



Red Hat build of Apache Camel K 1.10.5

Release Notes for Red Hat build of Apache Camel K 1.10.5

What's new in Red Hat build of Apache Camel K

Red Hat build of Apache Camel K 1.10.5 Release Notes for Red Hat build of Apache Camel K 1.10.5

What's new in Red Hat build of Apache Camel K

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Describes the Red Hat build of Apache Camel K product and provides the latest details on what's new in this release.

Table of Contents

CHAPTER 1. CAMEL K RELEASE NOTES	3
1.1. CAMEL K FEATURES	3
1.2. SUPPORTED CONFIGURATIONS	3
1.2.1. Camel K Operator metadata	3
1.3. IMPORTANT NOTES	4
1.4. SUPPORTED CAMEL QUARKUS EXTENSIONS	4
1.4.1. Supported Camel Quarkus connector extensions	5
1.4.2. Supported Camel Quarkus dataformat extensions	7
1.4.3. Supported Camel Quarkus language extensions	8
1.4.4. Supported Camel K traits	8
1.5. SUPPORTED KAMELETS	9
1.6. CAMEL K KNOWN ISSUES	13
1.7. CAMEL K FIXED ISSUES	14
1.7.1. Bugs resolved in Camel K 1.10.5.redhat-00016	14

CHAPTER 1. CAMEL K RELEASE NOTES

Camel K is a lightweight integration framework built from Apache Camel K that runs natively in the cloud on OpenShift. Camel K is specifically designed for serverless and microservice architectures. You can use Camel K to instantly run integration code written in Camel Domain Specific Language (DSL) directly on OpenShift.

Using Camel K with OpenShift Serverless and Knative, containers are automatically created only as needed and are autoscaled under load up and down to zero. This removes the overhead of server provisioning and maintenance and enables you to focus instead on application development.

Using Camel K with OpenShift Serverless and Knative Eventing, you can manage how components in your system communicate in an event-driven architecture for serverless applications. This provides flexibility and creates efficiencies using a publish/subscribe or event-streaming model with decoupled relationships between event producers and consumers.

1.1. CAMEL K FEATURES

The Camel K provides cloud-native integration with the following main features:

- Knative Serving for autoscaling and scale-to-zero
- Knative Eventing for event-driven architectures
- Performance optimizations using Quarkus runtime by default
- Camel integrations written in Java or YAML DSL
- Monitoring of integrations using Prometheus in OpenShift
- Quickstart tutorials
- Kamelet Catalog for connectors to external systems such as AWS, Jira, and Salesforce
- Support for Timer and Log Kamelets
- Support for IBM MQ connector
- Support for Oracle 19 database

1.2. SUPPORTED CONFIGURATIONS

For information about Camel K supported configurations, standards, and components, see the following Customer Portal articles:

- [Camel K Supported Configurations](#)
- [Camel K Component Details](#)

1.2.1. Camel K Operator metadata

The Camel K includes updated Operator metadata used to install Camel K from the OpenShift OperatorHub. This Operator metadata includes the Operator bundle format for release packaging, which is designed for use with OpenShift Container Platform 4.6 or later.

Additional resources

- [Operator bundle format in the OpenShift documentation](#) .

1.3. IMPORTANT NOTES

Important notes for the Red Hat Integration - Camel K release:

Removing support of metering labels from Red Hat Integration - Camel K

Metering labels for Camel K Operator and pods are no longer supported.

Security update for Red Hat Integration - Camel K

For details on how to apply this update, see [How do I apply package updates to my RHEL system?](#)



NOTE

You must apply all the previously release Errata upgrades to your system before applying this security update.

Support to run Camel K on ROSA

Camel K is now supported to run on Red Hat OpenShift Service on AWS (ROSA).

Support for IBM MQ source connector in Camel K

IBM MQ source connector kamelet is added to latest Camel K.

Support for Oracle 19

Oracle 19 is now supported in Camel K. Refer [Supported configurations](#) page for more information.

