

# Camel 2.4.0 Release

## Camel 2.4.0 release

### New and Noteworthy

Welcome to the 2.4.0 release which approx 182 issues resolved (new features, improvements and bug fixes such as...)

- Spring 3.0.3 is now the default Spring version used by Camel.
- Fully non blocking [Asynchronous Routing Engine](#) which all EIPs and some Camel [Components](#) supported.
- New feature to use scheduled [delayed redelivery attempts](#) (they are fully asynchronous and non blocking). You need to enable this using the option `asyncDelayedRedelivery`.
- Fixed a potential dead-lock when using [Aggregator](#) with timeout completion set.
- The **camel-spring** module now supports OSGi out of box, which renders **camel-osgi** as **@deprecated**.
- [OSGi blueprint](#) is now supported
- Added new `<contextScan>` to scan the [Registry](#) for `RouteBuilder` instances, just as scanning the classpath etc.
- Manually started routes is now also [Graceful Shutdown](#) by Camel
- Fixed issue using [RouteContextRef](#) with multiple [CamelContext](#) from the same Spring `ApplicationContext`.
- [Bindy](#) now supports Fixed-Length format message with padding, alignment (Left or Right)
- Improved and fixes some issues with the [FTP](#) component.
- Improved and fixed some issues using `MultiPartForm` with [Jetty](#).
- Add ability to configure [Bindy](#) to remove whitespace when binding `Numeric` positions
- Added option `useOriginalBody` to [OnCompletion](#)
- Fixed issue when using `mockito` for testing when it uses `CGLib` proxies
- Fixed issue with `@Consume` not running in an unit of work to ensure callbacks is invoked when its done
- [Routing Slip](#) now uses [Expression](#) to define the slip. This is similar to the [Recipient List](#) pattern
- The security context can be passed from [CXF](#) to other camel [Components](#).
- `ManagementAgent` settings can now be set using setters from Java, instead of having to use JVM system properties. See tip in [Camel JMX](#) wiki page.
- [Resequencer](#) in batch mode has two new options `allowDuplicates` and `reverse` which allows you to use it for re-ordering messages from [JMS](#) queues based on `JMSPriority`
- Tighten up the `ProducerTemplate` API a bit, see more details in the section *Important changes to consider when upgrading*.
- [HTTP](#) now better supports sending binary files.
- Fixed problem with using [Quartz](#) in OSGi and having Camel applications being hot-deployed which potentially could cause [Quartz](#) to stop scheduling in applications which haven't been stopped.
- Fixed issue with pausing and resuming jobs when running [Quartz](#) in clustered mode
- Added options on [Quartz](#) component to specify location of custom `quartz.properties` file to load from classpath.
- Upgraded to latest [Quartz](#) release 1.8.3.
- Introduced `StartupListener` to allow components/endpoints/producers/consumers etc. as a listener to be notified when `CamelContext` have just been started. This allows any custom logic to be executed as a late starter. This ensures [Quartz](#) scheduler is only started when all the routes and so forth is started and running.
- Fixed issue with [FTP](#) configuring starting directory from uri not accepting an absolute path. In the endpoint uri just define two leading / to indicate absolute, eg "ftp:admin:secret@somesever//absolutepath/foo/bar".
- Added `fromRoute` to [NotifyBuilder](#).
- Improved the [CSV](#) component.
- Fixed issue with [Direct](#) endpoint having no consumers if endpoint have been evicted from `CamelContext` internal `LRU`Cache.
- The `RouteBuilder` now have `simple` and `xpath` expression builder methods out of the box.
- Upgraded to latest release of `Scala 2.8.0RC7` in [Scala DSL](#)
- `camel-hawtdb` updated to [HawtDB 1.1](#) picks up several bug fixes that improves stability and performance.
- Fixed issue with `splitting files` using `tokenizer` not being able to delete/move file after processing is done on Windows platform.

- `maxMessagesPerPoll` is now applied eagerly to limit intaking files earlier for the [File](#) and [FTP](#) components, Thus avoid excessive memory consumption if you poll directories which contains 100.000+ files.
- Added `connectTimeout`, `soTimeout` and `timeout` to [FTP](#) so its easier to configure timeouts.
- Consolidated support for per platform disabling of tests across the entire test suite.
- Introduced [ProcessorFactory](#) to be able to use a custom factory to be able to manipulate [Processor](#) creation process. For example to add additional outputs or manipulate the route model.
- [camel-jms](#) now supports Oracle AQ
- Added a [Debugger](#) API to be leveraged by 3rd party tooling.
- The `beanRef` and `methodCall` DSLs will eager check on startup, the bean reference exists in [Registry](#) and the provided method name is a valid method name on the bean as well. This helps to fail fast if end users have a typo in their routes.
- [JMX](#) now exposes notifications which listeners can receive.
- Added `ExchangeRedeliveryEvent` event being emitted when Camel [Error Handler](#) attempts a redelivery.
- [Properties](#) component can lookup `java.util.Properties` from the [Registry](#) using the `ref:` prefix.
- [Spring Security](#) is upgrade to spring security 3.0.3.RELEASE.

## New Enterprise Integration Patterns

### New Components

- [EJB](#) to access [EJB](#) beans in a similar way as with the [Bean](#) component
- [Class](#) to invoke beans defined using the fully qualified class name
- [OSGi Blueprint](#) to use Camel with OSGi blueprint

### New DSL

### New Annotations

- `@RoutingSlip`

### New Data Formats

### New Languages

### New Examples

### New Tutorials

## API breaking

The [ToAsync](#) has been removed and replaced with a better [asynchronous routing engine](#).

