

Camel 2.10.0 Release

Camel 2.10.0 release



New and Noteworthy

Welcome to the 2.10.0 release with approximately 483 issues resolved - including new features, improvements, and bug fixes, such as:

- Added support for Java 1.7
- Added consumer to [JCR](#) component
- Internal code cleanup to reduce warnings reported by Eclipse
- Moved [Spring Testing](#) from `camel-test` to `camel-test-spring` component.
- Camel [Property Placeholders](#) are now supported in embedded [Expressions](#) within Camel routes.
- Added missing options in the XML DSL for the [XPath](#) language to control and set which XPath Engine to use, etc.
- Added option `logNamespaces` to the [XPath](#) language to have `INFO` logging of detected namespaces from message. This can be used to troubleshoot issues where an [XPath](#) expression doesn't work as expected (as it's often a namespace conflict).
- Added `replyToCacheLevelName` option to the [JMS](#) component, allowing configuration of cache levels for persistent reply consumers when doing request/reply over JMS. Some message brokers do not work well with caching, so users may need to set this option to `CACHE_NONE`.
- Added `antInclude` and `antExclude` options to [File](#) and [FTP](#) components to make it easier to specify include/exclude filters using ANT syntax.
- The [Camel archetypes](#) for Java and Scala no longer generate projects with Spring dependencies.
- Improved `JdbcMessageIdRepository` in the [SQL](#) to use custom SQL statements/queries.
- Added option to [Properties](#) to allow silently ignoring property locations not found, such as a `.properties` file.
- Added `managementNamePattern` attribute to `<camelContext>` to make it very easy to customize and control the JMX naming of the Camel application. See details at [Camel JMX](#). For example, in OSGi environments you can explicitly configure a `managementNamePattern` in case you do not want the bundleId as part of MBean names.
- Added streaming update support to [Solr](#) as well as support for `SolrInputDocuments`
- Added `SolrJ DirectXMLRequest` support to [Solr](#) to support XML based index updates
- Improved Camel [Mail](#) to support the `replyTo` option
- Added `traceEnabled` option to [Jetty](#) component so you have control over whether to allow HTTP TRACE calls against your Jetty endpoints.
- Polished the [Camel Maven Archetypes](#) to let the projects be prepared for [Testing](#), and fixed any WARNs from Maven.
- Added `dumpRouteStatsAsXml` methods to [JMX](#) to make it easier to get the total performance stats as XML, and in one JMX call.
- [Graceful Shutdown](#) is now more aggressive if timeout occurred during shutdown, rejecting continued processing of messages or redeliveries.
- Add `ShutdownPrepared` SPI to allow custom Services (also EIPs) in routes to do custom code in preparation for shutdown.
- The `camel-archetype-java` [Camel Maven Archetypes](#) is now a pure Java DSL based project (no longer Spring)
- [JMS](#) consumer is less verbose when logging uncaught exceptions thrown while processing a message. Added `errorHandlerLoggingLevel`, and `errorHandlerLogStackTrace` options to tweak the logging verbosity.
- Added `messageId` as an [MDC logging](#) key.
- Improved Camel's Transaction [Error Handler](#) logging to better pinpoint and correlate redelivered messages (from, e.g., a JMS broker), making the logs much easier to read.
- Added a timeout option to [Spring Web Services](#) for specifying the read timeout while invoking a webservice.
- Optimized [Type Converter](#) registry for possible concurrency contention when looking up a type converter.
- Improved [Simple](#) language to work better with floating point numbers in [Predicates](#)
- Camel is now less verbose when starting and stopping, with less being logged at `INFO` level.
- Add support for keyed data queues to [JT400](#) component.
- [Idempotent Consumer](#) is now enlisted in JMX, which allows monitoring the duplicate message count as well as the operation to reset it.
- Thread pools registered in [JMX](#) are now unregistered when Camel shuts them down. Note that all MBeans will be unregistered from JMX when Camel shuts itself down. This improvement is to unregister thread pools more eagerly, such as when routes are removed at runtime.
- Camel will now also copy the message headers for a `JmsMessage` if you leverage `useOriginalMessage` in an `onException` definition.
- Added `mockEndpointsAndSkip` to [AdviceWith](#) and the [Test Kit](#)
- The [File](#) and [FTP](#) components now support filtering directories. The `isDirectory` method on the `GenericFile` class can be used to skip unwanted directories, accelerating file consumption.
- [SMPP](#) connections can now be tunneled through an HTTP proxy by configuring HTTP proxy settings.
- Camel now also checks, when starting a route dynamically, whether there would be any violations on multiple consumers from the same [Endpoint](#). Previously this check only occurred during Camel startup - now it always happens.
- Stopping a `ConsumerTemplate` or a `ProducerTemplate` will now stop and shutdown its cache to cleanup any resources which otherwise may not have been reclaimed until shutdown.
- [ScheduledPollConsumer](#) will now shutdown thread pools more eagerly, such as when stopping a `ConsumerTemplate`.

