Camel 2.9.0 Release

Camel 2.9.0 release

ked URL

New and Noteworthy

Welcome to the 2.9.0 release which approx 497 issues resolved (new features, improvements and bug fixes such as...)

- Introduced `ThreadPoolFactory` with a simpler API for 3rd party SPI. See Threading Model for more details.
- The `ScheduledExecutorService` created by Camel is guarded with a queue size check, to guard against OOME due the JDK thread pool is unbounded task queue.
- The SOAP data format now supports using multiple parameters on the JAXB objects.
- The Bean component now supports specifying parameter values to use directly in the method syntax. See more details at Bean Binding. For example to invoke a bean with body as first parameter, and a boolean true as second you can now do: `.to("bean:myBean?method=myMethod(body, true)")`
- The Bean component supports binding to private class beans by fallback to use interface methods.
- Improved labels of all model definitions which appear in Tracer output
- Reduced dependency for `camel-core` to not require `commons-management` JAR anymore
- Added blueprint support of cxfrs endpoint.
- Camel no longer depend on using Spring JARs for having Camel enlisted in JMX.
- Introduced a set of Camel JMX annotations in `org.apache.camel.api.management` that can be used to easily instrument Camel Components, EIPs, custom Processors, Beans etc. for being managed in JMX. Using the Spring JMX annotations is still supported (you would need `camel-spring` on the classpath).
- Updated the Aggregator2 to allow for manually completing all exchange groups by sending a message containing the header `Exchange.AGGREGATION_COMPLETE_ALL_GROUPS` set to true.
- Updated the Aggregator2 to support forceCompletionOnStop option to complete all current aggregated exchanges when the context is stopped.
- Removed dependency on Spring JARs in the following components: Validation, XSLT, Velocity, FreeMarker, Flatpack, StringTemplate, iBATIS, Jing, MSV, XQuery, Test, Groovy, Scripting Languages, Mail and Quickfix.
- Using pollEnrich with a file/ftp endpoint will now only poll on demand, beforehand the polling could be started when the route started.
- Added option `ReplyToType` on JMS to allow people to control which kind of reply queue they are using (temporary, shared, exclusive). Added support for exclusive reply queues. Mind they have some limitations, by which the shared dont have. So use exclusive with a bit caution.
- Added `wereSentTo` to NotifyBuilder allowing to set expectation that Exchanges was sent to the given endpoint uri.
- Bindy in CSV format now uses trim, clip, and length attributes on @DataField when marshalling.
- Improved managed browsable endpoints (JMX) so you can browse all/range messages and whether to include message body or not.
- File and FTP endpoints is now browsable endpoints. For example using JMX people can browse the endpoint listing available files. The returned data does **not** contain the actual file body to avoid loading big data into memory.
- Added `replaceFromWith` to AdviceWith making it easy to swap a route input endpoint with something else, for easier unit testing.
- Added Java DSL support for route autoStartup(String) and autoStartup(boolean)
- Switched to a better performing non locking endpoint cache based on Google's ConcurrentLinkedHashMap.
- Added `isUseAdviceWith` to Camel Test which allows end users to tell Camel that the unit tests uses AdviceWith which allows you to defer starting CamelContext until after the adviceWith has been executed.
- Added `filtersRef` option to Jetty endpoint, and it supports to setup the Filters on the Jetty endpoints.
- Added `ciphers` option to the SFTP endpoint to allow configuration of the ciphers used by default.
- Added `fastExitsCheck` option to the SFTP FTP endpoint to check the file exits fast.
- Added `errorHandler` option to the JMS component which allows you to set a `org.springframework.util.ErrorHandler` to be invoked in case of any uncaught exceptions thrown while processing a `Message`.
- Added `idleConsumerLimit` option to the JMS component which sets the limit for the number of consumers that are allowed to be idle at any given time.
- Simple language has a much better syntax parser, being able to report what and where syntax errors are. Added unary operators, which for example can be used to increment counters.
- Simple language now supports nested functions, so you can do "${header.${header.bar}}", for example to refer to a header key from another header.
- Simple language can now configure custom start/end tokens for its built-in functions.
- The JPA component has been upgraded from using the JPA1 spec to use JPA2 spec.
- The CSV component now always returns a List<List> regardless if 1 or more rows returned, to be consistent.
- Resequencer now pre validates incoming Exchange and throws `CamelExchangeException` if invalid. Added new option `ignoreInvalidExchanges` to suppress the exception and ignore invalid exchanges.
- Improved existing and added more Karaf Camel commands.

