

Camel 2.16.0 Release

Camel 2.16.0 release

[red URL](#)

New and Noteworthy

Welcome to the 2.16.0 release which approx 600 issues resolved (new features, improvements and bug fixes such as...)

Noteworthy improvements:

- The route models (definition classes) preserve property placeholders, which allows round trip editing, and also prevents showing potential sensitive information
- Added onPrepareFailure option to [Dead Letter Channel](#) to allow a custom processor to prepare the Exchange before sending to the dead letter queue.
- [MongoDB](#) tailable consumers now allow setting the readPreference option.
- Using setFaultBody / setFaultHeader behaves similar to setBody / setHeader to preserve existing headers/attachments by setting on existing IN or OUT message.
- Java DSL using choice with end vs endChoice now validates you use them as intended. Any misconfiguration now throws an exception explaining the problem.
- [POJO Producing](#) and [POJO Consuming](#) lifecycle is now in sync with [CamelContext](#) which ensures it works as similar to Camel [Routes](#) does. Also the created producer/consumer is stopped when [CamelContext](#) is stopping to cleanup properly.
- [JMS](#) component now allows configuring concurrentConsumers/maxConcurrentConsumers separately between from(jms) vs to(jms) using request/reply over JMS
- Added [SEARCH](#) operation to Elasticsearch component
- The [Properties](#) component (property placeholder) now supports an encoding option to allow reading the .properties files using a specific charset such as UTF-8. By default ISO-8859-1 is used (latin1)
- Added receiveBufferSize option to the [FTP](#) component to make download faster. The size is by default 32kb buffer.
- The [Aggregator2](#) EIP now allows an external source using a AggregateController to control the aggregator, such as for forcing completing a group or all groups, or query runtime information from the aggregator. A default controller is provided out of the box if none has been configured.
- The [Aggregator2](#) now enlists in JMX using a ManagedAggregateProcessorMBean which has more information, and also allows using the aggregate controller to control it.
- The [Aggregator2](#) now includes more runtime statistics which can be accessed from Java and JMX about number of incoming and completed exchanges, and per different completion triggers.
- The [Aggregator2](#) now supports pre-completion mode if the aggregation strategy implements PreCompletionAwareAggregationStrategy. This supports the use-case where an incoming Exchange are used to determine if the current aggregated group is completed or not. If completed the group is emitted as-is, and the new incoming Exchange is used to start a new group from scratch.
- Added options to [Mail](#) consumer so it can skip or handle a mail message that cannot be read from the mail server in the batch; and then move on to read the next message. This ensures the consumer can deal with mail boxes that has problematic mails that otherwise will cause no mails to be retrieve by the consumer in the batch.
- The runtime route processors and their corresponding model definition are now linked by their id using HasId, so its easier to obtain the processor or model definition from the getProcessor(id) and getProcessorDefintion(id) api on CamelContext.
- CamelContext now also includes a getManagedProcessor API to easily get the mbean client api for the processor, so you can access the runtime JMX statistics from Java code.
- [Spring Boot](#) applications now [block the main thread of the execution](#)
- [Spring Boot](#) automagically [loads XML routes definitions](#) from classpath
- Added [MainListener](#) to make it easier for Camel Boot / Standalone Camel to implement custom logic to trigger during the starting and stopping of the application.
- Camel Jackson now provides [Map to Object converter](#)
- Ability to refer to existing hazelcast instance by id, using [Hazelcast Component](#)
- [Direct](#) and [Direct-VM](#) now supports the failIfNoConsumers option
- Making it easier to add custom [Type Converter](#) to your Camel applications without having to use the META-INF file. In Spring or Blueprint just add them as <bean>s. See more details at [Type Converter](#) documentation.
- camel-cxf PAYLOAD supports stream cache out of box.
- BridgePropertyPlaceholderConfigurer supports to handle the [jasypt](#) encryption inside and outside of the camelContext.
- The camel-aws component now support Amazon DynamoDB version 2
- The [Wire Tap](#) eip now also emits event notifications when sending the Exchange to the wire tap destination
- [Camel Test](#) has new option isDumpRouteStats that dumps all route usage statistics as XML files in target/camel-route-stats directory, which can be used to inspect "route coverage" reports.

