

```
1  /*
2  * Licensed to the Apache Software Foundation (ASF) under one or more
3  * contributor license agreements. See the NOTICE file distributed
4  * with
5  * this work for additional information regarding copyright ownership.
6  * The ASF licenses this file to You under the Apache License, Version
7  * 2.0
8  * (the "License"); you may not use this file except in compliance with
9  * the License. You may obtain a copy of the License at
10 *
11 * http://www.apache.org/licenses/LICENSE-2.0
12 *
13 * Unless required by applicable law or agreed to in writing, software
14 * distributed under the License is distributed on an "AS IS" BASIS,
15 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
16 * implied.
17 * See the License for the specific language governing permissions and
18 * limitations under the License.
19 */
20 package org.apache.harmony.security.tests.support.acl;
21
22 import java.security.Principal;
23 import java.security.acl.*;
24 import java.util.Enumeration;
25 import java.util.Vector;
26
27 /**
28 * Additional class for verification AclEntry interface
29 */
30 public class AclEntryImpl implements AclEntry {
31
32     private Principal user;
33     private Vector permissionSet;
34     private boolean negative;
35
36     public AclEntryImpl(Principal principal) {
37         user = null;
38         permissionSet = new Vector(10, 10);
39         negative = false;
40         user = principal;
41     }
42
43     public AclEntryImpl() {
44         user = null;
45         permissionSet = new Vector(10, 10);
46         negative = false;
47     }
48
49     public boolean setPrincipal(Principal principal) {
50         if(user != null) {
51             return false;
52         } else {
53             user = principal;
54             return true;
55         }
56     }
57
58     public void setNegativePermissions() {
59         negative = true;
60     }
61 }
```



```
58     }
59
60     public boolean isNegative() {
61         return negative;
62     }
63
64     public boolean addPermission(Permission permission) {
65         if(permissionSet.contains(permission)) {
66             return false;
67         } else {
68             permissionSet.addElement(permission);
69             return true;
70         }
71     }
72
73     public boolean removePermission(Permission permission) {
74         return permissionSet.removeElement(permission);
75     }
76
77     public boolean checkPermission(Permission permission) {
78         return permissionSet.contains(permission);
79     }
80
81     public Enumeration permissions() {
82         return permissionSet.elements();
83     }
84
85     public String toString() {
86         StringBuffer stringbuffer = new StringBuffer();
87         if(negative)
88             stringbuffer.append("-");
89         else
90             stringbuffer.append("+");
91         if(user instanceof Group)
92             stringbuffer.append("Group.");
93         else
94             stringbuffer.append("User.");
95         stringbuffer.append((new
96         StringBuilder()).append(user).append("=").toString());
97         Enumeration enumeration = permissions();
98         do {
99             if(!enumeration.hasMoreElements())
100                 break;
101                 Permission permission =
102                 (Permission)enumeration.nextElement();
103                 stringbuffer.append(permission);
104                 if(enumeration.hasMoreElements())
105                     stringbuffer.append(",");
106             } while(true);
107         return new String(stringbuffer);
108     }
109
110     public synchronized Object clone() {
111         AclEntryImpl aclentryimpl = new AclEntryImpl(user);
112         aclentryimpl.permissionSet = (Vector)permissionSet.clone();
113         aclentryimpl.negative = negative;
114         return aclentryimpl;
115     }
116
117     public Principal getPrincipal() {
```

G:\4 - Froyo22 (10\_06\_27)\dalvik\libcore\support\src\test\java\org\apache\harmon  
y\security\tests\support\acl\AclEntryImpl.java

---

```
116         return user;  
117     }  
118 }
```