- Improved Bean component and method call expressions to be able to invoke static methods, without creating a new instance of the class.
- JMX now exposes average load statics on CamelContext and Route mbeans. Its similar to unix load stats per 1, 5 and 15 minutes.
- Added support for `minDepth` and `maxDepth` option on the FTP consumers.
- Improved the AWS-S3 component to support more headers in the producer.
- Cometd now includes headers in the server message.
- SFTP now supports interactive keyboard authentication, but entering the password in the prompt.
- ManagedTracer now supports detailed JMX TraceEvent Notifications. This allows to watch message flows using JConsole and provides an ideal base for external tooling
- Exchange.HTTP_URI query part will take effect on the producer of HTTP and HTTP4.
- Add `camelContextId` to MDC logging.
- Hadoop consumer supports auto merging multiple segment files.
- JMS consumer has new option `asyncConsumer` which allows to enable Asynchronous Routing Engine
- The camel-archetype-component allows to specify the component name.
- Polling Consumers such as File or FTP supports new option `sendEmptyMessageWhenIdle` to send an empty message if there was no messages to poll.
- Added option `allowStAX` on XSLT and XQuery to control whether StAX (Streaming API for XML) is preferred as the `javax.xml.transform.Source`.
- Added `properties` function to Scripting Languages to make it easier to leverage Camel Properties component from scripts.
- Added `timeoutCheckerExecutorService` and `timeoutCheckerExecutorServiceRef` options to the aggregator. If using either of the `completionTimeout`, `completionTimeoutExpression`, or `completionInterval` options a background thread is created to check for the completion for every aggregator. This doesn't scale well when you have many routes with an aggregator (a thread is created for every aggregator). Setting these options allow you to control the number of threads by providing a custom thread pool.
- Splitting big files such as XML files can now be done in a streamed low-memory footprint mode using the Tokenizer language or the StAX component.
- The Language component can now load scripts from classpath or file system. As well having a cached option.
- Further improved masking of passwords in URIs specified in the user info part, in the Camel logs.
- Optimized Request Reply over JMS when using named replyTo queues to cache consumer if possible, making it go faster. This applies when using either the `replytToType=Exclusive` or `replyToDestinationSelectorName` options.
- Fixed JPA consumer to ensure to rollback if processing an Exchange failed.
- Added `quote` option to `@CsvRecord` in Bindy to make it easy to specify using a quote character when marshalling from Object to CSV format.
- Added `consumer.transacted` option to JPA to control whether the consumer should regard the batch as a single encompassing transaction or not.
- Added ability in XSLT component to dynamically override stylesheet used via `CamelXsltResourceUri` message header.
- Added ability in AWS-SNS component to specify the topic policy.
- Added ability in AWS-S3 component to specify the bucket policy and the storage class.
- Log component and LogEIP now supports the Marker option
- Log and Tracer now have a default `maxChars` option at `10000` to avoid logging very big data.
- Added ability in SQL to execute SQL batch statements.
- Added the configuration support of queueSize and concurrentConsumers options on the SEDA VM component.
- Added MBean interfaces in `org.apache.camel.api.management.mbean` which clients can use to access Camel MBeans.* Added MBean interfaces in `org.apache.camel.api.management.mbean` which clients can use to access Camel MBeans.
- Introduced `AbstractJdbcMessageIdRepository` in SQL Component to make it easier to create custom JDBC based Idempotent Consumer