Using Camel K CLI commands on Windows machine

When using `kamel cli` commands on Windows machine, the path in the **resource** option in the command must use linux format. For example:

```
//Windows path
kamel run file.groovy --dev --resource file:C:\user\folder\tempfile@/tmp/file.txt
```

```
//Must be converted to
kamel run file.groovy --dev --resource file:C:/user/folder/tempfile@/tmp/file.txt
```

Red Hat Integration - Camel K Operator image size is increased

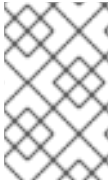
Since Red Hat Integration - Camel K 1.10.5.redhat-00016, the size of the Camel K Operator image is doubled.

Accepted Camel case notations in YAML DSL

Since Red Hat Integration - Camel K 1.10.5.redhat-00016, the YAML DSL will accept camel case notation (i.e, **setBody**) as well as snake case (i.e **set-body**). Please note that there are some differences in the syntax as schema is subject to changes within Camel versions.

1.4. SUPPORTED CAMEL QUARKUS EXTENSIONS

This section lists the Camel Quarkus extensions that are supported for this release of Camel K (only when used inside a Camel K application).

**NOTE**

These Camel Quarkus extensions are supported only when used inside a Camel K application. These Camel Quarkus extensions are not supported for use in standalone mode (without Camel K).

1.4.1. Supported Camel Quarkus connector extensions

The following table shows the Camel Quarkus connector extensions that are supported for this release of Camel K (only when used inside a Camel K application).

Name	Package
Attachments	camel-quarkus-attachments
AWS DynamoDB	camel-quarkus-aws-ddb
AWS 2 Kinesis	camel-quarkus-aws2-kinesis
AWS 2 Lambda	camel-quarkus-aws2-lambda
AWS 2 S3 Storage Service	camel-quarkus-aws2-s3
AWS 2 Simple Notification System (SNS)	camel-quarkus-aws2-sns
AWS 2 Simple Queue Service (SQS)	camel-quarkus-aws2-sqs
Azure Storage Blob (Technology Preview)	camel-quarkus-azure-storage-blob
Azure Storage Queue (Technology Preview)	camel-quarkus-azure-storage-queue
Bean	camel-quarkus-bean
Bean Validator	camel-quarkus-bean-validator
Browse	camel-quarkus-browse
Cassandra CQL	camel-quarkus-cassandraql
Core	camel-quarkus-core
Cron	camel-quarkus-cron
CXF	camel-quarkus-cxf-soap
Dataformat	camel-quarkus-dataformat
Direct	camel-quarkus-direct

Name	Package
File	camel-quarkus-file
FHIR	camel-quarkus-fhir
FTP	camel-quarkus-ftp
Google BigQuery	camel-quarkus-google-bigquery
Google Pubsub	camel-quarkus-google-pubsub
HTTP	camel-quarkus-http
Infinispan	camel-quarkus-infinispan
Jira	camel-quarkus-jira
JMS	camel-quarkus-jms
JPA	camel-quarkus-jpa
JTA	camel-quarkus-jta
Kafka	camel-quarkus-kafka
Kamelet	camel-quarkus-kamelet
Kubernetes	camel-quarkus-kubernetes
Log	camel-quarkus-log
Mail	camel-quarkus-mail
MicroProfile Fault Tolerance	camel-quarkus-microprofile-fault-tolerance
MicroProfile Health	camel-quarkus-microprofile-health
MicroProfile Metrics	camel-quarkus-microprofile-metrics
MLLP	camel-quarkus-mllp
Mock	camel-quarkus-mock
MongoDB	camel-quarkus-mongodb
Netty	camel-quarkus-netty

Name	Package
OpenAPI Java	camel-quarkus-openapi-java
Paho	camel-quarkus-camel-quarkus-paho
Paho MQTT 5	camel-quarkus-paho-mqtt5
Platform HTTP	camel-quarkus-platform-http
Quartz	camel-quarkus-quartz
Rest	camel-quarkus-rest
REST OpenApi	camel-quarkus-rest-openapi
Salesforce	camel-quarkus-salesforce
SEDA	camel-quarkus-seda
Slack	camel-quarkus-slack
SQL	camel-quarkus-sql
Telegram	camel-quarkus-telegram
Timer	camel-quarkus-timer
Validator	camel-quarkus-validator
XQuery	camel-quarkus-saxon
Zip File	camel-quarkus-zipfile

1.4.2. Supported Camel Quarkus dataformat extensions

The following table shows the Camel Quarkus dataformat extensions that are supported for this release of Camel K (only when used inside a Camel K application).