- Now [Camel Netty](#) component doesn't intercept consumers with `httpMethodRestrict=OPTIONS`
- Now [Netty HTTP](#) resolves `OPTIONS` prefix matches earlier
- Using [Load Balancer](#) with Java DSL now builds the route model using the specific model types, which renders a more accurate route model
- Enhanced the `PredicateBuilder` to support `vargs` and `list` in the `"or"` builder method
- Using custom `HttpBinding` with [HTTP](#) and [HTTP4](#) is now easier as they can be instantiated using a default no-arg constructor
- Added `readLock=idempotent` to the [File2](#) consumer which allows to reuse an idempotent repository as read-lock which enables Camel to run reliable in clustered mode, where each node competes for the same files from a shared file system. This requires that the idempotent repository supports clustering such as [Hazelcast Component](#) or [Infinispan](#) - or using the JDBC for a shared database table.
- [Camel RX](#) now ensures that `observe` and `subscribe` are done within an unit of work to ensure on completions is executed, such as they are done when you use regular Camel routes.
- The [JMS](#) component now includes the underlying `javax.jms.Session` instance on the Camel message which allows end users to easier to use client acknowledge mode, in case the client wants to use the session to call `recover` or `rollback`.
- Added `completionEager` option to [Idempotent Consumer](#) to whether to complete the idempotent consumer eager or when the exchange is done.
- Working with the Camel source code in Eclipse with `m2eclipse` plugin improved to avoid endless file changed detection and recompile the source.
- Add option `suppressExceptions` to [JsonPath](#) which allows to ignore exceptions such as `NoSuchPathException` when the json payload does not contain the path defined in the json expression.
- [Spring Boot](#) now creates `CamelContext`, `ProducerTemplate` and `ConsumerTemplate` beans only if those are not defined by the user.
- New `camel-archetype-spring-boot` to create a new Camel Spring Boot project using Maven.
- New `camel-archetype-cdi` to create a new Camel CDI project using Maven.
- [FTP2](#) doesn't support absolute directories in endpoint URIs. In such cases, paths will be treated as relative and `WARN` message will be printed in logs. In future versions, there'll be no such assumption and validation exception will be thrown instead.
- [Swagger](#) module now supports getting api-docs from multiple camel contexts in the JVM. The entry point at `/api-docs` now lists the contexts detected, and you need to append the context id in the path, eg `/api-docs/myCamel`
- [FTP2](#) supports `autoCreate=true` for ftp consumers
- [Rest DSL](#) now supports to define parameter using fine grained details for the [Swagger](#) api documentation. For an example see the `camel-example-servlet-rest-tomcat` in the Apache Camel distribution.
- The [Simple](#) language now has a `equals ignore case` operator (`=~`) when comparing with string values
- If using `<contextScan>` with Spring or Blueprint to filter `RouteBuilder` classes, then Camel will now by default only look for singleton beans. You can turn on the old behavior to include prototype scoped with the new option `includeNonSingletons`
- The [IdempotentRepository](#) now supports `clear` operation, to clear the repository, and expose it over JMX.
- [Netty4](#) in `clientMode` will now automatic re-connect if the connection was disconnected to the server.
- The `throwException EIP` now supports creating a new exception instance from a simple language expressions, which allows to make it dynamic based on information from the current Exchange.
- When using [Expression](#) or [Predicate](#) in the routes, you can refer to the value as an external resource by using `resource:classpath:path` or `resource:file:path` - for example `resource:classpath:com/foo/myscript.groovy`.
- Added option `useMessageBodyForSql` to the [SQL Component](#) which allows to use more dynamic SQL queries instead of the fixed SQL defined in the endpoint uri.
