Camel 2.6.0 Release

Camel 2.6.0 release

ked URL

New and Noteworthy

Welcome to the 2.6.0 release which approx 297 issues resolved (new features, improvements and bug fixes such as...)

- Fixed issue in OSGi with refreshing Camel bundles causing a race condition when discovering type converters
- Introduced ScheduledRoutePolicy with the Quartz component
- Java DSL RouteBuilder now fails if `onException` and the likes are not configured before routes, which they must.
- Easier to debug routes from `camel-test` using the Debugger
- Fixed problem installing `camel-cxf` feature on Apache Karaf.
- The EIP now supports property placeholders in the `String` based options (a few spots in Java DSL where its not possible). For example:

```
<convertBodyTo type="String" charset="{{foo.myCharset}}"/>
```

- Introduced ScheduledRoutePolicy to allow you to schedule when routes should be active.
- Introduced `stepwise` option to FTP component to allow end users to control how to traverse directories of the FTP servers. This allows you to set `stepwise=false` to return back to the behavior from Camel 2.0 - 2.4. See more at the FTP documentation.
- Added `separator` option to FTP component to allow end users to control what path separator to use when uploading files to remote FTP servers. For example if a Windows based FTP server only accept Windows paths, then you can set this option to `Windows`.
- Thread names now outputs CamelContext name which makes it easier to differentiate when running multiple Camel applications in the same JVM.
- Introduced `contentCache` option to XSLT component to allow easy reloading of stylesheets during development.
- Improved Camel JMX to always find next free name when registering in JMX and existing name exists (to remedy name clash). This makes it easier when deploying muliple WARs in the same JVM where the Camel applications may have same CamelContext id. In OSGi the name in JMX now has the bundle id as prefix.
- Fixed BeanShell language
- Quartz now supports using older versions (eg versions 1.6/1.7) of quartz JARs.
- The Aggregator EIP will use a synchronous invocation when processing completed aggregated Exchanges, when not running in parallel mode. This ensures no internal task queue is used, which otherwise could stack up tasks and under certain conditions eat up memory. Enable `parallelProcessing` if you want to use a separate thread for processing completed aggregated Exchanges.
- LDAP supports paging.
- CXF and CXFRS use the CXF continuation API when using CXF 2.3.x.
- You can access the CXF Message from the Camel message header by using the key "CamelCxfMessage" after the CXF consumer.
- Fixed a rare situation in which an Exchange could be lost during graceful shutdown on the SEDA endpoints, usually more apparent on high volume systems.
- Improved support for Locale in Bindy for Number and Date formats.
- Fixed Multicast, Splitter, Recipient List to work as the Pipes and Filters would do, breaking to continue routing if a sub exchange failed and was handled by an Exception Clause. Previously you would have to detect this manually in the `AggregationStrategy`.
- Constant fields from Exchange can be referred in the DSL from a String attribute using the class type style. This makes the Java DSL and Spring XML more similar. And you got type safe as Camel throws exception if you had a typo.
  For example:

```
<setHeader headerName="Exchange.FILE_NAME"><constant>myfile.txt</constant></setHeader>
```

- Exchange now contains the creation timestamp when it was created as a `java.util.Date` object stores as a property with the key `Exchange.CREATION_TIMESTAMP`
- Camel fails fast on startup if Load Balancer has been misconfigured, such as configuring two different load balancers in one
- Fixed `markRollbackOnlyLast` when used with inner/outer transactions.
- `InterceptStrategy` is now default ordered in the order they are added to the CamelContext. However you can dictate the order by implementing the `org.apache.camel.util.Ordered` interface.
- Made it easier to load routes from XML files using `loadRoutesDefinition` method from CamelContext.
- Added support for custom multipart filters in Jetty.

