Camel 2.11.0 Release

Camel 2.11.0 release

New and Noteworthy

Welcome to the 2.11.0 release with approximately 679 issues resolved - including new features, improvements, and bug fixes, such as:

- Camel RX provides a typesafe and composable API for working with asynchronous events from camel endpoints using Reactive Extensions using the RxJava library
- Added Binding support, so it is easy to combine things like a Data Format to an Endpoint for easier composition of routes.
- Added support for SOAP 1.2 in SOAP data format.
- Cache operation for add/update now supports expiry headers to control time to live/idle/eternal.
- Added `allowNullBody` option to JMS to configure whether sending messages with no body is allowed.
- Added `connectOnStartup` option to HDFS to allow to connect on demand, to avoid having Hadoop block for long time connecting to the HDFS cluster, as it has a hardcoded 15 minute retry mechanism.
- Added support for daily and weekly trends to Twitter component.
- The Camel Maven Archetypes now generates projects without any license headers.
- Added `rejectOld` option to the Resequencer to prevent out of order messages from being delivered after capacity/timeout events occur
- Further optimized XPath under concurrent load, and as well ensured resources are cleaned up eagerly
- Added options `allowNullBody` and `readLockMinLength` to the File and FTP components.
- Made `changed` read lock strategy on FTP go faster (eg when the FTP server has a lot of files in the directory) if you enable the `fastExistsCheck=true` option as well. Notice that some FTP server may not support this.
- HL7 moves to HAPI 2.0 and supports using a dedicated Parser instance in the HL7 MLLP codec and DataFormat. Added "Terser" language and expression to be able to extract fields from a parsed message. HL7 now uses Apache Mina 2.x.
- Add an option `HttpMethodRestrict` to restrict HTTP method in Jetty and SERVLET
- Add support for selection of Direct-VM consumers by using ant-like path expression.
- The POJO Producing, and POJO Consuming with @Consume, @Produce, @EndpointInject now supports a new `property` attribute to get the endpoint configuration from a bean property (eg using a getter method); this allows you to configure this on the bean using conventional bean configuration.
- Testing with `camel-test-blueprint` on Windows no longer tries to cleanup after testing taking up 5 seconds and logging WARNs.
- The File, and FTP components now support `fileExist=Move` option to move any existing files before writing a file.
- Added option `loadStatisticsEnabled` on Camel JMX to allow to disable load statistics if not needed (avoids a background thread being in use, to calculate the load stats).
- Enabled "lazy connections" for XMPP providers via the `testConnectionOnStartup` option
- Added a connection monitor to detect and fix dropped XMPP consumer connections at configurable `connectionPollDelay` intervals
- Added an `org.apache.camel.builder.ExchangeBuilder` to build the Exchange using a builder pattern.
- The Camel Run Maven Goal can now run CDI applications.
- The Camel CDI component has improved a lot.
- Added option `allowRedeliveryWhileStopping` to error handlers to control if redelivery is allowed during stopping/shutting down Camel or the route(s). Turning this option `false` allows to stop quicker by rejecting redelivery attempts.
- Added support for specifying user info in Camel Endpoint urls, which contains the @ sign; now the @ sign can be given as is; without being encoded to %40.
- Added robust connection support for JMX. Optional `testConnectionOnStartup` allows a JMX consumer to attach to a JMX server that becomes available *after* the JMX endpoint starts; `reconnectOnConnectionFailure` enables re-connection of failed JMX connections.
- JAXB and SOAP data format now supports controlling namespace prefix mappings when marshalling (eg to avoid prefixes such as ns2, ns3, ns4 etc.)
- Added support for using raw uris when Components create Endpoints. This gives component writers full power in case their component has special syntax/notation for endpoint uri configurations.
- EIPs configured with custom `AggregationStrategy` will now strict check the custom strategy exists in the Registry and no longer fallback and use a default strategy if not.
- Camel now logs on shutdown if any thread pools hasn't been properly shutdown, and will shutdown them as fail safe (can only be done for the thread pools Camel manages).
- The buffer size of Stream caching is now configurable.
- Improved startup performance.
- Added JMS URI option `messageListenerContainerFactoryRef` which allows you to specify a `MessageListenerContainerFactory` to determine what `org.springframework.jms.listener.AbstractMessageListenerContainer` to use to consume messages.
- MQTT producer now leverages the Asynchronous Routing Engine