- Added sticky option to failover [Load Balancer](#) so the load balancer can start from the last known good endpoint.
- Added option `shareUnitOfWork` to [Content Enricher](#) eip
- [Content Enricher](#) with `enrich` and `pollEnrich` now supports dynamic endpoint uris computed using an [Expression](#) that allows to use values from the current [Exchange](#) so you for example can poll a file which name is computed from data from the [Exchange](#) etc. This change breaks the XML DSL a little bit and there is an easy migration effort. The Java DSL stays backwards compatible.
- The Camel Karaf commands about routes now no longer require the context-id as 2nd parameter that is optional - if none provided then the command is a bulk operation for all Camel contexts.
- A simpler version of dynamic to with the new `<toD>` and `toD` in the DSL, documented at [Message Endpoint](#)
- [Wire Tap](#) now supports dynamic endpoint uris, in similar way the new `<toD>` and `toD` does
- [Rest DSL](#) now supports the new `<toD>` and `toD` in the rest-dsl.
- The `EIP's` now have more specialized processor mbeans in [Camel JMX](#) and you can also dump the processor as xml source to represent how it would appear in xml dsl
- Configuring [Camel JMX](#) using JVM system properties now all take precedence over any existing configuration (some option was not doing that)
- Introduced Extended JMX statistics level to gather additional metrics during runtime.
- All the [Camel-AWS](#) components now support the definition (through URI options) of Proxy Host and Proxy Port.
- The HTTP based consumers no longer include Camel headers in the responses by default (they already did this for the producers to not include Camel headers when calling a HTTP service).
- [Bindy](#) now supports having multiple pojos as models in the same java package. Instead of configuring using package names, you configure using class names.
- You can configure on the [Type Converter](#) registry what should happen if a duplicate type converter is being added (override, ignore or fail) as well configure the logging level.
- Using [CamelProxy](#) now binds the method parameters to the message body/header using Camel annotations to define the binding rules. If no annotations is defined the the parameter is assumed the message body. You can turn this off to have the old behavior.
- [Camel JMX](#) using custom JMX operations/attributes with `@ManagedResource` now also includes the default set of JMX from Camel, so you have both worlds.
- Added `completeAllOnStop` option to [Aggregator2](#) to allow to complete all pending exchanges from the aggregation repository when stopping. For example when using a memory based repository to ensure all of those exchanges are processed when stopping.
- Allow to configure the OK status code ranges for [HTTP](#), [HTTP4](#), [Netty HTTP](#), [Netty4 HTTP](#), and [Jetty](#) components for their producers. The default range is 200-299.
- Now [Vert.x](#) endpoints support [multiple consumers](#).
- Support for TAR files: marshalling and unmarshalling via a data format; splitting; aggregating. See *Data Formats* section below.
- [SJMS](#) component has been aligned to bind between Camel Message and JMS Message in the same way as [JMS](#) component does that.
- Added failsafe check to ensure [JMS](#) consumer will no longer send back a reply message to the same destination as itself to avoid endless loop. There is an option to turn this off in case you know what you are doing.
- The [Camel-hazelcast](#) component now supports the Ringbuffer data structure.
- New XSLT Aggregation Strategy to merge two messages with an XSLT file. See [javadoc](#) (link valid once this release is published). It is also accessible from the toolbox: `AggregationStrategies.xslt()`.
- A pure Java based [Swagger](#) component that also supports any JVM (standalone, war, jee, spring boot, et all) as the platform for API of the Camel [Rest DSL](#). (previously a servlet was required).

