Camel 2.1.0 Release

Camel 2.1.0 release

New and Noteworthy

Welcome to the 2.1.0 release which approx 303 issues resolved (new features, improvements and bug fixes such as...)

- Pluggable API for Management allowing 3rd party to manage Camel.
  - Use `org.apache.camel.spi.ManagementStrategy` to plugin 3rd party adapter
  - Fine grained events notifications is now being emitted using `org.apache.camel.spi.EventNotifier`
  - Events being fired can be created using a pluggable factory using `org.apache.camel.spi.EventFactory`
  - Camel now supports using multiple `org.apache.camel.spi.LifecycleStrategy` strategies, that also allows 3rd party to more easily integrate into Camel.
- Overhaul of default JMX management in Camel to be much improved with more attributes, mbeans, operations, statistics, and route timing working as well.
  - And yes you can now start/stop routes and individual consumers and producers
  - And some special processors such as Delayer and Throttler can be managed to change their delay/throttle values
  - You can dynamic manage tracing even for already running applications
  - Error handling in Camel is now also manageable so you for example can change the redelivery policies.
  - JMS consumers being restarted will pickup any configuration changes on JMS endpoints such as changing `concurrentConsumers`. And yes its fully managed using JMX.
  - Number of inflight exchanges per consumer, route and totally per camel context.
  - And much more...
- Improved IRC, XMPP, MINA, CXF and FreeMarker components with much thanks to the Camel community
- Added classLoader set and get method in the CamelContext to make Camel work better in the Container.
- Introduced camel-spring-osgi bundle which is merged from camel-spring and camel-osgi bundle to avoid the issue of installing upper bundles in a wrong order.
- Now Camel Karaf features are validated with the Karaf plugin.
- camel-cxf beans component supports JAXWS beans.
- camel-cxf MTOM attachment supports in Payload mode.
- Fixed the Null message will cause camel converter stop working issue.
- camel-cxf endpoint no longer requires the "serviceClass" option or Service Endpoint Interface (SEI) in Payload mode. It means only WSDL is needed but the intended port and service name in the WSDL must be provided for the camel-cxf endpoint.
- Velocity component supports to take the Velocity configuration by using propertiesFile option.
- Splitter and Multicast EIP patterns now support the option `stopOnException` to stop continue processing at first exception occurred. See their documentation for more details.
- Error handlers is restricted to be configured on camel context and/or routes only.
- Camel is now more robust and fault tolerant when shutting down.
- Fixed a race condition when shutting down Camel with JMS consumers that under some circumstances caused it to appear to hang
- Added pluggable `org.apache.camel.spi.InflightRepository` with a basic implementation that can help with graceful shutdown as you can now see how many exchanges are currently in progress.
- JmsEndpoint is now singleton
- Individual routes can be pre configured to not auto startup when Camel starts: `autoStartup=false` in XML DSL and `noAutoStartup()` in Java DSL.
- Camel is more robust on startup as route inputs are deferred to be started at the very end, ensuring all routes services have been pre started and thus ready when consumers is started.
- Improved unit test coverage, now having more than 4200 unit tests in total.
- Fixed `ConsumerTemplate` not working when it was dependency injected with `@EndpointInjected` in a Bean.
- Using IBM special JMS headers now work when sending messages to IBM WebSphere MQ. IBM special headers uses this prefix: `JMS_IBM_`. In fact it works with any `JMS_<vendor>_` header.
- The ordering of which routes should be started can now be configured by the `startupOrder` attribute which defines a number. Lowest number is started first.
- And `autoStartup` is also supported on CamelContext itself where it replaces the `shouldStartContext` option, which have been removed.
- Camel now checks on startup if multiple routes uses the **same** endpoint as inputs, if so it fails to start. This prevents starting Camel if you have routes with e.g. copy/paste typos. Only JMS and MINA support multiple routes to consume from **same** endpoint. The `MultipeConsumersSuppo rt` interface can be optionally implemented by the endpoint to control this behavior.