- The Netty producer scales up and down much better by leveraging a channel pool.
- Added new Camel Maven Archetypes for creating Camel and Groovy applications.
- Added a Groovy DSL that allows to use Groovy closures and Groovy XML parsing in routes.
- Camel now injects `CamelContext` on `Language` and `DataFormat` that implements `CamelContextAware`
- Thread pools in Camel with `maximumQueueSize` of 0 or negative will now not use a worker queue but do direct hand-over to thread instead.
- Simple unary operators for inc and dec must now have function on left hand side, to be parsed as unary function. Otherwise its now regarded as literal.
- Custom assigned id's on Camel Routes is now validated on starting the routes, that they must be unique within the same CamelContext.
- Added `AbstractListAggregationStrategy` abstract class to make it easier to aggregate into a List<V> using a custom `AggregationStrategy`.
- EIPs which support using `AggregationStrategy` now control the lifecycle of the strategy; allowing end users to do custom logic in the start /stop methods of their strategy implementation.
- Added option `includeSentJMSMessageID` to JMS to allow to enrich Camel Message with the actual JMSMessageID for the sent message. Can be used by end users for logging purpose etc.
- Added option `allowShortLines` and `ignoreExtraColumns` to Flatpack data format.
- Mail consumer now supports using search terms to filter mails by terms such as subject, from, body, sent date etc.
- The cached script file in the Language component can now be cleared via JMX.
- improved the karaf context-info command by adding a verbose mode to view endpoints list (now hidden by default) and adding stats on the number of active/inactive routes
- Added support for `UpdateList`, `DeleteList` and `ExecutorType` to MyBatis component.
- Added support for easier configuration of using concurrent message listener consumer when doing request/reply over JMS, by leveraging the existing `concurrentConsumers` and `maxConcurrentConsumers` options.
- Improved the SMPP consumer to expose more data by receiving a `DeliverSM` message, including the optional parameters.
- Camels property placeholders can now be used in route id / node id's in the routes in XML and Java DSLs, eg <route id="coolName">
- The thread name pattern can easily be configured on <camelContext> with the new threadNamePattern attribute
- Camel now validates SEDA and VM endpoints using same queue name, that if any queue size is in use, that size must match. As well Camel logs at INFO level when starting up the queues and the sizes. This allows end users to easily spot any issues if mistakenly using different queue sizes for the same queue names.
- Added JMX operations to throughput logger so you can get stats and also reset the stats at runtime.
- Added ACL support on the AWS-S3 component.
- Allow to configure Jetty http client thread pool settings in uri to make it easier to have different settings per endpoint.
- Added support for implied decimal formats to Bindy
- Added option `backlog` to Netty which can be used to allow more concurrent clients to connect to a Netty consumer.
- Added support for parsing CSV if a field has a ' (single quote) at it's beging
- Added support for using custom url rewrite plugins in HTTP, HTTP4, and Jetty producers, which gives more control of url mappings, such as when bridging/proxying http based routes. Provided `camel-urlrewrite` as out of the box implementation.
- Improved the Spring-Integration consumer to also send back the message headers.
- The Camel Guice component has been upgraded to latest Guice 3.0, and no longer dependent on Guice extension project `guiceyfruit`.
- ReplyTo Temporary Queue now refreshed when the underlying JMS Connection fails.