Fixed issues

- Fixed issue with Properties component may cache duplicates and thus take up unnecessary memory in the cache
- Fixed issue with Tracer out exchanges when using Asynchronous Processing by the Asynchronous Routing Engine
- Fixed issue with Components or Endpoints not being registered in JMX when using POJO Producing, POJO Consuming or other Camel bean annotations.
- Fixed issue which could cause `DefaultUnitOfWork` to throw a `java.util.EmptyStackException` under heavy load
- Fixed issue with MDC logging not remembering old values, for example when using a `ProducerTemplate` from within routes to send new messages.
- Fixed issue when invoking a Bean and that bean is using a `@Bean` in its method signature, causing the `@Bean` to not invoke its intended method
- Fixed issue with Bindy unmarshalling in key value or CSV or mode, may cause inconsistency for multiple threaded unmarshalling
- Fixed issue with `ExchangeSentEvent` being fired to early when being sent to asynchronously endpoint. Now its being fired when sent is done.
- Fixed issue with OSGi Blueprint when consuming JAXB annotated objects from ActiveMQ causing ClassCastException's.
- Fixed issue with using `OnException` and using `redeliveryPolicyRef` which could cause the Error Handler to fail with an assertion error at first redelivery attempt.
- Fixed issue with using `doneFile` option on FTP when from a Windows computer connecting to a Unix FTP server, due to wrong path separator in use.
- Fixed issue with Bindy in CSV mode when marshalling not using correct separator char if given an escaped separator char on the model class.
- Fixed issue with evaluation of Simple language not being as fast as it could be.
- Fixed issue with File consumer not auto creating start dir if having dot in name.
- Fixed issue with Splitter, Recipient List and Routing Slip throwing exceptions from custom expressions, not being triggered by Exception Clauses.
- Fixed issue with stale OUT message set on Exchange if exception was thrown from `unmarshal` or `marshal` operations from Data Format
- Fixed issue with request/reply over JMS when using `useMessageIDAsCorrelationID=true` and fixed reply queues, causing JMSMessage selector to not evict timed out correlationIDs, which cause the JMSMessage selector to keep growing.
- Fixed issue with XSLT, XQuery and XPath may lock file resource if exception occurred during transformation
- Fixed issue with Flatpack dropping all headers from the original message
- Fixed issue with `wereSentTo` on NotifyBuilder may trigger to soon, as well you no longer must configured it at first
- Fixed issue with SMPP by sending multipart and multilingual messages
- Fixed issue with using Saxon for XPath expressions with a Splitter and Content Based Router in the same route.
- Fixed issue with thread safety of the `XPathFactory` in the XPath builder
- Fixed issue when stopping Batch Consumer routes and have configured route with `CompleteAllTask`, there may be a slim chance the route will be shutdown too early before the last batch message had a chance to be enlisted as in flight exchange.
- Fixed issue with FTP consumers to always process synchronous, to ensure the UnitOfWork executes the done tasks in the same thread as the consumer, as the FTP libraries is not thread safe.
- Fixed memory leak when doing request/reply over JMS with a fixed reply queue, when under high load.

- Fixed issue with Bindy in CSV mode not handling separator char to be inside a quoted field. For example a comma, in "Wednesday, November 9th, 2011"
- Fixed issue when restarting CamelContext would lose any prior custom `LifecycleStrategy` in use.
- Fixed issue with using the same ScheduledRoutePolicy on 2+ routes not working on all the of the routes.
- Fixed issue that SEDA VM consumer doesn't reset the camel context of exchange when it pulls the exchange from queue.
- Fixed issue when using context scoped Exception Clause to handle exceptions, by which the `RouteScope` on the `UnitOfWork` from the `Exchange` may not refer to last used route id.
- Fixed Throttler EIP in async delayed mode to not eat up memory when messages flow in faster, than the throttler can process the delayed tasks. The throttler will now use the `maxQueueSize` option from the Threading Model to reject tasks if the task queue grows to big.
- Fixed issue with Throttler and Delayer EIP in async delayed mode, would cause a 2nd done on the `AsyncCallback` which could lead to a NPE in the logs.