- [Data Format](#)'s is now also enlisted in JMX under the dataformats group (requires the data format implementation is a Camel Service)
- Configuring context-path in [Rest DSL](#) will now use the context-path when creating the http server (except for servlet as servlet urls are relative)
- OptaPlanner compoent extended to support async solving, daemon mode and ProblemFactChange processing

Fixed the following issues:

- File language now properly supports file extensions having double dots, such as .tar.gz
- Fixed issue with doTry not stopping routing if a fault message was set using setFaultBody within the doTry.
- Fixed [Bean](#) language to support end parenthesis in quoted string literals as part of method parameters, and not as end of method signature.
- Fixed [FTP2](#) using pollEnrich with disconnect=true and delete=true, would disconnect before deleting the file. Now we disconnect last.
- Various EIPs such as [Splitter](#), [Recipient List](#), [Multicast](#) no longer reuse message id in their sub messages, but each sub message has its own unique message id (just as the exchange has).
- Fixed an issue with the circuit breaker [Load Balancer](#) not honoring the half open after period after an error was detected.
- Fixed [FTP2](#) now calling disconnect if there was no messages to poll, even when disconnect=true configured.
- Fixed [Throttler](#) using method call expression in XML DSL was not working
- Fixed stopping a route may stop context scoped OnException that would otherwise send message to an endpoint, now been stopped and an IllegalStateException was thrown while handling the exception.
- Fixed using a [Load Balancer](#) from a context scoped onException may double up its processors participating in the balancer per route in the context.
- Fixed parallelAggregate on multicast/splitter/recipient list to use a thread pool for running concurrently.
- Fixed [Stream caching](#) to not delete data that was spooled to disk before both exchanges are done routing.
- Fixed [DefaultErrorHandler](#) to use 1 sec redelivery delay out of the box as the other error handlers does.
- Fixed setting logExhaustedMessageHistory=true on [Dead Letter Channel](#) to log the message history. This option is by default false on DLC as it would be as before (not noisy).
- Fixed double refresh of the parent Spring Boot Application Context
- Fixed configuring endpoint uri's with primitive types when the value is looked up in the registry using #notation would not work.
- Fixed using transacted in the Scala DSL
- camel-swagger now filters out older Camel releases that do not support rest-dsl when it discovers Camel's in the JVM
- Fixed [Mail](#) consumer to use the default value of 60 seconds as poll interval again, instead of 0.5 sec.
- Fixed [Netty](#) and [Netty4](#) may perform x2 redelivery when attempting to redeliver to a netty producer endpoint
- Fixed [Recipient List](#) would not work if using RAW() syntax in endpoint uri
- Fixed Camel Commands to not show correct route status for suspended routes in the routes-list command.
- Fixed using [HTTP](#) and Exchange.HTTP_QUERY or Exchange.HTTP_URI may double encode the value.
- Fixed starting a route from a file endpoint, and then do a pollEnrich from another file endpoint, and both using the same readLock=marker, would only delete one of the marker files when done.
- Fixed [JGroups](#) managed routes starting too early.
- Fixed [HTTP4](#), [HTTP](#), [Netty4 HTTP](#), [Netty HTTP](#) producers forming URLs badly when using Exchange.HTTP_PATH and query parameters inside the endpoint URI.
- Fixed [MQTT](#) receiving duplicates in some cases when the connection to the broker was lost.
- Fixed [Routing Slip](#) and [Dynamic Router](#) to not evaluate expression again during each redelivery attempt from [Error Handler](#) if routing caused an exception.
- Fixed a JMX leak where FTP producers was not unregistered from JMX when their routes are removed.
- Fixed suspend/resume [SEDA](#) routes may in rare cases not poll new messages from the seda queue when resumed.

New Enterprise Integration Patterns

New Components

- [camel-paho](#) - For MQTT messaging using Eclipse Paho client
- [camel-pdf](#) - For creating PDF documents
- [camel-grape](#) - allows you to fetch, load and manage additional jars when CamelContext is running.
- [camel-slack](#) - allows you to connect to an instance of Slack and delivers a message contained in the message body via a pre established Slack incoming webhook.
- [camel-jolt](#) - for JSON to JSON transformation
- [camel-aws-ec2](#) - Component providing support to Amazon Web Services EC2
- [camel-undertow](#) - To use [Undertow](#) as HTTP server or client.
- [camel-git](#) - A generic Git component
- [camel-sjms](#) - SJMS Batch is a specialized component for highly performant, transactional batch consumption from a JMS queue
- [camel-http-common](#) - A common base component for reuse among all the various HTTP components we have.
- [camel-swagger-java](#) - A pure Java based [Swagger](#) component.
- [camel-elsql](#) - An extended [SQL Component](#) that uses ElSql for defining SQL queries
- [camel-jbpm](#) - Sends messages through kie-remote-client API to jBPM.