- When processing an Exchange fails, the failed route id is available for end users. See details at Dead Letter Channel.
- When using XSLT includes will use same file or classpath loading as endpoint configuration, if not explicit configured.
- Now Camel Transport for CXF supports to configure with blueprint.
- SQL Component now supports named parameters in the queries.
- SQL Component now supports Batch Consumer to consume data; for example to pickup new rows being inserted into a table. As well having queries being execute after each row has been processed, such as deleting/marking the row as processed.
- JDBC based aggregation repository now allows to store message body and headers as text in individual columns. This makes the data human readable, as by default the data is stored as binary.
- File consumer will on first poll run the check to delete old orphaned lock files. As opposed to before where this happened during startup of Camel itself; which could delay starting Camel if the delete process takes a long time.
- File and FTP can now use a custom idempotent key with the built-in idempotent consumer. For example you can now use both the file name and size as key.
- File and FTP producers support the `CamelOverruleFileName` header as a use once and discard header to overrule and use as filename for writing; while preserving the original filename in the `CamelFileName` header.
- The Languages and Language component can now load the script from external resource using `"resource:scheme:location"` syntax.
- HDFS producer without a split strategy will now close file stream after writing. You can use the `CamelHdfsClose` header to explicit to control if to close file stream or not.
- Simple language now has `type` function to refer to a type, enum or constant field.
- Using `allowRedeliveryWhileStopping=false` on Dead Letter Channel now moves the message to the DLC (when stopping) instead of rejecting the message.
- Unit testing with Blueprint Testing is now easier to override Blueprint property placeholder values from both external .cfg files, as well from placeholders defined directly in the XML file. See further details at Using PropertyPlaceholder page.
- Bindy FixedLengthRecord is improved with support for header and footer records, delimited fields, and field lengths defined within a record.
- Added `transacted` option to Hazelcast SEDA consumer to use Hazelcast transaction.
- Improved performance of Simple and Bean language when using OGNL expression, by leveraging an internal cache to avoid introspecting the same types over and over again.
- Camel now validates better when using Try Catch Finally in the routes has been configured properly.
- Added Karaf commands for using the BacklogTracer when running Camel application in a Apache Karaf container.
- The VM component can now safely communicate Camel Contexts residing in different bundles in an OSGi environment, as the issue leading to an incoherent state when restarting one of the bundles has been fixed.
- Made it easy to use Saxon as the transformer with the XSLT component.
- Improved AWS to not only support the Amazon provided clients for accessing the services.* Improved AWS to not only support the Amazon provided clients for accessing the services.
- Added multiple consumers support to Guava EventBus endpoints.
- Added support for listener interfaces to Guava EventBus in order to provide better `DeadEvent` handling.
- Made it easier to configure options such as passwords on Camel endpoint URIs and use the value *as is*, using the new RAW token. See more details at How Do I Configure Endpoints?
- Improved Flatpack and Flatpack DataFormat to allow short lines and ignore extra columns.
- Fully customize the output of the Log component by providing a custom `ExchangeFormatter`.
- You can now configure more options on GSON Data format, for example field mapping strategies.

