Java XML ACLs

Java XML ACLs

This page documents version 1 of Qpid ACLs that was implemented only in the Java broker and supported up to (and including) release 0.10. Newer releases support ACL V2 only.

- Java XML ACLs
  - Specification
    - XML Format
    - User Guide (SimpleXML)
    - Permission Limitations
    - Enabling XML ACLs
    - ACL Configuration
      - Background
      - ACCESS_CONTROL_LIST Section
      - PUBLISH Section
      - CONSUME Section
      - CREATE Section
      - ACCESS Section (since Qpid 0.6)
    - Durable topic subscriptions
    - Known Issues
      - Granting temporary queue and named queue consume rights

Specification

The XML ACL focus was to take to business style focus to access rather than the individual AMQP method level. As a result we have the following permissions:

- CONSUME
- PUBLISH
- CREATE
- ACCESS
- BIND
- UNBIND
- DELETE
- PURGE

XML Format

DTD TBC

User Guide (SimpleXML)

The XML ACLs have been implemented as per the ACLPlugin design, SimpleXML. Currently this class is only configurable via the main broker configuration file, this means that all the ACL configuration must be included in the main configuration file.

Permission Limitations

Only the first four permissions, CONSUME, PUBLISH, CREATE and ACCESS (since Qpid 0.6) have been implemented. An oversight in the original design resulted in the inability to specify negative permissions. As a result permission can only be granted to users and not taken away.

Enabling XML ACLs

To enable the ACLs the security access class in the main broker configuration needs to be updated as follows:

```xml
...<security>
  <access>
    <class>org.apache.qpid.server.security.access.plugins.SimpleXML</class>
  </access>
...```

This tells the broker that it should use the SimpleXML class to perform access control. When the broker starts up the SimpleXML class will look in the the `<security>` subsection for each virtualhost for the required ACLs.
ACL Configuration

Background

The configuration is described in reference to an example configuration used in a request/response application. In this example the 'client' creates a temporary queue and sends a request to a known queue which the 'server' application is processing. The 'server' then sends a response to the specified temporary queue which the 'client' can read. The ACLs have been configured such that the 'server' cannot create additional queues other than it's process queue and the 'client' is only allowed to create temporary queues.

ACCESS_CONTROL_LIST Section

The ACL configuration lives inside the <access_control_list> section, inside the <security> subsection of each virtualhost configuration.

```
...<security>
    <access_control_list>
        <!-- This section grants virtualhost-level access to the specified users, giving giving them full permissions to all artifacts in the containing virtualhost -->
        <access>...</access>

        <!-- This section grants publish rights to an exchange + routing key pair -->
        <publish>...</publish>

        <!-- This section grants users the ability to consume from the broker -->
        <consume>...</consume>