New Enterprise Integration Patterns

New Components

- AWS-SDB
- AWS-SES
- jclouds
- Krati
- Solr
- StAX
- Zookeeper

DSL Changes

New Annotations

- Added some JMX annotations in org.apache.camel.spi.management that replace the spring jmx annotations to make camel-core independent of spring

New Data Formats

- PGP

New Languages

New Examples

New Tutorials

API breaking

We did some fairly large refactoring in preparation for camel 3.0. In Camel 2.9.0 most changes should be compatible. If you hit any problematic API changes please report in jira and we will try to add compatibility stubs in a 2.9.x release.
The main goal of this refactoring is to prepare for a self contained minimal API that can be used as a basis for component development.
We already did some changes in regard to this in 2.9.0 to make backporting 3.0 features easier once we start with camel 3.0.

- The class `ExecutorServiceStrategy` is deprecated in favor of a new `ExecutorServiceManager` class, which have a slightly reduced and improved API. See Threading Model for more details.
- The `org.apache.camel.component.ResourceBasedComponent` class from `camel-spring` has been deprecated. Use `DefaultComponent` instead.
- The `org.apache.camel.component.ResourceBasedEndpoint` class from `camel-spring` has been deprecated in favor of a new `org.apache.camel.component.ResourceEndpoint` in `camel-core`, and its dependency on Spring JARs has been removed. Use the new APIs on `ResourceEndpoint` to load resources.
- Changed the method `beforePoll` slightly in the `org.apache.camel.PollingConsumerPollingStrategy` class.
- `ServiceSupport` and `EventNotifierSupport` classes are moved to org.apache.camel.support. This package is meant for mostly abstract base classes that make it easier to implement the API. To keep compatibility the old classes are kept as @Deprecated versions (see CAMEL-4358)
- `ServiceSupport` now does no handle child services anymore (was used only in two places in `camel-core`). A new class `ChildServiceSupport` now handles this case. Additionally the `org.apache.camel.StatefulService` interface was created which allows to access `ServiceSupport` methods without knowing the implementation.
- `DataFormatResolver` does not support `resolveDataFormatDefinition` anymore. The current implementations for OSGi and Default where exactly the same so this code was moved to `DefaultCamelContext`. If this is needed please open a jira issue
- All methods in CamelContext that reference model elements where deprecated. An additional interface `org.apache.camel.model.ModelCamelContext` was created to hold these methods
- `Ordered` interface was moved from `org.apache.camel.util` to `org.apache.camel`
- `BytesSource` and `StringSource` were moved from `org.apache.camel.converter.jaxp` to `org.apache.camel` as they are used in several places
- `TimeoutMap` interface was moved from `org.apache.camel.util` to `org.apache.camel`
- `DefaultTimeoutMap` was moved from `org.apache.camel.util` to `org.apache.camel.support` as it needs access to `ServiceSupport`
- Moved some management interfaces from `org.apache.camel.management` to `org.apache.camel.spi.management`
- Moved `DefaultChannel` from `org.apache.camel.processor` to `org.apache.camel.processor.interceptor`

- FileUtil.removeFile added new option, and will throw IOException in case of exception occurred
- The CamelTestSupport's context, template, consumer fields is no longer static (reason for changing is to foster faster unit testing)
- Added onRemove, onStart, onStop, onSuspend, and onResume methods to org.apache.camel.spi.RoutePolicy
- Removed Spring related methods from ScriptBuilder in camel-script
- Added RouteContext as parameter to configure method on ErrorHandlerBuilder
- The Mail component no longer uses Spring API, which means any custom JavaMailSender should now implement org.apache.camel.component.mail.JavaMailSender instead.
- The Quickfix component no longer uses Spring API, which mean if you use XML DSL to setup QuickFix settings, then you need to adjust accordingly. See details Quickfix.