- Simplified consuming REST requests with CXFRS with the *Simple Binding Style*.
- Improve JAXB to support validation by marshaling/unmarshaling objects from/to XML.
- Improved `org.apache.camel.spring.spi.BridgePropertyPlaceholderConfigurer` to support both Camel and Spring property placeholder syntax in the .properties files.
- New Karaf commands for using the BacklogTracer as well for profile routes.

Fixed issues

- Fixed RoutePolicy used from Spring XML with `routePolicyRef` may not invoke all callbacks on the policy.
- Fixed MINA consumer to force disconnect upon timeout when doing request/reply, as late replies may cause side effects otherwise.
- Fixed Simple language with empty quoted literals in predicates, so you can compare against empty strings.
- Fixed SOAP data format to set SOAP 1.1 namespace when marhsalling faults.
- Fixed SEDA producer to send to the right BlockingQueue being attached on the SEDA endpoint so that the sent messages don't get lost.
- Fixed using `sendEmptyMessageWhenIdle=true` on a batch consumer such as File now working with Aggregate EIP when using `completion FromBatchConsumer`
- Fixed `MockEndpoint.expectedBodiesReceivedInAnyOrder` to allow asserting on multiple arrival of the same/equal bodies.
- Fixed MyBatis to demarcate the running transactions properly while reading/writing from/to database.
- Fixed Jetty, HTTP, HTTP4, and AHC http components not working in bridge mode if content path have spaces
- Fixed the CXF fallback convertor issue which camel will unregister it when the converter return null.
- Fixed the issue that File2 endpoints on windows without volume name in the path url.
- Fixed the issue of unmarshaling pgp messages encrypted with ElGamal.
- Fixed random Load Balancer to be even spread (before was a little less likely to pick the 1st)
- Fixed Timer to not fire during starting CamelContext
- Fixed XMLSecurity DataFormat to enable retrieval of password-protected private keys for asymmetric decryption
- Fixed Exception Clause using `maximumRedeliveries(-1)` may be wrongly inherited by another `onException` during a redelivery process.
- Fixed logging Error Handler to use configured logging level.
- Fixed issue when using multiple RouteBuilder with the Java DSL should have the Exception Clause being scoped per RouteBuilder instance.
- Fixed issue with restarting a SEDA consumer would not stop previous used thread pool.
- Fixed issue when running low on memory and doing request/reply over JMS may GC producer, causing a new producer to be created, and leaving a message listener running from the previous producer.
- Fixed issue with Try Catch Finally should clear redelivery exhaust state when catching exception in doCatch.
- Fixed potential NPE in MQTT producer, if a connection has been disconnected, and sending fails.
- Unhandled exceptions occurred during routing from SMPP routes is now propagated back to the SMPP library to deal with.
- Fixed Restlet parsing of last modification date header to use RFC-2822 date pattern.
- Fixed Aggregator EIP when using completion size expression and fixed size together not working as expected.
- Fixed Crypto to close streams after usage, which otherwise could prevent deleting files on Windows, which tend to lock files if a stream is open.
- Fixed issue with Groovy language in OSGi
- Fixed issue with localWorkDirectory option on FTP not using the optimization to try to rename the work file when routing to a File endpoint.
- Fixed SMPP to use the right data coding by using Octet unspecified (8-bit binary) and using the specified alphabet from the exchange, if provided.
- Fixed issue with configuring Jetty component using a name other than `"jetty"`.
- Fixed issue with request/reply over JMS where a route is used on "the other end" to process the request and send back the correlated reply message; where a Camel route is used which has only 1 output, which can lead to no reply message being sent.
- Fixed regression in AWS-SQS by updating the existing queue without any attributes.
- Fixed issue with Using PropertyPlaceholder in Expressions in routes which are using <transacted> or <policy>.
- Fixed issue with Splitter which swallow the IOException catched by the underlying Scanner.
- Fixed issue with Splitter where tokenizeXML would not split XML messages containing nodes with namespace prefixes.
- Fixed issue when using Jetty component would create a http client having a live thread pool, which is only needed if using producers of Jetty. Also improved handling lifecycle of clients to be per producer by default instead of shared, which otherwise can cause issues.
- Fixed issue with File and FTP consumers when using `doneFileName` then avoid picking up files in middle of group if done file is written during scanning.
- Fixed issue with File and FTP consumers when using `doneFileName` may produce too many WARN logs about cannot delete doneFileName.
- Fixed issue with Temporary Queue ReplyTo destination being recreated every time a new multiple concurrent consumer was initialised (JMS component).
- Fixed issue with XSLT was not using custom assigned `UriResolver` to load the initial xslt resource, but only for other included resources.
- Fixed issue when Using PropertyPlaceholder with OSGi blueprint and using Jasypt for decrypting placeholders at the same time.
- Fixed JMX component to work in OSGi if output is XML.
- Fixed Aggregator to not check for timeout if completion size expression was in use, and not completed yet.
- Fixed issue with possible to have negative in-flight counts when using Content Enricher, Bean or annotation DSL under certain conditions.
- Fixed issue with Quartz consumer may not work if being configured to trigger with days apart causing its endpoint to be evicted from Camel's endpoint cache.
- Fixed issue with Karaf then Camel applications will not properly restart, if using blueprint-cm to reload bundle due .cfg file change.
- Fixed a race condition in Aggregate eip when recovery is enabled and a in progress exchange was completed after the recovery scan was executed, and before the recover task kicks in.
- Fixed using endpoint reference options in URIs (eg #beanName) to use the class type of the setter method to support looking up beans from the OSGI Service Registry.
- Fixed Simple language when using `bean` function to support specifying parameter values containing comma not being parsed into multiple parameter values.
- Fixed Aggregate eip when using `completionFromBatchConsumer` combined with other completions.
- Fixed pairs of VM producer-consumer disconnecting when the containing bundle was restarted in an OSGi environment.
- Fixed service shutdown logic which could potentially execute several times, putting the system in an incoherent state.
- Fixed so the `preMove` option on File and FTP consumers is now run after the file read lock has been acquired.