- Using Processor or Bean in routes is now also enlisted in JMX for management.
- The ObjectName for `processor` in JMX is now always using the node id it has been assigned in the route.
- Using a custom Processor or Bean will be enlisted in JMX having its custom attributes/operations listed. For example just use the Spring JMX annotations to easily expose the attributes/operations. See more details here.
- Fixed some issue with the `camel-blueprint` component.
- Added features to `camel-blueprint` component so its on-par with feature set from Spring XML.
- Fixed FTP producer may creating directory using windows path separators.
- Fixed InterceptSendToEndpoint not honoring `stop` DSL to stop immediately and not sending the Exchange to the intercepted Endpoint.
- Improved the IRC component
- Added `matchsMockWaitTime` method to NotifyBuilder which is a convenient method to use when testing with mocks and want to match using timeout.
- `org.apache.camel.spi.EventNotifier` events for Exchange done is now fired **after** OnCompletion has been executed.
- Added option `deleteOutputFile` to XSLT component.
- Idempotent Consumer's `IdempotentRepository` is now managed in JMX.
- File and FTP producers now support writing *done* files after the target file has been written, using the new `doneFileName` option.
- File and FTP consumers now support only consuming files if a *done* file exists, using the new `doneFileName` option.
- Thread Pools is now named more intuitive in JMX, making it much easier for end user to spot which pool is used for what. And it provides more information where the pool is used.
- Data Format now supports `Service` and `CamelContextAware` callbacks which means its much easier to perform custom initialization logic in your data formats.
- Fixed Printer component to use exact same constants for `mediaSize` option as from the Java Printer API.
- Fixed issue with route scoped Exception Clause may pick Exception Clause defined on another route, if exception is the same type. Also ensures that route scoped is preferred over context scoped.
- Fixed Bindy fixed length to throw exception if field exceeds allowed length. Added option `clip` on `@DataField` to allow Bindy to clip the field so it fits the length, instead of thrown an exception.
- Fixed Bindy when using quoted CSV records and fixed when using pipeline delimiter.
- Added header `Exchange.REDELIVERY_MAX_COUNTER` to message when Camel performs a redelivery attempt. This header contains the maximum redelivery setting. It may be `null` if you use unlimited redelivery attempts or use `retryWhile`.
- JMS producer using `InOnly` MEP now supports `JMSReplyTo` configured on either endpoint, or provided as message header. The message send to the JMS destination will now contain the `JMSReplyTo` information. The JMS producer which send the message will still be `InOnly` and not expect/wait for a reply but continue immediately. You have to have the `preserveMessageQos=true` option defined as well.
- Fixed JMS producer, when a `JMSReplyTo` header was provided as a message header, removing any `queue:` or `topic:` prefix causing the reply to destination to be resolved correctly (eg avoiding having `queue:` in the queue name).
- JMS producer now honors, if reply to was disabled using `disableReplyTo=true` in all situations.
- HTTP, Jetty, SERVLET and `camel-http4` now supports content type `application/x-java-serialized-object` on both the producer and consumer side. This allows you to transfer serialized Java objects over http.
- Added option `transferException` to HTTP, Jetty and SERVLET, which allows you to return back caused exceptions from the consumer side as serialized objects. On the producer side the caused exception will be thrown instead of the generic `HttpOperationFailedException`.
- Web Console now displays some basic route statistics if JMX is enabled.
- JPA `JpaTraceEventMessage` now uses `@Lob` for fields which may contain a lot of data, such as message body.
- Throughput logger now supports logging per interval (timer based).
- Fixed an issue with Splitter causing exception mistakenly from one sub-message being propagated to the next sub-message.
- XQuery now strips all whitespaces by default; there is an option to changes this to ignorable whitepsaces only.
- Quartz now enforces trigger name/group to be unique within the given component. Will throw exception on name clash detected. This behavior does not apply for clustered quartz.
- Fixed Splitter using too much memory when running in parallel mode. Now it aggregates on-the-fly and memory consumption is low and stable.
- Bean now has a type option which you can use if you use method overloading in your Bean and prevent Camel to fail with an `AmbiguousMethodCallException`.
- Added `Main` class to `camel-core` so you can do: Running Camel standalone and have it keep running
- Added JDBC based persistent support for Aggregator EIP
- Fixed Jetty may throw NPE during heavy load.
- Continuation timeout can be configured on Jetty. By default Jetty uses 30 sec timeout. There is also a new option to disable using Jetty continuation.
- Polling Consumer's `PollingConsumerPollStrategy` now provides number of messages actually processed in the poll in the `commit` callback message. You can use this to know if none messages was polled, or that X number of message was polled.
- Dozer now works with OSGi
- Fixed an issue with Aggregator which could lead to it not confirm completions on the `AggregationRepository`. And as well the internal "in progress completed exchanges" map wouldn't neither remove the completed exchange, causing it to grow memory over time.
- The `bean` DSLs will eager check on startup, 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.
- Graceful Shutdown a single route now supports a `abortOnTimeout` option, which if enabled will let the route be running if it could not shutdown the route within a given timeout period.
- Fixed `loadRoutesDefinition` and `addRouteDefinition` to used shared logic from `camel-core-xml` to ensure loading routes from XML files works exactly the same as if the routes was defined in Spring or Blueprint XML files.
- Camel now fail on startup if it cannot load type converters from the classpath, which is essential for it to operate correctly.
- Fixed `completionFromBatchConsumer` option on the Aggregator not aggregating the last incoming Exchange under certain conditions.
- Fixed a few code generation issues related to Camel Maven Archetypes
- Properties component will trim values from loaded properties. This fixes a problem when you have trailing spaces for values in a properties file.
- Fixed `trim` option on `@DataField` annotation not working in Bindy when unmarshalling fixed length records.
- Added `ref` function to Simple expression language so you can lookup any Bean from the Registry.