- Added option `asyncStartListener` to **JMS** to allow starting JMS consumers (e.g. routes) asynchronously. This can be used to avoid blocking in case the JMS consumer takes long time to establish a connection to the JMS broker.
- **Wire Tap** now supports **Error Handling** when sending a tapped message to an **Endpoint**.
- Added the `autogenColumns`, `configRef` and `strategyRef` options to the **CSV** dataformat in XML DSL whereas the new `skipFirstLine` option is provided in Java API as well.
- **Validator** component will now fail with a better error message if resource schema refers to element which cannot be resolved.
- Added support for retrieving generated keys from **JDBC** component.
- Added support for Spring 3.1.x.
- Added `retainFirst`, and `retainLast` options to **Mock** component.
- Improved **Simple** language to allow referring to headers using bracket notation.
- Added support for the `scp`: protocol via **Jsch**.
- Added support to exclude certain fields when marshalling POJO's to **JSON**. Available with both the Jackson and GSON library.
- Allow to configure frequency of timeout checker when doing request/reply over **JMS**
- Improved the Java DSL to make it easier to use predicates from the build-in DSL, that has commonly used predicates and expressions.
- **Aggregate** EIP now supports `TimeoutAwareAggregationStrategy` aggregation strategies.
- The default **Injector** now performs bean post processing to inject `@Produce`, `@Consume`, `@EndpointInject` fields etc.
- The `camel-run` maven plugin now supports running **OSGi Blueprint applications**.
- Added utilization statistics to the type converter registry (which can be accessed from **JMX**), and are logged as Camel is shutting down.
- Improved **JMX** to work better with servers such as WebSphere which alters MBean names when enlisting mbeans into the server (Camel now have an internal mapping table).
- All **EIPs** out of the box from Camel now provides **JMX** utilization statistics.
- The **File** and **FTP** consumers will now let `org.apache.camel.spi.ExceptionHandler` deal with exceptions occurred during acquiring real lock as well. See **File** page for example of using a custom `ExceptionHandler`.
- Reduce the logging level from **ERROR** to **WARN** on the default `LoggingExceptionHandler`.
- Added option `consumer.bridgeErrorHandler` to allow consumers such as **File** and **FTP** to use the routing engine **Error Handler** to deal with exceptions occurring in the consumer (outside the routing engine), for example if having trouble picking up a file from a shared network drive etc. See more details at **File** page. This option applies to all consumers which extends the `DefaultConsumer` from `camel-core`.
- Added `eagerMaxMessagesPerPoll` option to **File**, and **FTP** endpoints to allow to do a full sorting of the files and also limit the pool at the same time
- The **XSLT**, **Velocity**, **FreeMarker**, and **StringTemplate** components using http resource for the template now supports query parameters
- You can specify the class type with **Bindy** to make it stand out what type is being used, as well the `unmarshal` operation returns a list of the pojos (and the pojo directly if only 1). Which makes it easier to use/work with.
- Added `group` option to **Tokenizer** to make it easier with the **Splitter** to split by combining N number of parts together, eg splitting files in chunks by N lines combined.
- Camel is now compatible to the upcoming Karaf 3 release
- Added `ExchangeSendingEvent` to `EventNotifier` to emit an event **before** sending an **Exchange** to an **Endpoint**. Then we have events for both before and after.
- Relaxed syntax check in **Simple** language to only scan for functions if a start token was explicit configured. This allows to easier do json like expressions.
- Added support for escaping values in **Simple** language using the `\` character.
- Added new **Camel Maven Archetypes** for creating new **Component** using Scala.
- Added `trim` option to **Languages** in the XML DSL, to control trimming the expression values, to remove the whitespace noise from XML tags.
- Improved the **File** consumer to support competing consumers on a shared drive to not pickup files from other consumers, as marker file is always used, for `readLock` strategies such as `changed`, `rename` and `fileLock`. For example if you have a clustered environment and have multiple Camel applications pickup files from a shared drive with `readLock=changed`.
- The **File** consumer with `readLock=changed` will now consider zero length file as in progress files.
- Made it easier to **bridge Spring and Camel property placeholders**, by configuring only once, using a new `BridgePropertyPlaceholderConfigurer` bean type.
- Improved the **SMPP** component to refer to a `org.jsmpp.session.SessionStateListener` to receive callbacks when the session state changed.
- Added connection pooling to **JT400** component
- **Restlet** component can be configured with Restlet Server parameters, for example to configure thread pool sizes.
- **CamelContext** MBean in **JMX** now shows aggregated performance stats for all the routes
- Made it easy to override properties when **Testing** with Camel when **Using PropertyPlaceholder**. For example unit testing with OSGi Blueprint using config admin service to provide properties, is now easy as you can provide the properties from directly within the unit test source code itself.
- Improved **Netty** to support both shareable and non-shareable encoders and decoders.
- Improved **Netty** to support proxy use cases (eg `from(netty).to(netty)`)
- Added `CompletionAwareAggregationStrategy` to **Aggregator** EIP as callback when the aggregated exchange is completed.
- Scheduled **Polling Consumer** and poll based **Components** such as **File**, and **FTP** can now use a custom/shared thread pool for its consumer(s). For example having many file routes, can now share a thread pool, and reduce number of overall threads in the JVM.
- Optimized **Bean** expressions to cache bean introspections so subsequent evaluations is faster.