New Enterprise Integration Patterns

- ControlBus EIP

New Camel Karaf Command

New **Components**

- ControlBus Component - for easy management using a component supporting the ControlBus EIP
- `camel-aws` - New component to AWS for sending metrics to Amazon CloudWatch from Camel.
- `camel-cmis` - for integration of CMIS compliant content repositories
- `camel-couchdb` - for integration with Apache CouchDB
- `camel-elasticsearch` - for ElasticSearch integration
- `camel-rx` - provides a typesafe and composable API for working with asynchronous events from camel endpoints using Reactive Extensions using the RxJava library
- `camel-servletlistener` - for bootstrapping Camel in web applications, without the need for Spring Framework etc.
- `camel-sjms` - for "Springless" JMS integration
- `camel-spring-ldap` - for LDAP integration using the Spring LDAP template offering more functionality than existing LDAP component.
- `camel-spring-redis` - for talking to Redis key-value store.
- `camel-urlrewrite` - for bridging http endpoints and using the UrlRewriteFilter project to control url mappings.
- `camel-xmlrpc` - for talking to xmlrpc server from Camel.

DSL Changes

- Added LoggingErrorHandler fluent builder to Scala DSL.

New Annotations

New **Data Formats**

- `camel-base64` - support base64 encoding and decoding.
- `camel-xmlrpc` - support to marshal and unmarshal the request and response of xmlrpc.
- `camel-zipfile` - support to Zip files containing a single entry.

New **Languages**

- VTD-XML - for fast and efficient XPath evaluation/splitting using the VTD-XML Library. `camel-vtdxml` hosted at Camel Extra

New **Examples**

- ActiveMQ Camel Tomcat example shows how to embed Apache ActiveMQ and Camel in a web application running on Apache Tomcat.
- FTP Example shows how to use Camel to upload or download files to/from an existing FTP server.
- Hibernate Example shows how to work with Hibernate to exchange data using a shared database table.
- Servlet Tomcat No Spring Example example shows how to run Camel as a web application in any web container such as Apache Tomcat, without the need for Spring Framework.
- SQL Example shows how to work with SQL to exchange data using a shared database table.

New **Tutorials**

API changes/breaking

- HL7 upgraded to now use Apache Mina 2.x.
- Renamed `org.apache.camel.component.hbase.HBaseContats` to `org.apache.camel.component.hbase.HBaseConstants`
- Added getter/setter methods for loadStatisticsEnabled on `org.apache.camel.spi.ManagementStrategy`
- Added `shutdownGraceful`, and `awaitTermination` methods to `ExecutorServiceManager` to make it easier to shutdown a thread pool gracefully waiting for its tasks to complete.
- Added method `useRawUri()` to `org.apache.camel.Component` to allow Components to use raw uri when creating endpoints. (by default the uri has been encoded prior to creation).
- Added method `newThread` to `org.apache.camel.spi.ExecutorServiceManager` to create a new thread without using a thread pool.
- Added support to parse CSV if a field has a ' (single quote) at it's beging, this made camel-csv doesn't support the field which is quoted with ' (single quote) or "(double quote) at the same time.

Internal/Non API refactorings that may affect users

- The Camel Test Kit provides a default Properties component out of the box; which mean using `addComponent("properties", ...)` is no longer needed; instead get the component using `getComponent("properties")` and configure the component instead.
- Minor adjustments to `org.apache.camel.builder.ErrorHandlerBuilder`.
- `LRUCache` will attempt to stop service on eviction of entry.
- Removed `org.apache.camel.model.NodeFactory` as its no longer applicable.
- Added `getCamelContext()` methods to `ProducerTemplate`, and `ConsumerTemplate`
- The method `isMatched` on `GenericFileConsumer` is now abstract and has an extra parameter.
- The Camel Test Kit removed the support of JUnit3, please use JUnit4 version instead.
- The method `prepareOnStartup` on `org.apache.camel.component.file.GenericFileProcessStrategy` is now executed on first poll instead of starting the route; as if the job takes a long time, it will block Camel from starting other routes. If the job takes more than 1 second to run then its logged at INFO level so people can easier spot if the task takes a longer time, than usual.

- [HDFS](#) producer without a split strategy will now close file stream after writing. You can use the `CamelHdfsClose` header to explicit to control if to close file stream or not.
- The type for file last modified header is changed from `java.util.Date` to `long`
- Removed class `org.apache.camel.component.gae.context.GaeSpringCamelContext` from `camel-gae`
- Using `allowRedeliveryWhileStopping=false` on [Dead Letter Channel](#) now moves the message to the DLC (when stopping) instead of rejecting the message.
- When using [Spring](#) with Camel, then Camel will now lookup in Spring's ancestor application contexts as well, when using the [Registry](#)
- The JMX MBean name for the [Tracer](#) has removed the identity hash code, eg `org.apache.camel:context=localhost/camel-1,type=tracer,name=Tracer(0x548bb8b2)` is now named `org.apache.camel:context=localhost/camel-1,type=tracer,name=Tracer`.
- The [XSLT](#) component will now fail if the {{transformerFactoryClass} cannot be loaded on the classpath, instead of logging a WARN.
- The [SQL](#) producer will only access the message body if there is query parameters to prepare in the SQL to execute (eg using ? or :name).
- The `preMove` option on [File](#) and [FTP](#) consumers is now run after the file read lock has been acquired.