        <!-- This section grants clients the ability to create queues and exchanges -->
        <create>...</create>
    </access_control_list>
...
```

This gives the basic structure for the configuration the contents of each section naturally depend on what permissions are needed.

PUBLISH Section

This section allows the granting of permission for Publishers to send messages. Controls have been implemented to allow the publication of messages limited by Exchange to:

- specified routing keys.
- partial matching routing keys. Using * to match the end of a routing key.

Here the 'client' users is only give rights to PUBLISH messages using the key 'example.RequestQueue'. The 'server' user is allowed to publish to 'tmp_' and 'TempQueue' keys. The reason there are two values here is due to changes in the naming of temporary queues during the example's development. However, what occurs here is that the 'server' is granted permission to publish messages to any routing key that begins with 'tmp_' or 'TempQueue', the * matching is only completed at the end of the key so entries such as 'Special*Key' are not allowed.

Whilst not shown here multiple <user> values can be specified in the <users> section.

Remember that the routing_key value in the Java broker is the same as the queue name (correct at release of M4) for the amq.direct exchange. For topic exchanges the routing_key is the topic name that a Publisher uses to send messages.
<publish>
  <exchanges>
    <exchange>
      <!-- This is the name of the exchange to limit publication to. -->
      <name>amq.direct</name>
      <routing_keys>
        <!-- Allow clients to publish requests -->
        <routing_key>
          <value>example.RequestQueue</value>
          <users>
            <user>client</user>
          </users>
        </routing_key>
        <!-- Allow the processor to respond to a client on their Temporary Topic -->
        <routing_key>
          <value>tmp_*</value>
          <users>
            <user>server</user>
          </users>
        </routing_key>
        <routing_key>
          <value>TempQueue*</value>
          <users>
            <user>server</user>
          </users>
        </routing_key>
      </routing_keys>
    </exchange>
  </exchanges>
</publish>

CONSUME Section

This section allows the granting of permissions to Consumers. There are two formats the <queue> entry can take:

- Users can be granted permission to a named queue by the use of the <name> field.
- Users can be granted permission to ALL temporary queues with the addition of the <temporary> key.

These two formats can be combined to allow the consumption from a named queue and temporary queues. However, care must be taken if using multiple <queue> entries as access to temporary queues will be defined by the last <queue> definition. This is a known issue.

<consume>
  <queues>
    <!-- Allow the clients to consume from their temporary queues-->
    <queue>
      <temporary/>
      <users>
        <user>client</user>
      </users>
    </queue>
    <!-- Only allow the server to consume from the Request Queue--> 
    <queue>
      <name>example.RequestQueue</name>
      <users>
        <user>server</user>
      </users>
    </queue>
  </queues>
</consume>
CREATE Section

This section allows the granting of permissions to create new queues as used by Consumers. When a consumer is created it makes a request to create the queue for the consumer. This means that all your consumers must also be allowed to create the queue they are going to consume from or they will fail to create.

The <create> section contains a number of fields that can be present:

```xml
<create>
  <queues>
    <!-- Allow clients to create temporary queues-->
    <queue>
      <temporary/>
      <exchanges>
        <exchange>
          <name>amq.direct</name>
          <users>
            <user>client</user>
          </users>
        </exchange>
      </exchanges>
    </queue>
    <!-- Allow the server to create the Request Queue-->
    <queue>
      <name>example.RequestQueue</name>
      <users>
        <user>server</user>
      </users>
    </queue>
  </queues>
</create>
```

The first three behave as in <consume> limiting the list of users to a named queue or all temporary queues. The additional <exchanges> element contains a number of <exchange> entries, this entry contains a list of users that are limited to using only that exchange for the given queue. This is used in the example below to limit the user 'client' to only be able to create temporary queues on the 'amq.direct' exchange.

**NOTE:** This section also suffers from the same issue as <consume> with regard to the <temporary/> keyword. See the known issue for more details.

ACCESS Section (since Qpid 0.6)

This section allows granting virtualhost-level access permissions to specific users, giving them full permissions to all artifacts within the virtualhost irrespective of any rights assigned in the CREATE, CONSUME, and PUBLISH sections outlined above. This allows granting only certain users full access to certain virtualhosts.

The <access> section contains a <users> subsection, with a list of individual <user> elements:

```xml
<!-- This section grants virtualhost-level access to the specified users, giving
  giving them full permissions to all artifacts in the containing virtualhost -->
<access>
  <users>
    <user>admin</user>
  </users>
</access>
```

Durable topic subscriptions

Qpid implements durable topic subscriptions as a persistent queue bound to the topic exchange. This queue is named <clientid>:<subscriptionname>. To allow a JMS durable topic subscription it's necessary to allow queue creation and consumption for the user, eg:

```xml
<!-- This section provides durable subscriptions for the specified users
     and exchanges-->
<create>
  <queues>
    <!-- Allow durable subscriptions for the following users
        on the exchange 'amq.direct':
        <user>client1</user>
        <user>client2</user> -->
  </queues>
</create>
```
Known Issues

Granting temporary queue and named queue consume rights

When defining a `<queue>` entry the existence of the `<temporary>` key grants access to temporary queues. However, the lack of the key denies access to temporary queues. As a result if there are multiple `<queue>` entries the last entry will specify the value for access to temporary queues. i.e. In this example it is expected that ‘client’ can consume from temporary queues and named queue ‘exampleQueue2’. Infact what will happen is that the user will only have access to ‘exampleQueue2’.

To work around this issue the correct definition would be: