

# Camel 2.12.0 Release

Camel 2.12.0 release

[red URL](#)

## New and Noteworthy

Welcome to the 2.12.0 release which approx XXX issues resolved (new features, improvements and bug fixes such as...)

- [Endpoint Annotations](#) along with automatically created HTML documentation for the endpoint parameters; this makes it easier for component developers to add a few refactoring-safe annotations to their Endpoint or Consumer implementations and, (along with javadoc comments on the field or setter method), get nice user documentation on how to use the endpoint for free.
- [ComponentConfiguration](#) API provides a handy API for tools developers to introspect on a Component to find all the possible parameters, their types and any extra annotations (like Bean Validation Annotations) to be able to create/edit endpoints or URI strings so that tools can generate nicer UIs for configuring endpoints than just letting folks edit Strings.
- [EndpointCompleter](#) API provides a hook so that command line tools (like [Karaf's shell](#)), IDEs and web tools can get (bash tab like) auto-completion on endpoint paths (such as file or directory names, message queue names, database table names) when creating or using new endpoints
- Reduced stack-frames in use during routing, that also makes Camel's stack traces being logged much less verbose. This also allows people to easier debug the internals of Camel as less `AsyncCallback` callbacks are in use during routing.
- Easy to use [Message History](#) out of the box. And included message history as "route stack-trace" when exceptions logged by [Error Handler](#) to make it easier for end users to spot where the exception occurred.
- [Spring Web Services](#) now supports setting/receiving SOAP headers more easily using a header on the Camel [Message](#).
- Evaluating [Groovy](#) expressions is faster as we cache the compiled scripts.
- Added `base64` option to [Shiro Security](#) to allow transferring security token over [JMS](#) and other transports as base64 encoded representation.
- Made it easier to use [Shiro Security](#) as the credentials can be provided in headers, when sending a message to a secured route.
- [Bindy](#) now supports enums.
- Added new [BacklogDebugger](#) to perform live debugging of messages during routing. The [BacklogDebugger](#) has JMX API allows tooling to control the debugger.
- While using the [Jackson library](#) through the [JSON](#) Dataformat there's now a `jsonView` attribute you could make use of directly inside the [DSL](#) itself.
- [SMPP](#) now supports optional parameters in all commands where they are possible.
- [JDBC](#) now supports named parameters.
- Added timeout support for [Direct](#) producers to wait for consumer to become active.
- Added `stats` action to [ControlBus](#) to easily get performance statics in a single XML message.
- Added support for request timeout on [Netty](#) producer, and to configure logging level on [Netty](#) consumer to be less noisy for `ChannelClosedException` which can flood the logs when client disconnects abruptly.
- [Spring Batch](#) component producer now returns the `JobExecution` instance as the output message. Users can use the `JobExecution` instance to perform some operations using the Spring Batch API directly.
- Added support for NULL values in [SQL](#) with named parameters.
- Optimized [Jetty](#) streaming responses in non-chunked mode; and as well using buffer sizes based on `HttpServletResponse.getBufferSize()` instead of fixed size of 4kb.
- Added `greedy` option to [scheduled polling consumer](#).
- Added support for accessing the length field on Java arrays in [Simple](#) language.
- Added support for using JAXB annotations with Camel's Jackson [JSON](#) component.
- Added the ability for [Bindy](#) to skip content when parsing fixed-length records.
- [MongoDB](#) now supports aggregation queries.
- [Netty](#) allows to use shared Netty boss and worker thread pools.
- The [Camel Maven Archetypes](#) for component and data format now packages as OSGi bundles out of the box.
- Easier [Stream caching](#) configuration using `StreamCachingStrategy`. Also allows spool directory per [CamelContext](#) instead of shared per JVM. And insight at runtime using JMX management. As well allowing to plugin 3rd party implementations.
- Improved [Netty](#) to be able to join an UDP multicast group.
- Using Camel [Spring](#) is now setting up Camel routes later in the process allow Spring to complete all of its dependency injection, before Camel `RouteBuilder` instances is configured. This allows to safely use Spring dependency injection in these `RouteBuilder` classes.
- Services in a [CamelContext](#) which can only be enlisted once (eg it similar to a static in a JVM) has simpler and fixed MBean names in JMX.
- [JDBC based aggregation repository](#) for the [Aggregate](#) EIP now supports optimistic locking, allows clustered Camel applications to use a shared database for the aggregation repository.
- [Recipient List](#) stores a property (`Exchange.RECIPIENT_LIST_ENDPOINT`) on the [Exchange](#) with the [URIs](#) of the [Endpoint](#) the message was sent to.

- Added `mask` option to [Camel JMX](#) to hide sensitive information such as passwords.
- Added `outputType` option supporting `SelectList`, and `SelectOne` to [SQL Component](#) to dictate the output message body type when doing SQL SELECT queries.
- [Polling Consumers](#) such as [File](#), and [FTP](#) now supports using custom scheduler. Providing a new [Quartz2](#), and [Spring](#) based out of the box, that allows to use CRON based scheduler.
- [Polling Consumers](#) such as [File](#), and [FTP](#) now supports *backoff*, to let the consumer be less aggressive, when there has been no messages to poll, or errors keeps occurring for a while.
- Added support for XML Encryption 1.1 algorithms in the [XMLSecurity data format](#). It is now possible to specify stronger digest algorithms when using RSA OAEP Key Transport algorithms.
- [Netty](#) consumer now unbinds/binds the acceptor when the route is suspended/resumed.
- [EIPs](#) supporting [AggregationStrategy](#) such as [Aggregate](#), [Splitter](#), [Content Enricher](#) now supports using a POJO as the `AggregationStrategy`, which allows to implement the aggregate logic with no Camel API dependency. See more details at the [Aggregate](#) page.
- `camel-jpa` upgraded to use native JPA API instead of Springs deprecated `JpaTemplate`
- Improved validation of routes being misconfigured on startup, when using `onException`, `onCompletion`, `transacted`, etc. as they must be configured as top-level on the route (eg in the start), and not later (such as nested in a [Splitter](#)).
- Added new `@PropertyInject` annotation to inject [property placeholders](#) to POJOs
- New class `org.apache.camel.util.toolbox.AggregationStrategies` as a starting point to obtain commonly used [Aggregation Strategies](#) for Multicast, Splitter, Aggregator, etc. [EIPs](#).
- New [FlexibleAggregationStrategy](#) serving as a one-stop to perform typical aggregation strategy logic out-of-the-box: filtering results, storing them in properties, headers or bodies, as a list, casting results, etc.
- Improved performance when doing [Bean](#) method calls using [Simple](#) OGNL calls
- Improved [Dozer Type Conversion](#) to work better with OSGi Blueprint (though Dozer 5.5 may be required to work even better)
- [JMS](#) now also enlists topic destinations as managed endpoints in [JMX](#)
- [Graceful Shutdown](#) now allows to suppress logging after `CamelContext` has been stopped, due to a timeout hit, and there may still be inflight [Exchanges](#) that may afterwards be rejected being executed and causing WARN logs; which now can be suppressed.
- Made it easy to add custom services on startup when testing with [Blueprint Testing](#).
- XML Signature support via the [XML Security component](#).

## Fixed Issues

- Fixed setting `.id` in Java DSL on some [EIPs](#) such as the [Validate](#)
- Fixed [Dead Letter Channel](#) always handled exception, which wasn't the case if an [Exception Clause](#) was in use, which didn't had `handled=true`.
- Fixed [Using PropertyPlaceholder](#) in attributes defined on `<camelContext>`.
- All Camel [Components](#) will configure their consumers when being created to ensure you can configure from [URIs](#) with `consumer.` as prefix.
- Fixed [QuickfixJEndpoint](#) to adhere a given `InOut MEP` correctly.
- Fixed using property placeholders in a [Simple](#) predicate used inside a `when` in a [Content Based Router](#) when using [Java DSL](#)
- Fixed a race condition in [File](#) consumer which may lead to a tiny window for another process to pickup a file, as the read lock is committed before the file is moved afterwards etc. Now read locks is executed last.
- [HTTP4](#) now validates that only one instance of `org.apache.camel.util.jsse.SSLContextParameters` is used per component, as multiple is not supported, and could lead to undesired side effects.
- Fixed possible dead-lock during [Graceful Shutdown](#) if a route starts from [SEDA](#) and uses [Splitter](#) or [Multicast](#) and `maximumRedeliveries` was enabled as well.
- Fixed [TokenizeXML](#) to support self closing tags.
- Fixed [Blueprint Testing](#) to work with `isMockEndpointsAndSkip` and other similar functionality of the `CamelBlueprintTestSupport` class.
- Fixed [Routing Slip](#) to emit `ExchangeSentEvents` also
- Fixed [Quartz](#) routes with stateful jobs may not trigger if redeployment with changed endpoint uris.
- Fixed [Recipient List](#), [Routing Slip](#), and [Dynamic Router](#) [EIPs](#) to work with [Stream caching](#)
- Fixed suspending/resuming Camel [Quartz](#) routes
- Fixed [SQL Component](#) using queries with a single parameter and the data is JSON like structure (having comma separators)
- Fixed [Jetty](#) to be able to use stream cached streams that was spooled to disk, as its reply message when sending back reply to the caller.
- Fixed `endChoice()` in Java DSL to work when being nested in parent [Content Based Router\(s\)](#)
- Fixed [Lucene](#) to reuse existing index directory if Camel is being restarted etc.
- Fixed using `<routeContextRef>` to not share the routes when using multiple `<camelContext>`, but each `<camelContext>` have their own route instances to not cause side-effects between the [CamelContexts](#).
- Fixed using type converter from `NodeList` -> `Node`
- Fixed having 2+ routes from the **same** JMS queue, not stop the endpoint if there are still active listeners when a route is stopped.
- Fixed [SEDA](#) when using `queueSize` not able to identify its same queue size, when using multiple [SEDA](#) endpoints for the same queue name /size.
- Fixed using `<routeBuilder ref="name"/>` in OSGi Blueprint when referring to a `RouteBuilder` from another bundle, exposed in the OSGi service registry.
- Fixed OSGi Blueprint in some situations causing a `CircularDependencyException` when using Camel namespaces outside a `<camelContext>`, being referred from inside the `<camelContext>`.
- Fixed [HTTP4](#) in bridge mode to not store cookies, which otherwise would eat up memory.

## New Enterprise Integration Patterns

### New Components

- `camel-core` - A Camel [DataFormat Component](#) for working with [Data Formats](#) as if it was a regular [Component](#) supporting [Endpoints](#) and [URIs](#).
- `camel-disruptor` - [SEDA](#) behavior using the ultra-fast [LMAX Disruptor](#) library.
- `camel-facebook` - A Camel component for integrating with all of the Facebook APIs.
- `camel-geocoder` - For looking up geocodes (latitude, longitude) for a given address, or reverse lookup with [Java API for Google Geocoder](#)
- `camel-leveldb` - Added docs for this component.
- `camel-mustache` - For templating using [Mustache](#)
- `camel-mvel` - For templating using [MVEL](#)
- `camel-netty-http` - HTTP component using [Netty](#) as transport
- `camel-quartz2` - For scheduling using the Quartz 2.x scheduler

- [camel-sap-netweaver](#) - For integration with [SAP NetWeaver Gateway](#)
- [camel-stomp](#) - For communicating with [Stomp](#) compliant message brokers, like [Apache ActiveMQ](#) or [ActiveMQ Apollo](#)
- [camel-vertx](#) - For working with the [Vertx EventBus](#).
- [camel-weather](#) - For polling weather information from [Open Weather Map](#)
- [camel-salesforce](#) - Allows you to interact with the [Salesforce](#) using Java DTOs.
- [camel-yammer](#) - Allows you to interact with the [Yammer](#) enterprise social network
- [camel-rabbitmq](#) - Allows you to interact with the [RabbitMQ](#).
- [camel-xmlsecurity](#) - Support for the XML Signature specification.

## New DSL

- New fluent builder for Aggregation Strategies: `org.apache.camel.util.toolbox.AggregationStrategies`.

## New Annotations

- `@PropertyInject`

## New Data Formats

## New Languages

## New Examples

- [MyBatis Example](#)
- [Netty HTTP Server Example](#)
- `camel-example-jdbc` - An example using [JDBC](#) component with an embedded database
- `camel-example-restlet-jdbc` - An example using [Restlet](#) and [JDBC](#) with an embedded database

## New Tutorials

## API changes

- The interface `org.apache.camel.api.management.mbean.ManagedTracerBacklogMBean` has been renamed to `org.apache.camel.api.management.mbean.ManagedBacklogTracerMBean`
- Added `inMemory`, and `length` methods to `org.apache.camel.StreamCache`
- Changed the parameter type `ProcessorDefinition` to `NamedNode` on method `getObjectNamesForProcessor` in interface `org.apache.camel.spi.ManagementNamingStrategy`
- Added `removeTypeConverter` method to `org.apache.camel.spi.TypeConverterRegistry`
- Added `allowNull` method to `org.apache.camel.TypeConverter`
- Added `hasTimeoutOccurred()` method to `ShutdownStrategy`

## Known Issues

## Dependency Upgrades

- AHC 1.7.11 to 1.7.19
- AWS Java SDK 1.3.27 to 1.5.1
- Beanio from 2.0.5 to 2.0.6
- Bouncy Castle 1.47 to 1.49
- Commons Codec 1.6 to 1.8
- Commons Net 3.1 to 3.3
- Commons HttpClient 4.2.3 to 4.2.5
- Commons HttpClient 4.2.3 to 4.2.4
- CXF 2.7.4 to 2.7.6
- Disruptor 3.1.1 to 3.2.0
- Ehcache 2.5.2 to 2.7.2
- GAE 1.7.4 to 1.8.2
- HAPI 2.0 to 2.1
- Groovy 2.1.3 to 2.1.6
- Hadoop 1.1.1 to 1.2.0
- HBase 0.94.6 to 0.94.10
- Hawtdispatch 1.13 to 1.17
- Hibernate Validator 4.1.0.Final to 5.0.1.Final
- Hibernate 4.2.3 to 4.2.4
- Jackson 2.1.4 to 2.2.2
- Jersey 1.13 to 1.17.1
- Jettison 1.3.3 to 1.3.4
- JRuby 1.7.2 to 1.7.4
- Krati 0.4.8 to 0.4.9
- LevelDbjni 1.6 to 1.7
- Mail 1.4.5 to 1.4.7
- MongoDB Java Driver 2.9.1 to 2.11.2

- MQTT Client 1.4 to 1.5
- MVEL 2.1.3.Final to 2.1.6.Final
- Netty 3.6.5 to 3.6.6
- OpenEJB 4.5.1 to 4.5.2
- OpenJPA 2.2.1 to 2.2.2
- Saxon 9.4.0.4 to 9.5.0.2
- Scala 2.10.1 to 2.10.2
- Shiro 1.2.1 to 1.2.2
- SNMP4J 2.1.0 to 2.2.2
- Spring 3.1.4.RELEASE to 3.2.4.RELEASE
- Spring Batch 2.1.9.RELEASE to 2.2.1.RELEASE
- Spring Data Redis 1.0.3.RELEASE to 1.0.4.RELEASE
- Spring Security 3.1.3.RELEASE to 3.1.4.RELEASE
- Spring Web Services 2.1.2 to 2.1.3
- TestNG 6.8 to 6.8.5
- XML Security (Santuario) 1.5.2 to 1.5.5.
- XStream 1.4.3 to 1.4.4

## Internal changes

- The following classes in `org.apache.camel.processor` has been removed as they are no longer needed due internal optimization: `UnitOfWorkProcessor`, `ChildUnitOfWorkProcessor`, `RouteContextProcessor`, `RouteInflightRepositoryProcessor`, and `RoutePolicyProcessor`

## Changes that may affect end users

- The [File](#) and [FTP](#) consumers when using idempotent repository will no longer invoke `contains` with a directory name; this has been changed to only be checked for files.
- Shutting down using [Graceful Shutdown](#) now requires the timeout value to be positive. And improved logic to avoid getting stuck in waiting for inflight messages, and having counter go negative.
- [Tracer](#) now requires explicitly to be enabled on `CamelContext` to be available and in use.
- The class `org.apache.camel.processor.interceptor.BacklogTracerInterceptor` has been removed as it was no longer needed due internal optimizations.
- The [Stream caching](#) default threshold for spooling to disk has been changed from 64kb to 128kb.
- The unary operators in [Simple](#) is now only applied on functions
- Using Camel [Spring](#) is now setting up Camel routes later in the process allow Spring to complete all of its dependency injection, before Camel `RouteBuilder` instances is configured. This allows to safely use Spring dependency injection in these `RouteBuilder` classes.
- Changing destination using JMX on `org.apache.camel.api.management.mbean.ManagedSendProcessorMBean` no longer supported.
- Camel now uses `xxxxxx` instead of `*****` when masking passwords (sanitize) when logging [URIs](#) in the log. Using `xxxxxx` is more friendly with JMX naming.
- The `DefaultProducer` now delegates to the `Endpoint` in the `isSingleton()` method call, to use the same value as its endpoint.
- Requests that use RSA v1.5 as the key transport algorithm will be rejected in the camel-xmlsecurity component, unless this algorithm has been explicitly configured.
- The [Camel Maven Plugin](#) now uses "compile+runtime" as scope for required dependencies added to the classpath when running the plugin. This ensures also dependencies with provided scope gets included.
- The `userConfigURL` option from [FOP](#) now loads the resource from classpath by default.

## Important changes to consider when upgrading

- Spring 3.0.x is no longer supported.
- The usage of the deprecated built-in `CommonsHttpMessageSender` has been replaced with `URLConnectionMessageSender` by [Spring Web Services](#). Consider this in case either the `timeout` or `sslContextParameters` option is used through the `Producer` of this component.
- `camel-scala-29` module has been removed.

## Getting the Distributions

### Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	<a href="#">apache-camel-2.12.0.zip</a>	<a href="#">apache-camel-2.12.0.zip.asc</a>
Unix/Linux/Cygwin Distribution	<a href="#">apache-camel-2.12.0.tar.gz</a>	<a href="#">apache-camel-2.12.0.tar.gz.asc</a>



#### 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	<a href="#">apache-camel-2.12.0-src.zip</a>	<a href="#">apache-camel-2.12.0-src.zip.asc</a>

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>2.12.0</version>
</dependency>
```

## Git Tag Checkout

```
git clone https://git-wip-us.apache.org/repos/asf/camel.git
git checkout origin/camel-2.12.0 -b camel-2.12.0
```

## Changelog

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

- [release notes for 2.12.0](#)