New DSL

- [Script](#) - to execute a script during routing
- [Dynamic To](#) - to send a message to a dynamic computed endpoint. This is a simpler version than using [Recipient List](#)

New Annotations

New Data Formats

- [camel-boon](#) - For mapping POJO to/from json.

- camel-tarfile (pending documentation) - For compressing and extracting TAR files; also offers a Splitter (for processing TAR entries individually) and an Aggregation Strategy (for collecting the TAR entries into an archive).

New Languages

New Examples

New Tutorials

API breaking

- org.apache.camel.mode.LoadBalancerDefinition no longer implements org.apache.camel.processor.loadbalancer.LoadBalancer which is the runtime processor (this was never intended).
- The ref attribute on <loadBalance> has been removed, as it has been deprecated for a long time, and you should use a <customLoadBalancer> to refer to a custom load balancer.
- The copy method on StreamCache now takes an Exchange as parameter.
- Various APIs in camel-jms has been adjusted to support including the JMS session parameter - javax.jms.Session. Though these API changes are mostly internal facing, and we do not expect end users problems.
- The resourceUri and resourceRef attributes on <enrich> and <pollEnrich> has been removed as they now support a dynamic uris computed from an [Expression](#).
- Various APIs from camel-http in the package org.apache.camel.component.http has been moved to the camel-http-common module in the package org.apache.camel.http.common which mean you may need to change the imports.
- Renamed All enum on org.apache.camel.ManagementStatisticsLevel to Default
- Added new boolean parameter to method on org.apache.camel.spi.ShutdownPrepared
- Added configure method to allowing configuring CamelContext on org.apache.camel.main.MainListener
- Renamed org.apache.camel.component.sjms.jms.KeyFormatStrategy to org.apache.camel.component.sjms.jms.JmsKeyFormatStrategy

Known Issues

Dependency upgrades

- Amazon Web Service Java SDK from 1.8.9.1 to 1.10.12
- Apache Commons Cli from 1.2 to 1.3.1
- Apache Commons Compress from 1.5 to 1.10
- Apache Commons Lang from 3.3.2 to 3.4
- Apache Commons Math3 from 3.3 to 3.5
- Apache Commons Pool from 2.3 to 2.4.2
- Apache Deltaspike from 1.3.0 to 1.5.0
- Apache DS from 2.0.0-M19 to 2.0.0-M20
- Apache HttpComponents Core from 4.3.3 to 4.4.1
- Apache HttpComponents Client from 4.3.6 to 4.4.1
- Apache JClouds from 1.8.1 to 1.9.1
- Apache Lucene from 4.10.2 to 4.10.4
- Apache Olingo Odata from 2.0.0 to 2.0.4
- Apache Solr from 4.10.2 to 4.10.4
- Apns from 0.2.3 to 1.0.0.Beta6
- AspectJ from 1.6.2 to 1.8.6
- Async-http-client from 1.9.8 to 1.9.30
- Atmosphere from 2.2.6 to 2.3.5
- Cassandra from 2.1.2 to 2.1.8
- Cassandra Datastax Java Driver 2.1.2 to 2.1.7.1
- Cglib from 2.2 to 3.1
- Chunk Templates from 2.6.3 to 3.0.1
- Classmate from 1.0.0 to 1.3.0
- CMIS 0.8.0 to 0.13.0
- DNSJava from 2.1.6 to 2.1.7
- Docker-java from 0.10.5 to 1.3.0
- Dropbox core Java SDK from 1.7.6 to 1.8.1
- Dropwizard Metrics from 3.1.0 to 3.1.2
- Ehcache from 2.9.1 to 2.10.0
- Elasticsearch from 1.4.4 to 1.7.1
- Expression Language from 2.2.4 to 2.2.5
- Facebook4j from 1.1.12 to 2.3.0
- Felix Config Admin from 1.8.0 to 1.8.8
- Felix SCR annotations from 1.9.8 to 1.9.12
- Freemarker from 2.3.21 to 2.3.23
- Flapdoodle Embedded MongoDB from 1.46.4 to 1.50.0
- Google App Engine from 1.8.3 to 1.9.25
- Gson from 2.3 to 2.4
- Guice from 3.0 to 4.0
- iCal4j from 1.5.0.2 to 1.6.0
- Hazelcast from 3.4.2 to 3.5.1
- Hibernate Validator from 5.0.3.Final to 5.2.2.Final