- The File component handles renaming and deleting files on Windows better as Windows could block files. Also Antivirus software could causes files to be blocked.
- Fixed issues using `tempPrefix` option with the File in case of clashes with existing files. The same goes for the `localWorkDirectory` option on the FTP component which could have a similar issue.
- Optimized XPath to allow thread safe concurrency which greatly reduces contention.
- Camel works better with Google App Engine
- The Bean component is better at choosing the best method. If given a method name it will only pick among methods with such a name.
- The Bean component is now also more strict as it **must** be able to convert to all the parameter types. Was to lenient before by passing in `null` instead, when conversion was not successful.
- The Bean component now works with Spring @Transactional annotations where it uses CGLIB enhanced classes.
- Fixed an issue with setting global error handling in Spring XML not always working
- Camel adds properties to Exchange when sending to endpoints. This is useable with Dead Letter Channel where we can use this to know which Endpoint failed.
- Improved file operations on Windows platform to retry in case of issues.
- Fixed issue with `replyTo` option on JMS consumers which wasn't working.
- Fixed issue with using <camel:proxy> not working
- Improved `<camel:proxy>` so its easier to work with out of the box. It leverages Camels type converter mechanism to adapt to the interface. Also proxied methods with a single parameter turns that parameter value into the message body.
- Added document for using Camel proxy.
- Improved `enrich` and `pollEnrich` with reference lookup of endpoints and aggregation strategy.
- Added `markRollbackOnly` attribute on `<rollback>` to only mark the Exchange for rollback. This allows you to set a custom reply message. Previously a `RollbackExchangeException` was always set which causes Camel to thrown an exception to the client. This also works with `transacted` routes as well.
- Camel now logs on shutdown whether there was in flight exchanges or not. All together we are getting closer to having graceful shutdown supported in Camel.
- `@Bean` is now easier to use at you define `ref` and `method` instead of one combined value.
- Introduced `routeId` in the Java DSL to set the id of the route. Prefer to use this over `id`.
- Introduced RoutePolicy allowing to automatic and dynamically controlling routes at runtime.
- Supplied `ThrottlingRoutePolicy` out of the box for dynamic throttling route consumers using metrics based on the current number of inflight exchanges.
- Data Format is now resolved just as Components and Languages by looking for magic file under `META-INF/services/org/apache/camel/dataformat/`.
- The HTTP component now also supports using NTLM authentication
- Exception Clause improved to better select the best `onException` to use. Now a direct match will **always** be used in case one existed anywhere in the caused exception hierarchy. This avoid Camel eagerly selecting a too generic such as `onException(Exception.class)` when a direct match existed.
- FTP can now have its `FTPClient` and `FTPClientConfig` configured using URI parameters. Can for example be used to set a timeout on the `FTPClient`.
- FTP is now more robust when writing files where it will reconnect the connect in case of problems.
- Added support in camel-core for non blocking Request Reply using ToAsync.
- Jetty HTTP producer supports non blocking Request Reply. See more at ToAsync and the HTTP Async Example.
- `HttpOperationFailedException` now uses pure Java types so it can safely be transported over network without being dependent on 3rd part .jars.
- Improvements to Tracer to let it be more fine grained when used with Aggregator, Splitter etc. who has sub routes. It now also outputs correct doTry .. doCatch .. doFinally and onException etc.
- Added `adviceWith(RouteBuilder)` to `RouteDefinition` which allows you to advice any existing route with a route builder where you can define interceptors etc. This is useable for testing so you can write tests based on existing routes and advice them with interceptors so you can detour messages and simulate errors etc.
- Timer and Quartz now logs Exchanges that failed with an exception at ERROR level to help identify the problem.
- Google App Engine supported excellent work by our new committer Martin Krasser.
- Improved CXF producer in `POJO` mode to use the POJO body without having to wrap it in a `java.util.List` which allows for easier usage. See for example the CXF Async Example.
- Added option `testConnectionOnStartup` to JMS to test connection on startup. The goal is to ensures that when Camel is started all the JMS consumers have a valid connection to the JMS broker. However if a connection cannot be established then Camel throws an exception on startup.
- Dead Letter Channel is validating its `deadLetterUri` on startup to prevent failing at runtime.
- Failing to create routes is now reported with a more precise error message which pinpoints which route and what part of the route is causing the problem.
- Fixed issue with transacted routes mixed with Camel error handling doing redelivery from the beginning of the route and not only at the failed node in the route path.
- Fixed issue with using BAM with Hibernate.
- Options to the `java.sql.Statement` can be set from the URI in the JDBC component. Can be used to set maxRows etc.
- Added documentation for NMR and Quickfix.
- Fixed issues with Mail and having multiple endpoints defined with different recipients could cause Camel to use the recipients from the last defined endpoint only.
- Improved Mail to better recover from network failure on subsequent polls
- Improved Tracer to output route id
- Fixed a potential 2gb file copy limit using NIO when doing from("file:xxx").XXXX.to("file:yyy") routing.
- Upgraded and fixed EL to work with OSGi containers.

New Enterprise Integration Patterns

- Sampling A sampling throttler allowing you to extract a sample of exchanges from the traffic through a route.

New Components

- camel-printer
- camel-cache

- camel-snmp
- camel-javaspace
- camel-gae
- camel-quickfix

New DSL

- `sample`
- `markRollbackOnly`
- `routeId`
- `toAsync`
- `autoStartup`
- `startupOrder`

New Annotations

New Data Formats

- Camel-bindy Bindy includes important new features like :
    - Required (@DataField) to check mandatory field
    - Position (@DataField and @KeyValuePairField) when the CSV/FIX message to be generate contain fields which are placed at a different position then those used to parse it
    - Section which allow to define the class containing by example the header, body or footer section
    - OneToMany which allow to Read a FIX message containing repetitive groups (= group of tags/keys) and Generate a CSV with repetitive data

New Languages

New Examples

- camel-example-management
- camel-example-route-throttling
- camel-example-http-async
- camel-example-cxf-async

API breaking

DSL changes

In the Java DSL the `.errorHandler` is now restricted to be configured on camel context and/or routes only. That means for configuring on routes you have to set it directly after the `.from` DSL.