Fixed issues

- Fixed memory leak when dynamically adding and removing many routes at runtime. See this **FAQ** for more information.
- Fixed potential NPE in **FTP** consumer when using double slashes as starting directory.
- Fixed **InterceptSendToEndpoint** when using `skipSendToEndpoint=true` with a **when** predicate to only skip sending if the predicate was true.
- Fixed an issue with **Bean Binding** directly in a method name not working properly with method call expressions.
- Fixed issue with using custom namespaces in **XPath** expressions when using **OSGi Blueprint**
- Fixed **Validation** to support `xs:import` in the schema files.
- Fixed **Poll Enrich** where the timeout was not explicitly configured to use blocking mode (was wrongly using non-blocking mode).
- Fixed **Threads EIP** to handle rejected policies, ensuring the **Exchange** will finish the `UnitOfWork` and will be de-registered from the inflight registry.
- Fixed **Delayer** not working with a **Bean** expression to define the delay when using XML DSL
- Fixed **JMS** consumer to no longer complain by default if no `errorHandler` is configured when an uncaught exception was thrown while processing a message.
- Fixed the String trim issue in **Simple** expressions.
- Fixed the issue with **Recipient List** not triggering **Error Handler** for endpoints with non existing **Components**.

- Fixed [Load Balancer](#) to update statistics in [JMX](#)
- Fixed [SEDA](#) producer configured with `waitForTaskToComplete=Always` to not hand over on-completion tasks, which otherwise could be processed prematurely.
- Fixed a potential issue with configuring custom start and end tokens for the [Simple](#) language when using Spring XML.
- Fixed an issue with [Type Converter](#) where NaN values were converted to integers.
- Fixed [HTTP](#), [HTTP4](#), and [Jetty](#) producers to send empty HTTP headers for [Exchange](#) headers with an empty value.
- Fixed an issue with configuring a custom thread pool profile on some EIPs.
- Fixed an issue with [Intercept](#) causing processors to not update their performance statistics in [JMX](#)
- Fixed using `<errorHandler>` with embedded `<redeliveryPolicy>` to support [Using PropertyPlaceholder](#)
- Fixed issue with using `<redeliveryPolicy>`, and `<redeliveryPolicyProfile>` with OSGi Blueprint.
- Fixed failover [Load Balancer](#) to perform defensive copy of the message before attempting failover, to ensure previous attempts do not propagate back into succeeding attempts.
- Fixed issue when restarting application using [Quartz](#) with a persistent (non clustered) job store, where the app was not shutdown cleanly (such as crashed)
- Fixed issue when using consumer endpoints with [Property placeholders](#) not working with [interceptors](#) such as `interceptFrom`.
- Fixed issue with suspend/resume a [Timer](#) consumer, not causing timer task to stop firing.
- Fixed [Netty](#) producer to re-use existing connections when sending. This avoid re-creating a new connection for every message being sent.
- Fixed issue with [Simple](#) invoking overloaded methods using OGNL notation, where method name was explicitly declared to invoke the no-parameter method
- Fixed continued and handled [Predicates](#) of [OnException](#) to be evaluated once per exception, instead of twice.
- Fixed [Throttler](#) and [Delayer](#) EIPs when evaluating of max/delay expression fails at runtime, not causing error handling to react.
- Fixed [Mail](#) message with attachments may appear to be lost when [Error Handler](#) performs 2+ redelivery attempts.
- Fixed issue with a failed type conversion using [JAXB](#) may cause subsequent attempts to be regarded as failed as well