- Hsqldb from 2.3.2 to 2.3.3
- Infinispan from 6.0.2.Final to 7.2.5.Final
- Jackson2 from 2.4.3 to 2.6.2
- Jackrabbit from 2.10.0 to 2.11.0
- Jboss AS Maven Plugin from 7.4.Final to 7.7.Final
- Jboss Logging from 3.1.4.GA to 3.3.0.Final
- Jboss Marshalling from 1.4.4.Final to 1.4.10.Final
- Jboss Weld from 2.2.4.Final to 2.3.0.Final
- Jgroups from 3.6.1.Final to 3.6.4.Final
- Jibx from 1.2.5 to 1.2.6
- Joda-time from 2.5 to 2.8.1
- Jruby from 1.7.18 to 1.7.21
- Jsoup from 0.1.51 to 0.1.53
- Jsonpath from 1.2.0 to 2.0.0
- Jt400 from 6.7 to 8.6
- Kafka from 0.8.1.1 to 0.8.2.0
- LightCouch from 0.1.3 to 0.1.8
- Minimal JSON from 0.9.1 to 0.9.3
- MongoDB Java Driver from 2.12.4 to 3.0.4
- Mvel2 from 2.2.1.Final to 2.2.6.Final
- OGNL from 3.0.8 to 3.0.11
- OpenEJB from 4.6.0.2 to 4.7.2
- OpenJPA from 2.3.0 to 2.4.0
- OpenShift Java Client 2.5.0.Final to 2.7.0.Final
- Pax Exam from 4.3.0 to 4.6.0
- Pax Logging from 1.8.1 to 1.8.3
- Pax Swissbox Tinybundles from 1.3.1 to 1.3.2
- PDFBox from 1.6.0 to 1.8.10
- Rabbitmq Amqp client from 3.3.4 to 3.5.4
- Restlet from 2.2.x to 2.3.1
- Scala from 2.11.5 to 2.11.7
- Scala Maven Plugin from 3.2.0 to 3.2.2
- Scalatest from 2.2.0 to 2.2.5
- Shade Maven Plugin from 2.1 to 2.4
- SLF4J from 1.7.10 to 1.7.12
- Snappy from 1.1.0.1 to 1.1.2
- Snakeyaml from 1.14 to 1.16
- Spring upgrades
- Sshd from 0.13.0 to 0.14.0
- TestNG from 6.8.8 to 6.8.21
- Twitter4J from 4.0.2 to 4.0.4
- Univocity Parsers from 1.2.1 to 1.5.6
- VertX from 2.2.5 to 3.0.0
- Xmlunit from 1.5 to 1.6
- XStream from 1.4.7 to 1.4.8
- ZXing from 3.0.1 to 3.2.1