In Spring DSL the `errorHandlerRef` attribute is now only visible on the `<camelContext>` and `<route>` XML tags.

New .jar dependencies

camel-core now depends on commons-management.jar to facilitate the new overhauled management.
This project is hosted at FuseForge

This .jar can be retrieved from the maven central repo at: http://repo1.maven.org/maven2/org/fusesource/commonman/commons-management/1.0/
The maven dependency details are:

```
<groupId>org.fusesource.commonman</groupId>
<artifactId>commons-management</artifactId>
<version>1.0</version>
```

DataFormat resolution

Data formats is now resolved just as components and languages etc. making it consistent. If you have your own custom data formats you need to create the magic file under `META-INF/services/org/apache/camel/dataformat` which is a file with the data format name. See for example `camel-jackson` for an example how to do that.

camel-http

`HttpOperationFailedException` has been changed.

Core API

CamelContext

CamelContext have renamed the method `getLifecycleStrategy` to `getLifecycleStrategies` and now returns a List.

Exchange

The exchange id has been changed to use the `java.util.UUID` instead. This allows Camel to run on Google App Engine.

Management

The API in `org.apache.camel.spi.ManagementNamingStrategy` had its methods renaming and method signature adjusted to accommodate the big theme for Camel which was overhaul of management (JMX).

SPI

The `TraceableUnitOfWork` has been renamed to `TracedRouteNote` which is accessible from `UnitOfWork`.

DeadLetterChannel

The `deadLetterUri` uri now only accepts a single endpoint which is prechecked on startup. You cannot any longer using multiple endpoints separated by comma. That was in fact an internal hidden feature because a recipient list was used beforehand to produce to the dead letter queue. Adding the precheck removed the recipient list. You can use Direct and route to a route which can then use a recipient list in case you really need this.

Client API

Removed classes

- `org.apache.camel.processor.CompositeProcessor` was never used
- `org.apache.camel.impl.ProducerTemplateProcessor` was never used
- `org.apache.camel.impl.NoPolicy` was never used
- `org.apache.camel.spring.handler.LazyLoadingBeanDefinitionParser` was never used
- `org.apache.camel.spring.handler.ScriptDefinitionParser` was never used
- `org.apache.camel.spring.remoting.SendBeforeInterceptor` was never used
- `org.apache.camel.spring.spi.SpringConverters` was never used

Moved classes

- `org.apache.camel.PollingConsumerPollStrategy` is moved to `org.apache.camel.spi.PollingConsumerPollStrategy`
- `org.apache.camel.impl.scan.PatternBasedPackageScanFilter` is moved to `org.apache.camel.spring.PatternBasedPackageScanFilter` in the **camel-spring** jar.

Removed methods

- `getSingletonEndpoints` method have been removed from various classes in the `camel-test` component.

Removed in Spring XML

The option `shouldStartContext` on <camelContext> have been removed. Its replaced with `autoStartup` which you can read more about here Configuring route startup ordering and autostartup.

@Bean

@Bean now uses a `ref` attribute to refer to the bean name. Also added a new `method` attribute to set the method to use (optional).
You need to upgrade your code to use these attributes as the default `value` have been removed. We assume only a very few people knew and uses this @Bean annotation.

Direct component

The option `allowMultipleConsumers` has been removed as it did not make sense to have concurrent consumers on a direct endpoint.

Known Issues

Routing Slip EIP does not break when Exchange is failed, see CAMEL-2245.

Using <transacted/> in Spring XML does not work properly when using route scoped <onException> and/or <onCompletion>. Using context scoped works. And the Java DSL does not have this flaw. See CAMEL-2336.

When using the Recipient List together with MINA endpoints Camel will over time hold on to memory which could lead to OutOfMemoryErrors. See more at: CAMEL-2484

See known issues from previous releases.

Important changes to consider when upgrading

Getting the Distributions

Binary Distributions

Description	Download Link	PGP Signature file of download
Windows Distribution	apache-camel-2.1.0.zip	apache-camel-2.1.0.zip.asc
Unix/Linux/Cygwin Distribution	apache-camel-2.1.0.tar.gz	apache-camel-2.1.0.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 Mozilla Firefox

Source Distributions

Description	Download Link	PGP Signature file of download
Source for Windows	apache-camel-2.1.0-src.zip	apache-camel-2.1.0-src.zip.asc
Source for Unix/Linux/Cygwin	apache-camel-2.1.0-src.tar.gz	apache-camel-2.1.0-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>2.1.0</version>
</dependency>
```

SVN Tag Checkout

```
svn co http://svn.apache.org/repos/asf/camel/tags/camel-2.1.0
```

Changelog

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

- JIRA Release notes for 2.1.0