

## edu.mit.csail.aeolus.api

# Class AeolusTag

java.lang.Object

└─ edu.mit.csail.aeolus.api.AeolusTag

### All Implemented Interfaces:

AeolusSafe, java.io.Serializable, java.lang.Cloneable

---

```
public final class AeolusTag extends java.lang.Object implements java.io.Serializable, AeolusSafe, java.lang.Cloneable
```

This class provides AeolusTags.

### See Also:

[Serialized Form](#)

---

<b>Constructor Summary</b>	
<b>AeolusTag</b> ( )	
Create a new top-level tag.	
<b>AeolusTag</b> (AeolusTag t)	
Create a sub-tag of tag t.	

## Method Summary

AeolusTag	<b>clone</b> ( ) Returns a new AeolusTag which will be equal to this.
void	<b>delegate</b> (PID p1, PID p2) Assuming that the current thread acts for principal p1, and p2 isn't the public pid, and p1 is a node in the delegation graph of this, the delegate method gives authority for this to grantee p2.
boolean	<b>equals</b> (AeolusTag t) Returns true if this is the same as tag t, otherwise false.
boolean	<b>hasAuthority</b> ( ) Returns true if current thread has authority for the tag, otherwise false.
boolean	<b>hasAuthority</b> (PID p) Returns true if principal p has authority for the tag, otherwise false.
boolean	<b>isSubTag</b> (AeolusTag t) Returns true if t is a subTag of this, otherwise false.
boolean	<b>isSuperTag</b> (AeolusTag t) Returns true if t is a superTag of this, otherwise false.
void	<b>revoke</b> (PID p1, PID p2) Revokes authority for this tag from grantor principal from p1 to grantee principal to p2.
java.lang.String	<b>toString</b> ( ) Returns a string representation of this.

## Methods inherited from class java.lang.Object

`equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

### AeolusTag

```
public AeolusTag()  
    throws AeolusException,  
           InfoFlowControlException,  
           AuthorityException
```

Create a new top-level tag. Caller principal will have authority for the new tag.

**Throws:**

`AuthorityException` - - if caller principal not authorized to create a new tag (i.e. if the caller is the public principal)  
`AeolusException` - - if failure in communicating with the authority server or an unexpected platform failure occurred  
`InfoFlowControlException` - - if caller's secrecy label is not null

---

## AeolusTag

```
public AeolusTag(AeolusTag t)
    throws AeolusException,
           InfoFlowControlException,
           AuthorityException
```

Create a sub-tag of tag t. Tag t must be a top-level tag. Caller principal will have authority for the new sub-tag.

**Throws:**

`AuthorityException` - - if caller principal not authorized to create a new sub-tag (i.e. if the caller is the public principal or t is not a top-level tag.)  
`AeolusException` - - if caller is inside a shared state object  
`InfoFlowControlException` - - if caller's secrecy label is not null

## Method Detail

### clone

```
public AeolusTag clone()
```

Returns a new `AeolusTag` which will be equal to this.

**Overrides:**

`clone` in class `java.lang.Object`

---

### delegate

```
public void delegate(PID p1,
                    PID p2)
    throws AeolusException,
           InfoFlowControlException,
           AeolusException
```

Assuming that the current thread acts for principal p1, and p2 isn't the public pid, and p1 is a node in the delegation graph of this, the `delegate` method gives authority for this to grantee p2. The principal p1 must have authority for the tag. The caller's secrecy label must be null.

**Parameters:**

p1 - principal that is authoritative for tag  
p2 - principal that will be authoritative for tag

**Throws:**

`AuthorityException` - - if thread does not act for p1 or p1 does not have authority for this or p2 is the public pid  
`InfoFlowControlException` - - if caller's secrecy label isn't null  
`AeolusException` - if failure in communicating with the authority server or an unexpected platform failure occurred or p1 isn't a node in the delegation graph for this or adding the link would cause a cycle in the graph

---

**equals**

```
public boolean equals(AeolusTag t)
```

Returns true if this is the same as tag t, otherwise false.

**Parameters:**

t - - an AeolusTag

---

**hasAuthority**

```
public boolean hasAuthority()
```

Returns true if current thread has authority for the tag, otherwise false.

---

**hasAuthority**

```
public boolean hasAuthority(PID p)
```

Returns true if principal p has authority for the tag, otherwise false.

**Parameters:**

p - principal with respect to which authority checking is requested

---

**isSubTag**

```
public boolean isSubTag(AeolusTag t)
```

Returns true if t is a subTag of this, otherwise false.

**Parameters:**

t - possible child of this

---

**isSuperTag**

```
public boolean isSuperTag(AeolusTag t)
```

Returns true if t is a superTag of this, otherwise false.

**Parameters:**

t - possible parent of this

---

**revoke**

```
public void revoke(PID p1,  
                   PID p2)  
    throws AeolusException,  
           AuthorityException
```

Revokes authority for this tag from grantor principal from p1 to grantee principal to p2. Current thread must act for the grantor principal p1 and the tag t should have been previously delegated from p1 to p2. Secrecy label must be null.

**Parameters:**

p1 - principal that is authoritative for tag and granted authority to p2 for tag  
p2 - principal to remove from being authoritative for tag

**Throws:**

AuthorityException - - if caller principal does not act for p1 or if p1 isn't a node in the delegation graph of this  
InfoFlowControlException - - if caller's secrecy label is not null an unexpected platform failure occurred  
AeolusException - - if failure in communicating with the authority server or

---

**toString**

```
public java.lang.String toString()
```

Returns a string representation of this.

**Overrides:**

toString in class java.lang.Object

---

**Overview Package Class Tree Index Help**

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY: NESTED | FIELD | CONSTR | METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

[DETAIL: FIELD | CONSTR | METHOD](#)

---