Camel 2.3.0 Release

Camel 2.3.0 release

New and Noteworthy

Welcome to the 2.3.0 release which approx 276 issues resolved (new features, improvements and bug fixes such as...)

- Overhauled threading model and introducing `threadPoolProfile` to define profiles for thread pools to have easy configuration.
- Property placeholders in for example endpoint uris is not natively supported in **camel-core** which means you no long need to use Spring for that. It works therefore in any environment as its provided out of the box in camel-core.
- Property placeholders also works in the `<CamelContext>` when using Spring XML. Basically nearly all places where you enter text the property placeholder is supported.
- Total overhaul of the Aggregator EIP, which now has better completion triggers, and supports pluggable repository. For example to use camel-hawtdb as persistent store. Also added support for recovery and transactional behavior when using camel-hawtdb.
- Added `ExchangeSentEvent` to `EventNotifier` which contains time taken and is emitted when an Exchange is sent to an Endpoint. This allows end users to easily gather performance stats for Exchange send to endpoints.
- Added `disconnect` option to MINA to close Mina session right after usage.
- Jetty now supports to get the `HttpServletRequest` and `HttpServletResponse` from the Message header.
- XPathBuilder now supports being used without an Exchange which allows you to use it in a custom/generic fashion.
- XPath now supports using the JVM system property specifying a custom `XPathFactory` to be used. You can use this to switch from default to use e.g. Saxon.
- XSLT now supports using `<xsl:include>` where the files is loaded from classpath, and being able to load relative according to the endpoint configured location. See the wiki page for details.
- File, FTP added option `eagerDeleteTargetFile` to control the behavior when using `tempFile` whether or not to eagerly delete the target file or wait until last moment.
- Reduced registering producers in JMX to prevent using too much memory and potentially registering short lived producers which didn't bring much value of being managed as well.
- Added camel-http4 component using Apache HTTP Client 4.0.1. This can be used by early adaptors to try out the new Apache HTTP Client 4.x architecture.
- Upgraded Jetty to use Jetty 7.0.1.
- Individual routes can be Graceful Shutdown at runtime, which allows you to stop the route in a more reliable and gentle way.
- Added `removeHeaders` to remove multiple headers, e.g. use `removeHeaders("Camel*")` to remove all Camel related headers.
- Improved the Failover load balancer to support round robin mode and a few other options as well.
- Fixed issue with Camel not being able to run deployed as a WAR in WebLogic server.
- Fixed Polling Consumer not working with JMS selector.
- pollEnrich now handover completions which means you can use the `move` like options from File or FTP component. For example to enrich with a file and have that file moved when the route completes.
- Added @Attachments annotation to Bean binding.
- Simple and Bean language now has a build in OGNL notation which allows you to reference from `Map` or `List` structures and as well to invoke methods in a OGNL like notation. See more details at Simple wiki page.
- camel-jetty supports the multipart/form post out of box.
- Graceful Shutdown now shut down routes in the reverse order in which they was started. Option `shutdownRoutesInReverseOrder` can be used to control this behavior.
- XSLT component now allows you to chose which output type to use with the `output` option. For example you can now stream directly to a file, for example when transforming very big xml messages.
- Error Handler can be configure with specialized <errorHandler/> tag to make it easier to configure error handling in Spring XML.
- Added `depends-on` attribute to `<camelContext/>` so you can have other beans created before Camel when using Spring XML.
- Added `org.apache.camel.builder.ProxyBuilder` to easier create Camel Proxy in Java / Java DSL.
- `ProducerTemplate` and `ConsumerTemplate` can now be configured with a `maximumCacheSize` to control how many producers /consumers they can cache.
- Methods on `CamelContext} to create {{ProducerTemplate` and `ConsumerTemplate` now pre starts the templates so they are ready to use asap.
- Fixed polling Files on a Windows network share may skip files with a log message `Ignoring unsupported file type for file`.
- `ProducerCache` is now exposed for management in JMX which allows you to 'keep an eye on it' at runtime. This cache is used by certain EIP patterns.
- Fixed polling Files or FTPs when recursive and idempotent is enabled, which could cause Camel to not pickup files in sibling folders with similar name to already consumed files.
- Improved request/reply messaging over SEDA and VM endpoints to work properly with many endpoints (chained).
- JPA component will now auto lookup `EntityManagerFactory` and `TransactionManager` from Registry to support convention over configuration. Then you do not any longer need to explicit configure this on the `JpaComoponent`. If auto lookup failed a `WARN` log is logged.
- Fixed camel-spring having factory beans for `ProducerTemplate` and `ConsumerTemplate` set to non singleton, which caused it to create new instances. Obviously it should have been singleton as you want to reuse the template.
- Fixed camel-spring when @Produce/@Consume/@EndpointInjected annotated was used on prototype scoped beans, to **not** register those annotated services to be closed in `CamelContext`. This caused Camel to keep references to these services even when the prototype was no longer in used. And causing Camel to eat memory.