[Routing Slip](#) now requires using [Expression](#) to specify the routing slip header, the XML DSL of [Routing Slip](#) is broken.

```
<route>
  <from uri="direct:a"/>
  <!--NOTE from Camel 2.4.0, you need to specify the expression element inside of
the routingSlip element -->
  <routingSlip ignoreInvalidEndpoints="true">
    <header>myHeader</header>
  </routingSlip>
</route>
```

The `retryUntil` method on [onException](#) have been renamed to `retryWhile` to better match what it does. It keeps retrying while the predicate returns true.

`org.apache.camel.spi.Policy` now has an additional method `beforeWrap` which means you need to implement this method in your implementations. Just add a empty method to adapt.

The class `org.apache.camel.management.event.ExchangeFailureEvent` has been renamed to `org.apache.camel.management.event.ExchangeFailedEvent`.

## Known Issues

The [Tracer](#) may not output all details for some situations such as when using `onCompletion` or `intercept` etc.

Not all [Examples](#) have ANT build.xml files to run the example using ANT.

The read `timeout` in [Netty](#) is causing it to spawn threads and you end up with out of memory error. You have to disable using timeout by setting the `timeout=0` in the endpoint configuration. This issue has been fixed in the next release.

The [LoggingErrorHandler](#) does not work when also using `onException`. It may go into an endless loop. So refrain from using logging error handler.

The [JMX](#) statistics for `type=processors` is being counted double. The stats on the `type=routes` is correct though.

The [Jetty](#) component sends back HTTP headers with lower-case keys.

[HawtDB](#) does not work in OSGi

[HawtDB](#) has a bug in version 1.4 or older which prevents the filestore to free unused space. Version 1.5 fixes this.

There is a concurrency issue when using predicates using binary operators such as `isEqualTo`. See [CAMEL-3188](#).

The [FTP](#) consumer may stop continue polling. You can start the consumer from eg. a management console and it will start again.

## Important changes to consider when upgrading

Camel now uses Spring 3.0.3 as the default Spring version. Spring 2.5 is still supported. Spring 2.0.x support is now **@deprecated** and will in future releases **not** be supported.

### Spring 3.0 dependency injection changed

Mind that Spring 3.0 has changed how it performs dependency injection. In fact it may even perform this differently than what Spring 2.5 would do. For example the order and which beans is dependency injected on startup could potential affect your existing Camel applications when upgrading. It appears as Spring 3.0 has gotten worse in this area and you may have to specify `depends-on` attributes on your spring bean to help Spring.

[IntrospectionSupport](#) has been improved to be consistent for `getProperty` and `getProperties`. Also `getProperties` now find all relevant properties. This will often only be relevant for Camel end users who build their own components any may use [IntrospectionSupport](#).

The [ProducerTemplate](#) have been tighten up a bit. All `send/request` `-body` methods will now throw a `CamelExecutionException` with the cause wrapped. This ensures its consistent.

The `ftps` is now using secure data channel also when transferring files. There has been added some options which you can use to control the behavior and turn it off if you want the old behavior (only secure login).

The [FTP](#) starting directory can now be set using absolute paths as well. In the endpoint uri just define two leading `/` to indicate absolute, eg `"ftp:admin:secret@someserver//absolute/path/foo/bar"`.

The default port from `ftps` has been changed from 2222 to 21.

The [FTP](#) is now using a default of 10 sec connect timeout (ALL), and 30 sec data timeout (FTP/FTPS only).

The [RouteBuilder](#) now have `simple` and `xpath` expression builder methods out of the box, which means you don't have to use static imports to use those two [Languages](#). The compiler may complain if you have a static import, which you simply fix by removing the static import.

## Getting the Distributions

### Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	<a href="#">apache-camel-2.4.0.zip</a>	<a href="#">apache-camel-2.4.0.zip.asc</a>

Unix/Linux/Cygwin Distribution	<a href="#">apache-camel-2.4.0.tar.gz</a>	<a href="#">apache-camel-2.4.0.tar.gz.asc</a>
--------------------------------	---	---

**The above URLs use redirection**

The above URLs use the Apache Mirror system to redirect you to a suitable mirror for your download. Some users have experienced issues with some versions of browsers (e.g. some Safari browsers). If the download doesn't seem to work for you from the above URL then try using [FireFox](#)

## Source Distributions

Description	Download Link	PGP Signature file of download
Source for Windows	<a href="#">apache-camel-2.4.0-src.zip</a>	<a href="#">apache-camel-2.4.0-src.zip.asc</a>

Source for Unix/Linux/Cygwin	<a href="#">apache-camel-2.4.0-src.tar.gz</a>	<a href="#">apache-camel-2.4.0-src.tar.gz.asc</a>
------------------------------	---	---

## Getting the Binaries using Maven 2

To use this release in your maven project, the proper dependency configuration that you should use in your [Maven POM](#) is:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-core</artifactId>
  <version>2.4.0</version>
</dependency>
```

## SVN Tag Checkout

```
svn co http://svn.apache.org/repos/asf/camel/tags/camel-2.4.0
```

## Changelog

For a more detailed view of new features and bug fixes, see:

- [JIRA Release notes for 2.4.0](#)