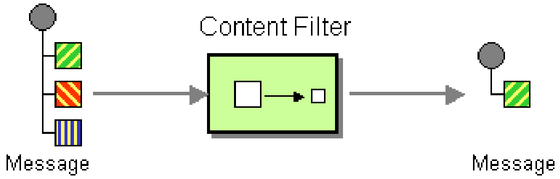
Content Filter

Overview

The *content filter* pattern describes a scenario where you need to filter out extraneous content from a message before delivering it to its intended recipient. For example, you might employ a content filter to strip out confidential information from a message.

Figure 24. Content Filter Pattern



A common way to filter messages is to use an expression in the DSL, written in one of the supported scripting languages (for example, XSLT, XQuery or JoSQL).

Implementing a content filter

A content filter is essentially an application of a message processing technique for a particular purpose. To implement a content filter, you can employ any of the following message processing techniques:

- *Message translator*—see message translators on page 34.
- *Processors*—see *Implementing a Processor* in the *Programmer's Guide*.
- Bean integration [http://activemq.apache.org/camel/bean-integration.html]. *REVISIT - NB: Need to replace this with a link to FUSE docs, once bean integration is properly documented.*

Java DSL example

If the message body is encoded as XML, one way of filtering the message content would be to use an XSLT script. For example, if the XSLT script, `content_filter.xsl`, is located in the `com/acme` directory on your CLASSPATH, you could invoke it to process incoming messages as follows:

```
from("activemq:My.Queue")
  .to("xslt:classpath:com/acme/content_filter.xsl")
  .to("activemq:Another.Queue");
```

Where the message is pulled from the `My.Queue` JMS queue, then filtered by the XSLT endpoint, `xslt:classpath:com/acme/content_filter.xsl`, and finally sent to the `Another.Queue` JMS queue.

For more details about the XSLT component, see XSLT in the *Component Reference*.

XML configuration example

The following example shows how to configure the same route in XML:

```
<camelContext xmlns="http://act
ivemq.apache.org/camel/schema/spring">
  <route>
    <from uri="activemq:My.Queue"/>
    <to uri="xslt:classpath:com/acme/content_filter.xsl"/>
    <to uri="activemq:Another.Queue"/>
  </route>
</camelContext>
```