptionAction<T> action,  
    AccessControlContext acc)  
    throws PrivilegedActionException
```

Perform privileged work as a particular **Subject**.

This method behaves exactly as **Subject.doAs**, except that instead of retrieving the current **Thread**'s **AccessControlContext**, it uses the provided **AccessControlContext**. If the provided **AccessControlContext** is **null**, this method instantiates a new **AccessControlContext** with an empty collection of **ProtectionDomains**.

Parameters:

subject - the **Subject** that the specified action will run as. This parameter may be **null**.

action - the code to be run as the specified **Subject**.

acc - the **AccessControlContext** to be tied to the specified *subject* and *action*.

Returns:

the value returned by the **PrivilegedExceptionAction**'s **run** method.

Throws:

PrivilegedActionException - if the **PrivilegedExceptionAction.run** method throws a checked exception.

NullPointerException - if the specified **PrivilegedExceptionAction** is **null**.

SecurityException - if the caller does not have permission to invoke this method.

getPrincipals

```
public Set<Principal> getPrincipals()
```

Return the **Set** of **Principals** associated with this **Subject**. Each **Principal** represents an identity for this **Subject**.

The returned **Set** is backed by this **Subject**'s internal **Principal Set**. Any modification to the returned **Set** affects the internal **Principal Set** as well.

Returns:

The **Set** of **Principals** associated with this **Subject**.

getPrincipals

```
public <T extends Principal> Set<T> getPrincipals(Class<T> c)
```

Return a **Set** of **Principals** associated with this **Subject** that are instances or subclasses of the specified **Class**.

The returned **Set** is not backed by this **Subject**'s internal **Principal Set**. A new **Set** is created and returned for each method invocation. Modifications to the returned **Set** will not affect the internal **Principal Set**.

Parameters:

c - the returned **Set** of **Principals** will all be instances of this class.

Returns:

a **Set** of **Principals** that are instances of the specified **Class**.

Throws:

NullPointerException - if the specified **Class** is **null**.

getPublicCredentials

```
public Set<Object> getPublicCredentials()
```

Return the Set of public credentials held by this Subject.

The returned Set is backed by this Subject's internal public Credential Set. Any modification to the returned Set affects the internal public Credential Set as well.

Returns:

A Set of public credentials held by this Subject.

getPrivateCredentials

```
public Set<Object> getPrivateCredentials()
```

Return the Set of private credentials held by this Subject.

The returned Set is backed by this Subject's internal private Credential Set. Any modification to the returned Set affects the internal private Credential Set as well.

A caller requires permissions to access the Credentials in the returned Set, or to modify the Set itself. A `SecurityException` is thrown if the caller does not have the proper permissions.

While iterating through the Set, a `SecurityException` is thrown if the caller does not have permission to access a particular Credential. The `Iterator` is nevertheless advanced to next element in the Set.

Returns:

A Set of private credentials held by this Subject.

getPublicCredentials

```
public <T> Set<T> getPublicCredentials(Class<T> c)
```

Return a Set of public credentials associated with this Subject that are instances or subclasses of the specified Class.

The returned Set is not backed by this Subject's internal public Credential Set. A new Set is created and returned for each method invocation. Modifications to the returned Set will not affect the internal public Credential Set.

Parameters:

c - the returned Set of public credentials will all be instances of this class.

Returns:

a Set of public credentials that are instances of the specified Class.

Throws:

`NullPointerException` - if the specified Class is null.

getPrivateCredentials

```
public <T> Set<T> getPrivateCredentials(Class<T> c)
```

Return a Set of private credentials associated with this Subject that are instances or subclasses of the specified Class.

The caller must have permission to access all of the requested Credentials, or a `SecurityException` will be thrown.

The returned Set is not backed by this Subject's internal private Credential Set. A new Set is created and returned for each method invocation. Modifications to the returned Set will not affect the internal private Credential Set.

Parameters:

c - the returned Set of private credentials will all be instances of this class.

Returns:

a Set of private credentials that are instances of the specified Class.

Throws:

[NullPointerException](#) - if the specified Class is null.

equals

```
public boolean equals(Object o)
```

Compares the specified Object with this Subject for equality. Returns true if the given object is also a Subject and the two Subject instances are equivalent. More formally, two Subject instances are equal if their Principal and Credential Sets are equal.

Overrides:

[equals](#) in class [Object](#)

Parameters:

o - Object to be compared for equality with this Subject.

Returns:

true if the specified Object is equal to this Subject.

Throws:

[SecurityException](#) - if the caller does not have permission to access the private credentials for this Subject, or if the caller does not have permission to access the private credentials for the provided Subject.

See Also:

[Object.hashCode\(\)](#), [HashMap](#)

toString

```
public String toString()
```

Return the String representation of this Subject.

Overrides:

[toString](#) in class [Object](#)

Returns:

the String representation of this Subject.

hashCode

```
public int hashCode()
```

Returns a hashCode for this Subject.

Overrides:

[hashCode](#) in class [Object](#)

Returns:

a hashCode for this Subject.

Throws:

[SecurityException](#) - if the caller does not have permission to access this Subject's private credentials.

See Also:

[Object.equals\(java.lang.Object\)](#), [System.identityHashCode\(java.lang.Object\)](#)

[Overview](#) [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

Java™ Platform
Standard Ed. 7

Prev Class **Next Class** [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#) Detail: [Field](#) | [Constr](#) | [Method](#)

Submit a bug or feature

For further API reference and developer documentation, see [Java SE Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright © 1993, 2020, Oracle and/or its affiliates. All rights reserved. Use is subject to [license terms](#). Also see the [documentation redistribution policy](#). Modify [Cookie Preferences](#). Modify [Ad Choices](#).