- Added camel-spring-security component, now you can use spring security within camel route.
- Mock endpoints now report more detailed assertion failures when binary predicates are used, as you can see the evaluated expression which failed.
- Improved the Tracer to make it easier for Camel end users to extend it and use custom tracing such as storing to a database using JPA but with their own `@Entity` class.
- Improved Wire Tap when sending a new Exchange by having access to the original Exchange so you can grab that information when creating the new Exchange.
- Fixed `Policy` to be applicable per processor, multiple processors or the entire route
- Camel now checks more aggressively on startup if route nodes which mandates child nodes also have such configured. If not an exception will be thrown on startup.
- XPath is now easier to work with as we added more type converters which supports the default type, `org.w3c.dom.NodeList`, use by XPath.
- When using Spring XML you can now import routes from other XML files
- Much friendlier syntax to configure time periods in endpoint uris, for example the Timer component. Or how often to poll from a File or FTP consumer.
- Camel Http component will not consumer the message body when it parses the POST message with the content-type of "application/x-www-form-urlencoded", and it also honors the charset setting of the "content-type".
- Now you can enable or disable gzip processing by setting the exchange property with name Exchange.SKIP_GZIP_ENCODING and value Boolean.True in Camel Http component.
- Improved `NodeList` to `String` converter to include xml tags, attributes and whatnot, eg you get outputs such as `<foo id="123">bar<year>2010</year></foo>`.
- Added `continued` option to Exception Clause so you can use it catch the exception and **continue** routing from the point of where the exception was thrown (Camel will continue to the next processor in the route graph).
- Recipient List and Routing Slip now has option `ignoreInvalidEndpoints` to just ignore and continue if an endpoint uri was invalid.
- Now you can override the default XSL output properties by setting the CamelContext properties
- Now camel provides a karaf feature file with Spring 3.0.2.RELEASE, the features url is "mvn:org.apache.camel.karaf/apache-camel/2.3.0/xml/features-spring3"

New Enterprise Integration Patterns

New Components

- camel-nagios for sending passive checks to Nagios
- properties for using property placeholders to resolve endpoint uris.
- camel-gae extensions
  - gauth component for implementing OAuth consumers.
  - glogin component for programmatic login to Google App Engine applications from Java clients.
- camel-hawtdb as persistent store for the Aggregator EIP.
- camel-netty for working with TCP and UDP protocols using Java NIO based capabilities offered by the JBoss Netty.
- camel-exec for executing system commands
- camel-bean-validator performs bean validation of the message body using the Java Bean Validation API (JSR 303).
- camel-spring-security support to integrate the spring security with Camel.
- camel-crypto using Camel cryptographic endpoints and Java's Cryptographic extension it is easy to create Digital Signatures for Exchanges.
- camel-eclipse for using Camel with Eclipse RCP.

New DSL

- `removeHeaders`
- `validate`

New Annotations

- `@Attachments`

New Data Formats

- SOAP data format provides basic webservice support without the CXF Stack
- Crypto data format integrates the Java Cryptographic Extension into Camel

New Languages

New Examples

- camel-example-aggregate shows the new overhauled Aggregator in use with persistence storage using HawtDB.
- camel-example-spring-security shows how to use camel-spring-security to implement role based authorization.
- camel-example-loadbalancing-mina shows how to use Load Balancer EIP to balance communication with remote servers using MINA.