New Enterprise Integration Patterns

- None

New Components

- EventAdmin - OSGi Event Admin
- JDBC-AggregationRepository - Persist ongoing aggregated message in JDBC data store
- JMX
- Kestrel
- Pax-Logging - OSGi Pax Logging
- Spring Web Services -
- SQS - Queuing service for Amazon Web Services
- Routebox

New DSL

- Added `excludePattern` parameter to `removeHeaders` DSL.

New Annotations

- None

New Data Formats

- JiBX
- Syslog

New Languages

- None

New Examples

- JMX Component Example
- Simple Jira Bot
- Spring Web Services Example

New Tutorials

- None

API breaking

- `camel-test` JAR in the `CamelSpringTestSupport` classes now returns a more generic `AbstractApplicationContext` type in the `createApplicationContext` method.
- Added `prepareShutdown` method to `ShutdownAware` SPI interface.
- `org.apache.camel.spi.IdempotentRepository` now extends `org.apache.camel.Service`.
- The `onThreadPoolAdd` method on `LifecycleStrategy` has additional parameters for information where the thread pool is used.
- `processBatch` method in `BatchConsumer` now returns number of messages that was actually processed from the batch.
- `commit` method in `PollingConsumerPollStrategy` now has a parameter with the number of message processed in the poll.
- `poll` method in `ScheduledPollConsumer` now returns number of messages that was actually processed from the poll.
- `TypeConverterLoaded` has been moved from package `org.apache.camel.impl.converter` to `org.apache.camel.spi`.

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.
- Project cannot be fully build using Maven 3.0
- One may encounter build errors in camel-ftp with java versions older than "1.5.0_24"
- `camel-script` which uses `ScriptEngineManager` from the JDK, cannot load script engines from packages outside the JDK (i.e. when running in OSGi).
- Stopping a route after using `adviceWith` maybe cause issues with error handlers, see CAMEL-3534
- If you use Quartz then pay attention that its UpdateChecker is enabled by default. You should disable this, see more details here
- Installing `camel-ftp` feature in Apache ServiceMix may cause issue. See here for more details and workaround.
- Splitter in parallel mode may block if empty body as input. See CAMEL-3685
- Recipient List in parallel mode doesn't reuse background aggregator thread. See CAMEL-3727
- The `camel-spring-ws` feature does not work in Apache ServiceMix or Apache Karaf.

Important changes to consider when upgrading

> Starting with camel-2.7.0 we are dropping support for java 1.5, spring 2.5.6 and junit 3. These technologies may still be supported for a short while on a 2.6.x branch depending on the community interest. See Camel 2.7 - Roadmap.

- Upgraded to Spring 3.0.5

- Upgraded to Apache CXF 2.3.1 (although CXF 2.2.11 is still supported)
- Upgraded to Jetty 7.2.2.v20101205
- The Java DSL now enforces `onException` and the likes to be defined **before** routes, if not Camel will fail starting the route.
- The Aggregator EIP will use a synchronous invocation when processing completed aggregated Exchanges, when not running in parallel mode. This ensures no internal task queue is used, which otherwise could stack up tasks and under certain conditions eat up memory. Enable `parallelProcessing` if you want to use a separate thread for processing completed aggregated Exchanges.
- Camel now fails fast if staring a `LifecycleStrategy` fails on startup. For example if there is issue with JMX that will now cause Camel to not startup at all. If you have trouble with JMX before you can disable JMX on Camel and allow it to startup.
- `ManagementAware` has been @deprecated. Just enrich your class with the Spring JMX annotations, if you want custom management of your components, endpoints, processors and so fort. See more at Why is my processor not showing up in JConsole.
- Data Format now supports `Service` and `CamelContextAware` callbacks which means its much easier to perform custom initialization logic in your data formats.
- The `mediaSize` option on Printer now uses exact same constants as the Java Printer API. Which means it uses underscores instead of hyphens in the constants.
- JPA `JpaTraceEventMessage` now uses `@Lob` for fields which may contain a lot of data, such as message body.
- Quartz now enforces trigger name/group to be unique within the given component. Will throw exception on name clash detected. This behavior does not apply for clustered quartz.
- Camel now fail on startup if it cannot load type converters from the classpath, which is essential for it to operate correctly.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.6.0.zip	apache-camel-2.6.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.6.0.tar.gz	apache-camel-2.6.0.tar.gz.asc

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	apache-camel-2.6.0-src.zip	apache-camel-2.6.0-src.zip.asc
Source for Unix/Linux/Cygwin	apache-camel-2.6.0-src.tar.gz	apache-camel-2.6.0-src.tar.gz.asc

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>2.6.0</version>
</dependency>
```

SVN Tag Checkout

```
svn co http://svn.apache.org/repos/asf/camel/tags/camel-2.6.0
```

Changelog

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

- JIRA Release notes for 2.6.0