Important changes to consider when upgrading

- The `dumpRoutesAsXml` operation now preserve property placeholder used in the route models.
- Using `setFaultBody` / `setFaultHeader` behaves similar to `setBody` / `setHeader` to preserve existing headers/attachments by setting on existing IN or OUT message.
- If using concurrent consumer on [JMS](#) endpoints for request/reply over JMS then you must use the new `replyToConcurrentConsumers`, `replyToMaxConcurrentConsumers` options to configure the values.
- When the [Aggregator2](#) is forced completed the exchange property `Exchange.AGGREGATED_COMPLETED_BY` value has been changed from "forceCompletion" to "force" so its named like the other completion triggers.
- Removed unsupported modules camel-web and camel-web-standalone.
- Removed unsupported camel:dot functionality from camel:run plugin.
- Removed unsupported camel-archetype-scala-component from maven archetypes.
- The Maven coordinate for linkedin and olingo2 components changed to be aligned like all the others having groupId as just org.apache.camel.
- If using MongoDB component, the option `invokeGetLastError` doesn't exist anymore. The option `invokeGetLastError` instructs the MongoDB Java driver to invoke `getLastError()` after every call. In the 3.0.0 MongoDB driver implementation, the `getLastError()` method in `WriteResult` has been removed. So, if needed, use an acknowledged `WriteConcern` when executing the write operation and then verify the correctness of the operation with the method `wasAcknowledged()` of `WriteResult`.
- The [Jing](#) component now uses `jing` as schema name in uris, instead of `rng` or `rnc`. Those names has been removed.
- [Swagger](#) module now supports getting api-docs from multiple camel contexts in the JVM. The entry point at `/api-docs` now lists the contexts detected, and you need to append the context id in the path, eg `/api-docs/myCamel`
- If using `<contextScan>` with Spring or Blueprint to filter `RouteBuilder` classes, then Camel will now by default only look for singleton beans. You can turn on the old behavior to include prototype scoped with the new option `includeNonSingletons`
- camel-vertx has been upgraded to vertx 3.0 which requires Java 8 at runtime.
- camel-cdi is now using CDI 1.1 api - support for 1.0 has been dropped.
- [Content Enricher](#) with `enrich` and `pollEnrich` now supports dynamic endpoint uris computed using an [Expression](#) that allows to use values from the current [Exchange](#) so you for example can poll a file which name is computed from data from the [Exchange](#) etc. This change breaks the XML DSL a little bit and there is an easy migration effort. The Java DSL stays backwards compatible.

- [WireTap](#) now supports dynamic endpoint uris computed using an [Expression](#) that allows to use values from the current [Exchange](#) so you for example can tap to endpoints based on the content from the [Exchange](#) etc. This change breaks the XML DSL a little bit and there is an easy migration effort. The Java DSL stays backwards compatible.
- Support for CXF 2.7 is dropped.
- camel-jetty8 is being considered deprecated and we would like to drop this in a nearby future release as Jetty 8.x is EOL.
- If you have explicit configured the JMX statistics level to All then that option is now called Default.
- The HTTP based consumers no longer include Camel headers in the responses by default (they already did this for the producers to not include Camel headers when calling a HTTP service).
- [Bindy](#) requires to be configured using class names instead of package names, as it now supports having multiple model classes in the same java packages now.
- [Using CamelProxy](#) now binds the method parameters to the message body/header using Camel annotations to define the binding rules. If no annotations is defined the the parameter is assumed the message body. You can turn this off to have the old behavior.
- [SJMS](#) component has been aligned to bind between Camel Message and JMS Message in the same way as [JMS](#) component does that, in the process a few APIs and behavior has been changed.
- `DefaultExchangeHolder` now only keep primitive / String type headers / exchange properties (like JMS component) and filter out other types such as java instances (caught exception on exchange property is kept as well).
- The Scala based [Swagger](#) (camel-swagger) is deprecated in favor of the new camel-swagger-java component.
- The camel-ruby feature has been removed, use camel-script-jruby instead.
- The camel-jetty component is upgraded to Jetty 9, use camel-jetty8 if you need to use Jetty 8.
- camel-jetty8 is deprecated as Jetty 8 is EOL.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.16.0.zip	apache-camel-2.16.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.16.0.tar.gz	apache-camel-2.16.0.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-2.16.0-src.zip	apache-camel-2.16.0-src.zip.asc
Source for Unix/Linux/Cygwin	apache-camel-2.16.0-src.tar.gz	apache-camel-2.16.0-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>2.16.0</version>
</dependency>
```

Changelog

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

- [release notes for 2.16.0](#)