New Tutorials

- The OAuth tutorial demonstrates how to implement OAuth in web applications with Camel's new gauth component.

API breaking

The Aggregate has been overhauled and thus you need to migrate you Camel application if you use it.
See the Aggregate wiki page for which options it has you should use.

In MINA the header key `MinaConsumer.HEADER_CLOSE_SESSION_WHEN_COMPLETE` is moved to `MinaConstants.MINA_CLOSE_SESSION_WHEN_COMPLETE`

`org.apache.camel.spi.PollingConsumerPollStrategy` now returns a boolean in the `begin` method. Use `true` to accept to begin polling, and `false` to skip polling at this time.

A new method `onInit` has been added to `org.apache.camel.spi.RoutePolicy`.

Added method `removeHeaders` on the `org.apache.camel.Message` API.

`getExecutorService/setExecutorService` have been removed from `DefaultComponent` and `DefaultEndpoint`.
You should use `CamelContext.getExecutorServiceStrategy()` which is the API for creating thread pools in Camel.

Changed `Exception` to `Throwable` on the `org.apache.camel.spi.EventFactory` and the various `xxxEvent` objects as Camel now catches `Throwable` on shutdown to ensure a more robust shutdown in case a `Throwable` was thrown.

Removed unused class `org.apache.camel.spi.Provider`.

Fixed spelling in the `parallelProcessing` option from the `@RecipientList` annotation.

`GenericFile` is no longer `java.io.Serializable`

Method `adviceWith` on `RouteDefinition` now requires `CamelContext` as first parameter.

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.

If using camel-quartz in OSGi it may stop the scheduler if you update bundles with Camel applications using Quartz endpoints.
See more details at this discussion.

There is a potential dead lock in the Aggregator when using timeout condition. See more at CAMEL-2824.

The LoggingErrorHandler does not work when also using onException. It may go into an endless loop. So refrain from using logging error handler.

There is a concurrency issue when using predicates using binary operators such as `isEqualTo`. See CAMEL-3188.

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.

Important changes to consider when upgrading

The Aggregator has been overhauled and thus you need to migrate you Camel application if you use it.
See the Aggregator wiki page for which options it has you should use.

The ToAsync feature has been **@deprecated**. It will be replaced by a better asynchronous routing engine in the next Camel release.

The SEDA endpoint is now by default unbounded in size, where as before they had a default size of 1000.

camel-jetty has been upgraded to use Jetty 7.0.1 from 6.1.22. These two versions is much different as 7.x is moved to be hosted at Eclipse, which means all it package names has been renamed.

In camel-http some options in relation to authentication has been renamed to avoid clashing with using `username` and `password` as uri parameters, which can be common names to be used. Therefore if you have configured authentication on the HTTP endpoint you have to rename those options, and as well a authentication method (`authMethod`) must be provided as well (BASIC, DIGEST, NTML).

Camel will now shutdown routes in reverse order in which they where started.

The outbound Exchange from Splitter will now by default use the original Exchange (input to Splitter). To use the old behavior you can configure the Splitter to use `UseLatestAggregationStrategy`.

The camel-cxf component PAYLOAD mode has been improved to delegate all SOAP message parsing to CXF. The camel-cxf component now requires CXF 2.2.8 or newer.

XPath or working with XML. Camel now converters `NodeList` to `String` a bit differently as it now outputs the String including the XML tags, childtags, attributes etc. If using XPath you can use the `text()` function to grab only the text content.

The default charset which is used in camel converters can be set by using the System property with the key `org.apache.camel.default.charset`, if not set it will fallback to use `UTF-8` by default.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.3.0.zip	apache-camel-2.3.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.3.0.tar.gz	apache-camel-2.3.0.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-2.3.0-src.zip	apache-camel-2.3.0-src.zip.asc
Source for Unix/Linux/Cygwin	apache-camel-2.3.0-src.tar.gz	apache-camel-2.3.0-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>2.3.0</version>
</dependency>
```

SVN Tag Checkout

```
svn co http://svn.apache.org/repos/asf/camel/tags/camel-2.3.0
```

Changelog

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

- JIRA Release notes for 2.3.0