

# UuidGenerator

## UuidGenerator

Starting with **Camel 2.5**, Camel supports 3rd party UUID generator(s). This is useful, if e.g. your messaging provider does not support UUID's with a length of 36 characters (like Websphere MQ). Another useful scenario is to use a simple counter for testing purpose. With this it is easier to correlate the exchanges in the log/debugger.

Camel uses UUIDs in the exchange and message ids, and other unique ids it uses.

You only have to implement `org.apache.camel.spi.UuidGenerator` and tell Camel, that it should use your custom implementation:

## Configuring from Java DSL

```
getContext().setUuidGenerator(new MyCustomUuidGenerator());
```



You should not change the UUID generator at runtime (it should only be set once)!

## Configuring from Spring DSL

Camel will configure this UUID generator by doing a lookup in the Spring bean registry to find the bean of the type `org.apache.camel.spi.UuidGenerator`.

```
<bean id="activeMQUuidGenerator" class="org.apache.camel.impl.ActiveMQUuidGenerator" />

<camelContext id="camel" xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="direct:start" />
    <to uri="mock:result" />
  </route>
</camelContext>
```

## Provided implementations

Camel comes with three implementations of `org.apache.camel.spi.UuidGenerator`:

- `org.apache.camel.impl.JavaUuidGenerator` - This implementation uses `java.util.UUID`. The `java.util.UUID` is synchronized and can therefore affect performance on high concurrent systems. Therefore consider one of the generators.
- `org.apache.camel.impl.SimpleUuidGenerator` - This implementation use internally a `java.util.concurrent.atomic.AtomicLong` and increase the ID for every call by one. Starting with 1 as the first id.
- `org.apache.camel.impl.ActiveMQUuidGenerator` - This implementation use the ActiveMQ style of ID's. This implementation may use some APIs from the JDK which is forbidden to use if running in the cloud (such as Google App Engine) and therefore you may have to use one of the other generators.

## ActiveMQUuidGenerator

From **Camel 2.10.7/2.11.1** onwards the JVM system property with key: `activemq.idgenerator.port` can be used to assign a specific port which is used during initialization of the UUID generator. By default the port number 0 is used. Though in some cloud infrastructures this is not allowed, and thus a specific port can be assigned instead.

## The default generator

From Camel 2.5 onwards the `ActiveMQUuidGenerator` is the default generator because its the fastest.

In Camel 2.4 or older the default is the `JavaUuidGenerator` generator.

In Camel 1.x the default generator is likewise the `ActiveMQUuidGenerator`.