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 (e.g. running `"mvn site"`). There is no plan to fix this, as the project does not use maven site.
- JBoss 7 may log a WARNing when running Camel as a WAR file.
- [CXF Example OSGi](#) doesn't work without enabling dynamic imports for this bundle.
- The [FTP](#) component may have issues for Windows clients or servers in the 2.11.0 release. Downgrading to use commons-net 3.1 may fix this, see [CAMEL-6219](#)

Dependency Upgrades

- Aries Blueprint 0.3 to 1.0.1
- Async Http Client 1.7.5 to 1.7.13
- Avro 1.6.2 to 1.7.3
- AWS 1.3.10 to 1.3.27
- BeanIO 2.0.0 to 2.0.5
- Bouncycastle 1.46 to 1.47
- Commons CSV 1.0-r706900_3 to 1.0-r706899_5
- Commons Exec 1.0.1 to 1.1
- Commons Logging 1.1.1 to 1.1.2
- Commons Net 3.1 to 3.2
- ConcurrentLinkedHashMap 1.2 to 1.3.2
- Castor 1.3.1 to 1.3.2
- CXF 2.6.5 to 2.7.4
- Dozer 5.3.2 to 5.4.0
- Ehcache 2.5.1 to 2.5.2
- Google App Engine 1.6.6 to 1.7.4
- Google Guava 13.0.1 to 14.0.1
- Groovy 1.8.6 to 2.1.3
- GSon 2.1 to 2.2.2
- Guice 2.0 to 3.0
- Hadoop 1.0.4 to 1.1.1
- Hazelcast 2.0.2 to 2.5
- Hibernate 4.1.9 to 4.1.11
- HBase 0.90.5 to 0.94.3
- HTTP Client 4.1.3 to 4.2.3
- HTTP Core 4.1.4 to 4.2.3
- Icu4j 4.0.1 to 4.8.1.1
- Jackson 1.9.7 to 2.1.14
- Jaxen 1.1.3 to 1.1.4
- JClouds 1.4.0 to 1.5.7
- Jettison 1.3.1 to 1.3.3
- Jetty 7.5.4 to 7.6.8
- JRuby 1.6.7 to 1.7.2
- JT400 6.0 to 6.7
- Jython 2.2.1 to 2.5.3
- Krati 0.4.5 to 0.4.8
- LevelDBJNI 1.2 to 1.6
- Lucene 3.6.0 to 3.6.1
- Mina 2.0.4 to 2.0.7
- MongoDB Java Driver 2.7.3 to 2.9.1
- MQTTClient 1.2 to 1.4
- MyBatis 3.1.1 to 3.2.2
- Netty 3.5.1 to 3.6.5
- Ognl bundle 3.0.4_1 to 3.0.5_1
- OSGi 4.2.0 to 4.3.1
- Pax Logging 1.5.3 to 1.6.10
- QPid 0.16 to 0.20
- Quartz 1.8.5 to 1.8.6
- Quickfix 1.5.2 to 1.5.3
- Restlet 2.0.14 to 2.0.15
- Saxon 9.3.0.11 to 9.4.0.1
- Scala 2.9.1 to 2.10.1

- ServiceMix Specs 1.9.0 to 2.2.0
- Shiro 1.2.0 to 1.2.1
- SLF4J 1.6.6 to 1.7.5
- Solr 3.6.0 to 3.6.1
- Spring Batch 2.1.8 to 2.1.9
- Spring Framework 3.1.1 to 3.1.4
- Spring Integration 2.1.2 to 2.2.3
- Spring Security 3.1.0 to 3.1.3
- SSHD 0.6.0 to 0.8.0
- StringTemplate 3.2.1 to 4.0.2
- TestNG 6.0.1 to 6.8
- Twitter4j 2.2.5 to 3.0.3
- Woodstox 4.1.2 to 4.2.0
- Xbean-Spring 3.12 to 3.13
- Xerces 2.9.1 to 2.10.0
- XmlBeans 2.5.0 to 2.6.0
- XStream 1.4.2 to 1.4.3
- Zookeeper 3.3.5 to 3.4.5

