Graceful Shutdown

Graceful Shutdown

Available as of Camel 2.2

Camel now supports a pluggable shutdown strategy using `org.apache.camel.spi.ShutdownStrategy`. Its responsible for shutting down routes in a graceful manner. The other resources will still be handled by CamelContext to shutdown. This leaves the problem at hand with properly shutting down all the routes in a reliable manner to the `ShutdownStrategy`.

Camel provides a default strategy in the `org.apache.camel.impl.DefaultShutdownStrategy` which is capable of doing that.

DefaultShutdownStrategy

The default strategy will gracefully shutdown routes:

- **Camel 2.2:** in the same order they was started
- **Camel 2.3:** in the reverse order they was started. The option `shutdownRoutesInReverseOrder` can be used to use the old behavior.
- let pending and current in flight exchanges run to completion before shutting down
- using a timeout of 300 seconds which then forces a shutdown now

You can configure the timeout, and whether it should shutdown now remaining routes when the timeout occurred or ignore. See the setters on the class.

It will output to log the progress during graceful shutdown as shown in an example below

```
2009-12-20 10:56:53,055 [main ] INFO DefaultCamelContext - Apache Camel (CamelContext:camel-1) is stopping 2009-12-20 10:56:53,056 [main ] INFO
DefaultShutdownStrategy - Starting to graceful shutdown routes (timeout 300 seconds) 2009-12-20 10:56:53,059 [1: ShutdownTask] INFO
DefaultShutdownStrategy - Waiting as there are still 5 inflight exchanges to complete before we can shutdown 2009-12-20 10:56:54,060 [1:
ShutdownTask] INFO DefaultShutdownStrategy - Waiting as there are still 4 inflight exchanges to complete before we can shutdown 2009-12-20 10:56:
55,061 [1: ShutdownTask] INFO DefaultShutdownStrategy - Waiting as there are still 3 inflight exchanges to complete before we can shutdown 2009-12-
20 10:56:56,065 [1: ShutdownTask] INFO DefaultShutdownStrategy - Waiting as there are still 2 inflight exchanges to complete before we can shutdown
2009-12-20 10:56:57,066 [1: ShutdownTask] INFO DefaultShutdownStrategy - Waiting as there are still 1 inflight exchanges to complete before we can
shutdown 2009-12-20 10:56:58,069 [main ] INFO DefaultShutdownStrategy - Graceful shutdown of routes complete in 5 seconds. 2009-12-20 10:56:
58,072 [main ] INFO DefaultInflightRepository - Shutting down with no inflight exchanges. 2009-12-20 10:56:58,077 [main ] INFO DefaultCamelContext -
Apache Camel (CamelContext:camel-1) stopped
```

Notice how it waits while there are inflight exchanges still being processed before it can shutdown.

Suppressing logging due to timeout not allowing all inflight messages to complete

Available as of Camel 2.12

If a graceful shutdown could not shutdown cleanly within the given timeout period, then Camel performs a more aggressive shutdown by forcing routes and thread pools etc to shutdown. And as well the routing engine will reject continue processing Exchanges. If this happens you may see WARN logs about Exchanges being rejected and other failures due the forced shutdown.

If you do not want to see these logs, you can suppress this by setting the option SuppressLoggingOnTimeout to true.

```
context.getShutdownStrategy().setSuppressLoggingOnTimeout(true);
```

Notice the suppress is a "best effort" though there may still be some logs coming from 3rd party libraries and whatnot, which Camel cannot control.

Logging inflight exchange information on timeout

Available as of Camel 2.15

If a graceful shutdown could not shutdown cleanly within the given timeout period, then Camel performs a more aggressive shutdown by forcing routes and thread pools etc to shutdown. When the timeout happens, then Camel logs information about the current inflight exchanges, which shows from which route the exchange origins, and where it currently is being routed. For example the logging below, shows that there is 1 inflight exchange, that origins from route1, and currently is still in route1 at the "delay1" node. The elapsed is time in millis how long at the current node (eg delay1) and duration is total time in mills.

If you enable DEBUG logging level on `org.apache.camel.impl.DefaultShutdownStrategy` then it logs the same inflight exchange information during graceful shutdown

```
2015-01-12 13:23:23,656 [ - ShutdownTask] INFO DefaultShutdownStrategy - There are 1 inflight exchanges: InflightExchange: [exchangeId=ID-
davsclaus-air-62213-1421065401253-0-3, fromRouteId=route1, routeId=route1, nodeId=delay1, elapsed=2007, duration=2017]
```

If you do not want to see these logs, you can turn this off by setting the option logInflightExchangesOnTimeout to false.

```
context.getShutdownStrategy().setLogInflightExchangesOnTimeout(false);
```

Controlling ordering of routes

You can configure the order in which routes should be started, and thus also the same order they are being shutdown.
See more at Configuring route startup ordering and autostartup.