- Fixed method call [Expression](#) to preserve headers and [Exchange](#) properties, which allow to use it for storing state with the [Dynamic Router](#) EIP
- Fixed locale issue with configuring endpoints using `uris`, such as Turkish with the lower/upper case of the letter `i`.
- Fixed issue reading and writing files when a `charset` was configured on the file endpoint.
- Fixed issue with `convertBodyTo` not working internally as `transform` and `setBody`, by resetting the message type to `DefaultMessage`.
- Fixed issue with [Loop](#) using `copy` and an exception occurred
- Fixed so [File](#) consumer using done file name, will not automatic delete the done file if `noop=true` is configured.
- Fixed issue with [SERVLET](#) may send back incorrect content length for data with unicode characters outside the normal ASCII range.
- Fixed [Jetty](#) http producer (client) may corrupt streams if used in a bridge mode (eg like a proxy)
- Fixed [Jetty](#) http consumer (server) may corrupt streams if chunked was disabled
- Fixed issue with `camel-blueprint` and using `CamelContextAware` beans for `EventNotifier` (and the beans from this [list](#)) injecting the `CamelContext` which is not ready, causing the blueprint container to hang waiting for the service.
- Fixed [Bindy](#) to not use default locale, but only use locale if explicit configured.
- Fixed [Restlet](#) not propagating headers during routing.
- Fixed running multiple Camel apps with same CamelContext name in a single JVM with [Quartz](#) component, causing jobs to pick wrong CamelContext.
- Fixed URI normalization to support % sign itself, by specifying the % sign using its encoded value (eg %25).
- Fixed [JMS](#) producer to trigger Camel error handling if destination resolution failed due to an exception.
- Fixed [Error Handler](#) with async redelivery to use shared thread pool per CamelContext, instead of per JVM
- Fixed issue with added headers, using `setHeader` may be lost if a [JMS](#) route uses the [Dynamic Router](#) EIP

New Enterprise Integration Patterns

New Camel Karaf Command

- Add `camel:endpoint-list` to list all Camel endpoints available in Camel Context deployed on Karaf/ServiceMix

New Components

- `camel-avro` - for [Apache Avro](#) integration
- `camel-aws` - for [DynamoDB](#) integration within [Amazon Web Services](#)
- `camel-beanio` - [Data Format](#) to map objects to flat structured format
- `camel-cdi` - to deploy Apache Camel in CDI container with Dependency Injection (JSR299, 330)
- `camel-core` - new support for "[stubbing](#)" endpoints, for development and testing purposes
- `camel-core` - new [direct-vm](#) for synchronous messaging between different CamelContext in the same JVM (for example to support transactions, across CamelContexts).
- `camel-fop` - to render messages into different outputs using [Apache FOP](#)
- `camel-gson` - provides a [JSON](#) data format using the [Google GSON project](#)
- `camel-guava-eventbus` - provides a integration bridge between Camel and [Google Guava EventBus](#) infrastructure.
- `camel-hbase` - provides integration with [HBase](#)
- `camel-jsch` - Secure Copy using [JSCH](#)
- `camel-mina2` - `camel-mina2` is expected to be refactored in Camel 2.11, so use on own risk of having to migrate.
- `camel-mongodb` - for [MongoDB](#) integration
- `camel-spring-batch` - For using Spring Batch with Camel.
- `camel-ssh` - For executing commands on an [SSH server](#)
- `camel-test-blueprint` - for testing Blueprint routes
- `camel-test-spring` - for testing Spring routes
- `camel-twitter` - for [Twitter](#) integration
- `camel-websocket` - for [WebSocket](#) integration

