

PreferencesDesignNotes

Copyright 2004 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

- Preferences design considerations.

java.util.prefs.Preferences implementation.

The preferences implementation has been designed to be able to persist the
following preferences structure:

```
<preferences EXTERNAL_XML_VERSION="1.0">
<root type="user">
<map />
  <node name="user">
    <map />
      <node name="principal1">
        <map />
          <node name="propertyset1">
            <map>
              <entry key="set1prop1" value="256" />
              <entry key="set1prop2" value="314" />
              <entry key="set1prop3" value="540" />
              <entry key="set1prop4" value="63" />
            </map>
          </node>
          <node name="propertyset2">
            <map>
              <entry key="set2prop1" value="256" />
              <entry key="set2prop2" value="314" />
              <entry key="set2prop3" value="540" />
              <entry key="set2prop4" value="63" />
            </map>
          </node>
        </map>
      </node>
    </map>
  </node>
</map>
</root>
</preferences>
```

```

</node>
<node name="principal2">
<map />
  <node name="propertyset1">
  <map>
    <entry key="set1prop1" value="256" />
    <entry key="set1prop2" value="314" />
    <entry key="set1prop3" value="540" />
    <entry key="set1prop4" value="63" />
  </map>
</node>
  <node name="propertyset2">
  <map>
    <entry key="set2prop1" value="256" />
    <entry key="set2prop2" value="314" />
    <entry key="set2prop3" value="540" />
    <entry key="set2prop4" value="63" />
  </map>
</node>
</node>
-----
etc.
-----
</node>

<node name="group">
<map />
  <node name="groupid1">
  <map />
    <node name="propertyset1">
    <map>
      <entry key="set1prop1" value="256" />
      <entry key="set1prop2" value="314" />
      <entry key="set1prop3" value="540" />
      <entry key="set1prop4" value="63" />
    </map>
    </node>
    <node name="propertyset2">
    <map>
      <entry key="set2prop1" value="256" />
      <entry key="set2prop2" value="314" />
      <entry key="set2prop3" value="540" />
      <entry key="set2prop4" value="63" />
    </map>
    </node>
  </node>
  <node name="groupid1">
  <map />
    <node name="propertyset1">
    <map>
      <entry key="set1prop1" value="256" />
      <entry key="set1prop2" value="314" />
      <entry key="set1prop3" value="540" />
      <entry key="set1prop4" value="63" />
    </map>
  </node>

```

PreferencesDesignNotes

```
        </map>
        </node>
        <node name="propertyset2">
        <map>
                <entry key="set2prop1" value="256" />
                <entry key="set2prop2" value="314" />
                <entry key="set2prop3" value="540" />
                <entry key="set2prop4" value="63" />
        </map>
        </node>
    </node>
    -----
    etc.
    -----
</node>
</root>
</preferences>
```

The idea is to be able to persist group or user principals properties where properties are logically grouped into property sets.

Therefore, we could have preferences nodes of type user such as:

1. /user/userprincipal1/propertyset1 where propertyset1 contains properties according to a property set definition.
2. /group/groupprincipal1/propertyset2 where propertyset1 contains properties according to a property set definition.

Good preferences article:

<http://www.onjava.com/pub/a/onjava/synd/2001/10/17/j2se.html>

- Implementation notes.

- Added the following method to the OJB persistence plugin:

```
In org.apache.jetspeed.services.persisistence
LookUpCriteria: add(String, java.util.Collection)
PersistencePlugin: deleteByQuery(java.lang.Object)
```

```
In org.apache.jetspeed.services.persisistence.impl
OjbLookUpCriteria: add(String, java.util.Collection)
AbstractOJBPersistencePlugin: deleteByQuery(java.lang.Object)
```

The test were modified to test deleteByQuery.

- Documentation and javadoc can be found under design-docs and docs.
- Implemented java.util.prefs.Preferences spi.
- Updated to checkstyle 3.0.