Fine grained configuration

You can control two areas that influence graceful shutdown in the Camel routing:

- `ShutdownRoute`
- `ShutdownRunningTask`

These options can be configured on two scopes: `context` and `route`. Where a route will fallback to the `context` scoped option, if not explicit configured. (same principle as Error Handler, etc.).

ShutdownRoute

This option can control how a given route should act during graceful shutdown. It has two values `Default` and `Defer`. The `Default` is obviously the default option which lets Camel shutdown the route as early as possible. The `Defer` is used to defer shutting down this route to a later stage. This is useful when other routes are dependent upon it. For example an internal route which other routes reuse.

For example in the route below we have two routes, where route 1 is dependent upon route 2. At shutdown we want route 1 to complete all its current messages and we also want the 2nd route to do this as well. So we can mark both routes to `Defer` but since route 1 is a SEDA based route its `Defer` by default (it uses `ShutdownAware`).

A Java DSL based example to defer shutting down the 2nd route:{snippet:id=e1|lang=java|url=camel/trunk/camel-core/src/test/java/org/apache/camel/processor/ShutdownDeferTest.java}The same route in Spring XML would be:{snippet:id=e1|lang=xml|url=camel/trunk/components/camel-spring/src/test/resources/org/apache/camel/spring/processor/ShutdownDeferTest.xml}

Defer shutting down internal routes only
Its best to only defer shutting down internal routes only. As **public** routes should shutdown as quickly as possible otherwise it will just keep intake new messages which will delay the shutdown processor. Or even have it timeout if a lot of new messages keep coming in.

ShutdownRunningTask

This option control how a given route consumer acts during shutdown. Most route consumer will only operate on a single task (message), however the Batch Consumer can operate on many messages (in a batch). This option is for those kind of consumers. By default it uses the option `CompleteCurrentTaskOnly` which mean that the current *in progress* task (message) will be completed and then the consumer will shutdown. The other option `CompleteAllTasks` allows the consumer to complete all the tasks (messages) before shutting down. For example a File consumer will process all the pending files it has picked up before shutting down.

A Java DSL based example to complete all messages during shutting down the first route:{snippet:id=e1|lang=java|url=camel/trunk/camel-core/src/test/java/org/apache/camel/processor/ShutdownCompleteAllTasksTest.java}The same route in Spring XML would be:{snippet:id=e1|lang=xml|url=camel/trunk/components/camel-spring/src/test/resources/org/apache/camel/spring/processor/ShutdownCompleteAllTasksTest.xml}

JMX managed

The `ShutdownStrategy` is JMX aware as well so you can manage it from a JMX console. For example you can change the timeout value.

Shutting down individual routes

Available as of Camel 2.3
Its now possible to gracefully shutdown an individual route using `shutdownRoute(routeId)` method on `CamelContext`. Its also possible to provide a specific timeout to use instead of the default timeout settings using `shutdownRoute(routeId, timeout, timeUnit)`.

Developer related

If you develop your own Camel component or want to implement your own shutdown strategy then read this section for details.

ShutdownStrategy

You can implement your own strategy to control the shutdown by implementing the `org.apache.camel.spi.ShutdownStrategy` and the set it on the `CamelContext` using the `setShutdownStrategy` method.

When using Spring XML you then just define a spring bean which implements the `org.apache.camel.spi.ShutdownStrategy` and Camel will look it up at startup and use it instead of its default. See more at Advanced configuration of CamelContext using Spring.

ShutdownAware

The interface `org.apache.camel.spi.ShutdownAware` is an optional interface consumers can implement to have fine grained control during shutdown. The `ShutdownStrategy` must be able to deal with consumers which implement this interface. This interface was introduced to cater for in memory consumers such as SEDA which potentially have a number of pending messages on its internal in memory queues. What this allows is to let it control the shutdown process to let it complete its pending messages.

The method `getPendingExchangesSize` should return the number of pending messages which reside on the in memory queues.
The method `deferShutdown` should return `true` to defer the shutdown to a later stage, when there are no more pending and inflight messages.

Batch Consumer should implement `ShutdownAware` so they properly support the `ShutdownRunningTask` option. See `GenericFileConsumer` for an example.

See Also

- Configuring route startup ordering and autostartup
- Advanced configuration of CamelContext using Spring
- User Guide