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

java.security

Class BasicPermission

java.lang.Object
    java.security.Permission
        java.security.BasicPermission

All Implemented Interfaces:

Serializable, Guard

Direct Known Subclasses:

AudioPermission, AuthPermission, AWTPermission, DelegationPermission, JAXBPermission, LinkPermission, LoggingPermission, ManagementPermission, MBeanServerPermission, MBeanTrustPermission, NetPermission, PropertyPermission, ReflectPermission, RuntimePermission, SecurityPermission, SerializablePermission, SQLPermission, SSLPermission, SubjectDelegationPermission, WebServicePermission

---

```
public abstract class BasicPermission
extends Permission
implements Serializable
```

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.

The name for a BasicPermission is the name of the given permission (for example, "exit", "setFactory", "print.queueJob", etc). The naming convention follows the hierarchical property naming convention. An asterisk may appear by itself, or if immediately preceded by a "." may appear at the end of the name, to signify a wildcard match. For example, "*" and "java.*" are valid, while "*java", "a*b", and "java*" are not valid.

The action string (inherited from Permission) is unused. Thus, BasicPermission is commonly used as the base class for "named" permissions (ones that contain a name but no actions list; you either have the named permission or you don't.) Subclasses may implement actions on top of BasicPermission, if desired.

See Also:

`Permission`, `Permissions`, `PermissionCollection`, `SecurityManager`, Serialized Form

Constructor Summary

Constructors

Constructor and Description
`BasicPermission(String name)` Creates a new BasicPermission with the specified name.
`BasicPermission(String name, String actions)` Creates a new BasicPermission object with the specified name.

Method Summary

Methods

Modifier and Type	Method and Description
`boolean`	`equals(Object obj)` Checks two BasicPermission objects for equality.
`String`	`getActions()` Returns the canonical string representation of the actions, which currently is the empty string "", since there are no actions for a BasicPermission.
`int`	`hashCode()` Returns the hash code value for this object.
`boolean`	`implies(Permission p)` Checks if the specified permission is "implied" by this object.
`PermissionCollection`	`newPermissionCollection()` Returns a new PermissionCollection object for storing BasicPermission objects.

Methods inherited from class java.security.Permission

`checkGuard`, `getName`, `toString`

Methods inherited from class java.lang.Object

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

BasicPermission

```
public BasicPermission(String name)
```

Creates a new BasicPermission with the specified name. Name is the symbolic name of the permission, such as "setFactory", "print.queueJob", or "topLevelWindow", etc.

Parameters:

`name` - the name of the BasicPermission.

Throws:

`NullPointerException` - if `name` is `null`.

`IllegalArgumentException` - if `name` is empty.

BasicPermission

```
public BasicPermission(String name,
               String actions)
```

Creates a new BasicPermission object with the specified name. The name is the symbolic name of the BasicPermission, and the actions String is currently unused.

Parameters:

`name` - the name of the BasicPermission.

`actions` - ignored.

Throws:

`NullPointerException` - if `name` is `null`.

`IllegalArgumentException` - if `name` is empty.

Method Detail

implies

```
public boolean implies(Permission p)
```

Checks if the specified permission is "implied" by this object.

More specifically, this method returns true if:

- *p*'s class is the same as this object's class, and
- *p*'s name equals or (in the case of wildcards) is implied by this object's name. For example, "a.b.*" implies "a.b.c".

Specified by:

`implies` in class `Permission`

Parameters:

`p` - the permission to check against.

Returns:

true if the passed permission is equal to or implied by this permission, false otherwise.

equals

```
public boolean equals(Object obj)
```

Checks two BasicPermission objects for equality. Checks that *obj*'s class is the same as this object's class and has the same name as this object.

Specified by:

`equals` in class `Permission`

Parameters:

`obj` - the object we are testing for equality with this object.

Returns:

true if *obj*'s class is the same as this object's class and has the same name as this BasicPermission object, false otherwise.

See Also:

`Object.hashCode()`, `HashMap`

hashCode

```
public int hashCode()
```

Returns the hash code value for this object. The hash code used is the hash code of the name, that is, `getName().hashCode()`, where `getName` is from the Permission superclass.

Specified by:

`hashCode` in class `Permission`

Returns:

a hash code value for this object.

See Also:

`Object.equals(java.lang.Object)`, `System.identityHashCode(java.lang.Object)`

getActions

```
public String getActions()
```

Returns the canonical string representation of the actions, which currently is the empty string "", since there are no actions for a BasicPermission.

Specified by:

`getActions` in class `Permission`

Returns:

the empty string "".

newPermissionCollection

```
public PermissionCollection newPermissionCollection()
```

Returns a new PermissionCollection object for storing BasicPermission objects.

BasicPermission objects must be stored in a manner that allows them to be inserted in any order, but that also enables the PermissionCollection `implies` method to be implemented in an efficient (and consistent) manner.

Overrides:

`newPermissionCollection` in class `Permission`

Returns:

a new PermissionCollection object suitable for storing BasicPermissions.

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.