DSL Changes

New Annotations

New Data Formats

- `camel-avro` - for [Apache Avro](#) integration
- `camel-beanio` - for mapping objects to flat structured format
- `camel-gson` - provides a [JSON](#) data format using the [Google GSON project](#)

New Languages

New Examples

- `camel-example-ssh`
- `camel-example-twitter-websocket` - shows how to use Twitter to push live tweet updates to a webpage with Websocket
- `camel-example-console` - a new beginner example that shows how to read input from the console

New Tutorials

API changes/breaking

- Added `shutdownForced` and `forceShutdown` methods to `org.apache.camel.spi.ShutdownStrategy`
- Add boolean parameter to `prepareShutdown` on `ShutdownAware`
- The `errorHandler` method in `RouteBuilder` now returns `void`, as it's only used for configuring context-scoped error handlers.
- Removed the `SimpleLanguage` constructor that takes custom start and end tokens. Instead use the `change` methods to set custom tokens.
- Added `onThreadPoolRemove` and `onErrorHandlerRemove` methods to `LifecycleStrategy`
- Removed `retryWhile(Expression)` on `OnExceptionDefinition`; use `retryWhile(Predicate)`.
- Removed options `corePoolSize` and `maxPoolSize` on [Netty](#) endpoint, as we rely on [Netty](#) default implementation.
- The `resourceClasses` option on [CXFRS](#) no longer supports the semicolon (;) for separating class names. Now only the comma (,) is supported (which is the standard in Camel)
- Added `tryConvertTo` methods to `TypeConverter`, which works as the regular `convertTo` methods, by ignoring any exceptions during conversion.
- The `convertTo` methods on `TypeConverter` now throws `TypeConversionException` if an exception occurred during type conversion.
- The `getBody(type)`, and `getHeader(name, type)` methods on `Message` now throws `TypeConversionException` if an exception occurred during type conversion.
- The [Simple](#) language no longer trim the expression in Java DSL. Instead we introduced a `trim` option in the XML DSL to remove the white space noise from XML tags.
- Added `containsSynchronization` to `UnitOfWork` and `Exchange` to check if a given on completion has already been added to the given exchange.

Internal/Non API refactorings that may affect users

- Removed method `setManagementName` from `CamelContext` as it should not be used by Camel end users
- Added method `isTransactedRedelivered` to [Exchange](#)
- Added method `isDirectory` to `GenericFile`
- API changed in the `ClientPipelineFactory` and `ServerPipelineFactory` abstract classes in [Netty](#)
- Added `getStatistics` method to `TypeConverterRegistry`, and the registry now extends `Service` as well.
- The `convertBodyTo` with a `charset` configured, will not propagate the `charset` as an `Exchange` property (`Exchange.CHARSET_NAME`), which otherwise may lead to payload being double converted.
- Added `abort` method to `GenericFileProcessStrategy`.
- `camel-mail` now excludes dependency on `javax` activation JAR as that's embedded in the JVM from JRE6 onwards.
- URI normalization will now better detect [pre existing % decimal encoded values](#).

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.
- `camel-spring-test` bundle has the classes from `camel-test`, if you are using `camel-spring-test` in OSGi platform, please don't install the `camel-test` at the same time.
- [CXF Example OSGi](#) doesn't work without enabling dynamic imports for this bundle.