Internal/Non API refactorings that may affect users

- org.apache.camel.WrappedFile interface was introduced as abstraction of GenericFile for usage outside the file component
- ModelHelper was moved from org.apache.camel.util to org.apache.camel.model to avoid references from util to model
- The Karaf commands have been renamed

Known Issues

- The Tracer may not output all details for some situations such as when using onCompletion or intercept etc.
- The project cannot fully build the site using Maven (eg running "mvn site". There is no plan to make this work as the project do not use the maven site.
- The source code cannot fully build with JDK 7

Dependency Upgrades

- AHC 1.6.4 to 1.6.5
- AWS-Java-SDK 1.1.1 to 1.2.12
- CometD Java Server 2.1.1 to 2.3.1
- Commons-pool 1.5.4 to 1.5.6
- CXF 2.4.1 to 2.5.1
- Dozer 5.3.1 to 5.3.2
- EasyMock 2.5.2 to 3.0
- EHCache 2.3.0 to 2.4.3
- Groovy 1.8.0 to 1.8.5
- HAPI 0.5.1 to 1.2
- HawtDB 1.5 to 1.6
- Hazelcast 1.9.3 to 1.9.4.4
- Http4 4.1.1 to 4.1.2
- Jackson 1.8.4 to 1.9.2
- Jackrabbit 1.5.5 to 2.2.4
- JCR API 1.0 to 2.0
- Jersey 1.8 to 1.10
- JIBX 1.2.2 to 1.2.3
- JRuby 1.5.6 to 1.6.5
- MyBatis 3.0.4 to 3.0.6
- Netty 3.2.4 to 3.2.6
- OGNL 2.7.3 to 3.0.2
- QPID 0.5 to 0.12
- QuickFix 1.5.0 to 1.5.1
- Restlet 2.0.9 to 2.0.10
- Scala 2.9.0 to 2.9.1
- Scalate 1.5.0 to 1.5.3
- Smack 3.1.0 to 3.2.0
- Spring Framework 3.0.5 to 3.0.6
- Spring Security 3.0.6 to 3.0.7
- Spring WebService 2.0.2 to 2.0.3
- XStream 1.3.1 to 1.4.1

Important changes to consider when upgrading

- The XSLT, and Validation components has been moved from camel-spring to camel-core as they no longer depend upon Spring JARs
- Building the source with Apache Maven 2.x is **deprecated** and support for that is to be removed in the next release.
- Debugger is now disabled by default in Camel Test. You would need to override isUseDebugger() method and return true to enable it.
- Simple language improved syntax parser. Simple is now more strict and will report syntax errors for invalid input. For example Predicates now requires literal texts to be enclosed in quotes, just as any programming language such as Java would do. Range operator also requires the range to be enclosed in quotes as well. See more details at Simple.
- The JPA component has been upgraded from using the JPA1 spec to use JPA2 spec.
- The CSV component now always returns a List<List> regardless if 1 or more rows returned, to be consistent.

Notice

- Removed ANT support for the Examples
- The commons-management JAR is no longer required as dependency.

- Camel no longer depend on using Spring JARs for having Camel enlisted in JMX.
- Building the source with Apache Maven 2.x is **deprecated** and support for that is to be removed in the next release.
- If you want to deploy Apache Camel 2.9.0 into Apache Karaf, we highly recommend to use the 'jre.properties.cxf' configuration, which is part of Apache Karaf 2.2.5.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.9.0.zip	apache-camel-2.9.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.9.0.tar.gz	apache-camel-2.9.0.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 Distribution	apache-camel-2.9.0-src.zip	apache-camel-2.9.0-src.zip.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>2.9.0</version>
</dependency>
```

SVN Tag Checkout

```
svn co http://svn.apache.org/repos/asf/camel/tags/camel-2.9.0
```

Changelog

For a more detailed view of new features and bug fixes, see the:
TODO: Update link

- release notes for 2.9.0