Configuring a JMS Event Filter

JMS events are emitted for repository updates, and the machinery for processing the events is defined in eventing.xml.

By default, internal events are filtered by an instance of DefaultFilter that only filters out extraneous events about non-repository resources. There is currently one other filter available, SuppressByMixinFilter, which will filter out events on nodes with the specified mixin types. To configure SuppressByMixinFilter, update eventing.xml to replace the DefaultFilter configuration:

```
<bean name="fedoraEventFilter" class="org.fcrepo.kernel.impl.observer.DefaultFilter"/>
```

with the SuppressByMixinFilter configuration:

```
<!-- suppress events for resources with any of these mixins -->
<util:set id="suppressedMixins">
  <value>ldp:DirectContainer</value>
</util:set>

<!-- used by bean above to filter which events get put on the bus -->
<bean name="fedoraEventFilter" class="org.fcrepo.kernel.impl.observer.SuppressByMixinFilter"
    c:suppressedMixins-ref="suppressedMixins"/>
```

The `<value>` element can be repeated as many times as needed to list all of the mixins that should be suppressed. A full example with all of the required Spring namespaces declared can be found in the test configuration: eventing-suppress.xml.

Custom Event Filters

For more flexibility, you can create your own EventFilter implementation and configure it in the same way. Implementations need to implement the following methods:

- `getFilter(javax.jcr.Session)`: used to instantiate the filter
- `apply(javax.jcr.observation.Event)`: used to filter events, returning true if an event should be processed and false if it should be ignored