Important changes to consider when upgrading

- Spring 3.0.x support is @deprecated and some of the camel-spring components require Spring 3.1 onwards.
- Apache Camel Karaf features now uses Spring version range "[3.1,4)" as Spring 3.1 is now required as minimum Spring version.
- `camel-mina2` now uses `OrderedThreadPoolExecutor` by default. There is a new option `orderedThreadPoolExecutor` you can configure it to `false` to revert back to unordered if needed. If using SSL etc. then ordered is required to be used.
- `camel-netty` now uses `OrderedMemoryAwareThreadPoolExecutor` by default. There is a new option `orderedThreadPoolExecutor` you can configure it to false to revert back to unordered if needed. If using SSL etc. then ordered is required to be used.
- The default value of the `delay` option on Timer has been changed from `0` to `1000`
- The default value of the `convertLFtoCR` option of the HL7 MLLP codec has been changed from `true` to `false`. Likewise, the HL7 DataFormat and TypeConverter do not perform this conversion anymore.
- The Simple language has limited its support for the escape character to only new line, carriage return and tab characters. This makes it easier to use back-slash characters in the Simple language without double escaping etc.
- `camel-netty` now requires `commons-pool` as dependency.
- The `maximumQueueSize` of Thread Pools and Thread Pool Profiles with values 0 or negative now means disable the worker queue, and use direct-handover. If you want an unbounded queue size, then set the `maximumQueueSize` to `Integer.MAX_VALUE` which was what the previous code did.
- Custom assigned id's on Camel Routes is now validated on starting the routes, that they must be unique within the same CamelContext.
- `camel-blueprint` now requires Apache Aries 1.0, and OSGi 4.3.0.
- Removed @deprecated classes form `camel-spring` in the `org.apache.camel.component` package which was causing split-package issue with OSGi. Component authors who are using these @deprecated classes, should migrate to use equivalent classes from `camel-core` in the same package.
- The default Spring version is upgraded from 3.0.7 to 3.1.3 out of the box.
- Camel now validates SEDA and VM endpoints using same queue name, that if any queue size is in use, that size must match. As well Camel logs at INFO level when starting up the queues and the sizes. This allows end users to easily spot any issues if mistakenly using different queue sizes for the same queue names.
- The Zookeeper endpoint `awaitExistence` option has been marked as @deprecated and is removed from it's documentation as it's usage would have no effect in the current codebase.
- Marked the 2 classes in the `junit4` package from `camel-test-spring` component as @deprecated, and refer to use the classes from the `spring` package instead.
- If using XSLT with includes, then mind that if not explicit defined file: or classpath: as prefix in xsl:include, then Camel will now use the same as configured on the endpoint. Before "classpath:" was always used as fallback, despite endpoint could have been configured with "file:".
- `camel-scala` has been upgraded to Scala 2.10.1, there is a `camel-scala_2.9` module that is built against Scala 2.9.2. Though `camel-scala_2.9` is deprecated and expected to be removed in the next bigger release (eg Camel 2.12/3.0).
- `camel-gae` now longer has Spring dependency, and we removed class `org.apache.camel.component.gae.context.GaeSpringCamelContext.`
- `camel-jackson` has been upgraded from Jackson 1.x to 2.x.
- Type Converter utilizations statistics has been disabled by default, due there is a little performance impact under very high concurrent load. You can enable the statistics easily as documented at Type Converter.

Notice

- Maven 3.0.4 or better is required to build the Source
- Apache Karaf 2.3.1 or better is required for deploying Camel 2.11 onwards, when using Apache Karaf.
- Spring 3.1 is now required as minimum Spring version. Spring 3.0 is @deprecated and only partially supported with this Camel release.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.11.0.zip	apache-camel-2.11.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.11.0.tar.gz	apache-camel-2.11.0.tar.gz.asc

Source Distributions

Description	Download Link	PGP Signature file of download
Source Distribution	apache-camel-2.11.0-src.zip	apache-camel-2.11.0-src.zip.asc