Dependency Upgrades

- AHC 1.6.5 to 1.7.5
- AWS 1.2.2 to 1.3.10
- Commons-codec 1.4 to 1.6
- Commons-net 2.2 to 3.1.0
- CXF 2.5.1 to 2.6.1
- EHCache 2.4.3 to 2.5.1
- Freemarker 2.3.18 to 2.3.19
- Google App Engine 1.5.0 to 1.6.6
- Groovy 1.8.5 to 1.8.6
- Hadoop 0.20.203.0 to 1.0.3

- HTTP4 core 4.1.2 to 4.1.4
- HTTP4 client 4.1.2 to 4.1.3
- Hazelcast 1.9.4.4 to 2.0.2
- Hawtbuf 1.7 to 1.9
- Jackson 1.9.2 to 1.9.7
- Jackrabbit 2.2.4 to 2.2.11
- Jasypt 1.7 to 1.9.0
- Javax Mail 1.4.4. to 1.4.5
- Jersey 1.10 to 1.12
- JClouds 1.3.1 to 1.4.0
- Jettison 1.3 to 1.3.1
- Jetty 7.5.3 to 7.5.4
- JRuby 1.6.6 to 1.6.7
- JSCH 0.1.44 to 0.1.48
- JuEL 2.1.3 to 2.1.4
- Kratti 0.4.1 to 0.4.5
- Logback 1.0.0 to 1.0.6
- Lucene 3.0.3 to 3.6.0
- MyBatis 3.0.6 to 3.1.1
- Netty 3.2.6 to 3.5.1
- OGNL 3.0.2 to 3.0.4
- QPid 0.12 to 0.16
- QuickFIX/J 1.5.1 to 1.5.2
- Resetlet 2.0.10 to 2.0.14
- SNMP 1.8.1 to 1.10.1
- Solr 3.5.0 to 3.6.0
- Shiro 1.1.0 to 1.2.0
- Stringtemplate 3.0 to 3.2.1
- Spring 3.0.6 to 3.0.7/3.1.1
- Spring Integration 2.0.5 to 2.1.2
- Spring Security 3.0.7 to 3.1.0
- Spymemcached 2.5 to 2.8.0
- Tagsoup 1.2 to 1.2.1
- Woodstox 4.1.1 to 4.1.2
- XStream 1.4.1 to 1.4.2
- XML Security 1.4.5 to 1.5.1

Important changes to consider when upgrading

- Moved [Spring Testing](#) from `camel-test` to `camel-test-spring` component.
- Combination of `skipSendToEndpoint=true` and `when` predicate now triggers conditional skipping (you may have been relying on unconditional skipping up until now)
- The thread name pattern has changed tokens to avoid clashing with [Using PropertyPlaceholder](#) (see [Threading Model](#) for more details).
- [Poll Enrich](#) will now block if no messages are available and no explicit timeout has been configured.
- The option `lazyLoadTypeConverter=true` has been deprecated, and will be removed in a future release. The best practice is to load type converters while bootstrapping Camel.
- The [Camel Test kit](#) no longer lazily loads type converters.
- Keys for [MDC logging](#) have been prefixed with "camel.".
- [Wire Tap](#) can no longer change destination from [JMX](#)
- Camel now fail faster during type conversion, by throwing `TypeConversionException` to the caller, from APIs such as `TypeConverter` and `Message`. For example `getBody(type)` on the `Message` API.
- Camel no longer supports using `java.beans.PropertyEditor` for type conversion. These are slow, neither thread safe, and using 3rd party JARs on the classpath may cause side effect as if they install `java.beans.PropertyEditor`.
- Use the `charset` option on the [File](#) component to explicitly specify encoding for reading and writing files.
- Maven 3.0.2 or better is required to build the [Source](#)

Notice

- If you want to deploy Apache Camel 2.10.0 into Apache Karaf, we highly recommend to use the 'jre.properties.cxf' configuration, which is part of Apache Karaf 2.2.5 onwards. For example `camel-cxf`, and `camel-spring-ws` requires this to be installable; as well other features may as well.
- Support for JUnit 3.x in the Camel [Tests](#) components is being discontinued and expected to be removed from Camel 2.11 onwards.

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.10.0.zip	apache-camel-2.10.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.10.0.tar.gz	apache-camel-2.10.0.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-2.10.0-src.zip	apache-camel-2.10.0-src.zip.asc