

定制拦截器可以部署在 *plugins.d* 目录，如在第 8 章“部署自定义代码”一节的说明。

Channel 选择器

Channel 选择器是决定 Source 接收的一个特定事件写入哪些 Channel 的组件。它们告知 Channel 处理器，然后由其将事件写入到每个 Channel。

由于 Flume 并不是两阶段提交，事件被写入到一个 Channel，然后在事件被写入到下一个 Channel 之前提交。如果写入一个 Channel 时出现故障，可能已经发生在其他 Channel 的相同事件的写入不能被回滚。当这样的故障发生时，Channel 处理器抛出 `ChannelException` 并且事务失败。如果 Source 试图再次写入相同的事件（在大多数情况下，它会；只有类似 Syslog、Exec 等 Source 不能重试，因为没有办法再次生成相同的数据），重复的事件将写入到 Channel，而先前的提交实际上是成功的。这是在 Flume 管道中发生重复的一种方式。

Channel 选择器配置是通过 Channel 处理器完成的，虽然配置看起来像 Source 子组件的配置。传递到 Channel 选择器的所有参数作为 Source 的上下文中的参数使用 selector 后缀传递。对于每个 Source，选择器通过使用一个配置参数 `type` 指定。Channel 选择器可以指定一组 Channel 是必需的（`required`），另一组是可选的（`optional`），如第 2 章的描述。所有 Channel 选择器的一个公共参数在表 6-7 中有描述。

151

表6-7 Channel选择器配置

| 配置参数 | 默认值 | 描述 |
|-------------------|--------------------------|----------------------------------|
| <code>type</code> | <code>replicating</code> | 用于 Source 的 Channel 选择器的别名或 FQCN |

Flume 内置两种 Channel 选择器：`replicating` 和 `multiplexing`。如果 Source 的配置中没有指定选择器，那么会自动使用复制 Channel 选择器。一个 Channel 选择器配置例子如下所示：

```
agent.sources.avroSrc.selector.type = multiplexing
agent.sources.avroSrc.selector.header = priority
agent.sources.avroSrc.selector.mapping.1 = channel1
agent.sources.avroSrc.selector.mapping.2 = channel2
agent.sources.avroSrc.selector.default = channel2
```

复制 Channel 选择器

如果 Source 没有指定 Channel 选择器，则该 Source 使用复制 Channel 选择器。该复制 Channel 选择器复制每个事件到通过 Source 的 `channels` 参数所指定的所有的 Channel 中。