Name	Package
Avro	camel-quarkus-avro
Avro Jackson	camel-quarkus-jackson-avro
Bindy (for CSV)	camel-qaurkus-bindy

Name	Package
HL7	camel-quarkus-hl7
Jackson	camel-quarkus-jackson
JacksonXML	camel-quarkus-jacksonxml
JAXB	camel-quarkus-jaxb
JSON Gson	camel-quarkus-gson
Protobuf Jackson	camel-quarkus-jackson-protobuf
SOAP dataformat	camel-quarkus-soap

1.4.3. Supported Camel Quarkus language extensions

In this release, Camel K supports the following Camel Quarkus language extensions (for use in Camel expressions and predicates):

- Bean method
- Constant
- ExchangeProperty
- File
- Header
- HL7 Terser
- Ref
- Simple
- Tokenize
- JsonPath
- XPath
- XQuery

1.4.4. Supported Camel K traits

In this release, Camel K supports the following Camel K traits.

- Builder trait
- Camel trait

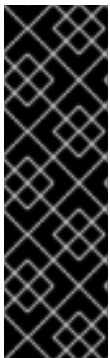
- Container trait
- Dependencies trait
- Deployer trait
- Deployment trait
- Environment trait
- Jvm trait
- Kamelets trait
- Owner trait
- Platform trait
- Pull Secret trait
- Prometheus trait
- Quarkus trait
- Route trait
- Service trait
- Error Handler trait

1.5. SUPPORTED KAMELETS

The following table lists the kamelets that are provided as OpenShift resources when you install the Camel K operator.

For details about these kamelets, go to: <https://github.com/openshift-integration/kamelet-catalog/tree/kamelet-catalog-1.8>

For information about how to use kamelets to connect applications and services, see https://access.redhat.com/documentation/en-us/red_hat_integration/2022.q3/html-single/integrating_applications_with_kamelets.



IMPORTANT

Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production.

These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see <https://access.redhat.com/support/offerings/techpreview>.

Table 1.1. Kamelets provided with the Camel K operator

Kamelet	File name	Type (Sink, Source, Action)
Ceph sink	ceph-sink.kamelet.yaml	Sink
Ceph Source	ceph-source.kamelet.yaml	Source
Jira Add Comment sink	jira-add-comment-sink.kamelet.yaml	Sink
Jira Add Issue sink	jira-add-issue-sink.kamelet.yaml	Sink
Jira Transition Issue sink	jira-transition-issue-sink.kamelet.yaml	Sink
Jira Update Issue sink	jira-update-issue-sink.kamelet.yaml	Sink
Avro Deserialize action	avro-deserialize-action.kamelet.yaml	Action (data conversion)
Avro Serialize action	avro-serialize-action.kamelet.yaml	Action (data conversion)
AWS DynamoDB sink	aws-ddb-sink.kamelet.yaml	Sink
AWS Redshift sink	aws-redshift-sink.kamelet.yaml	Sink
AWS 2 Kinesis sink	aws-kinesis-sink.kamelet.yaml	Sink
AWS 2 Kinesis source	aws-kinesis-source.kamelet.yaml	Source
AWS 2 Lambda sink	aws-lambda-sink.kamelet.yaml	Sink
AWS 2 Simple Notification System sink	aws-sns-sink.kamelet.yaml	Sink
AWS 2 Simple Queue Service sink	aws-sqs-sink.kamelet.yaml	Sink
AWS 2 Simple Queue Service source	aws-sqs-source.kamelet.yaml	Source
AWS 2 Simple Queue Service FIFO sink	aws-sqs-fifo-sink.kamelet.yaml	Sink
AWS 2 S3 sink	aws-s3-sink.kamelet.yaml	Sink
AWS 2 S3 source	aws-s3-source.kamelet.yaml	Source
AWS 2 S3 Streaming Upload sink	aws-s3-streaming-upload-sink.kamelet.yaml	Sink

Kamelet	File name	Type (Sink, Source, Action)
Azure Storage Blob Source (Technology Preview)	azure-storage-blob-source.kamelet.yaml	Source
Azure Storage Blob Sink (Technology Preview)	azure-storage-blob-sink.kamelet.yaml	Sink
Azure Storage Queue Source (Technology Preview)	azure-storage-queue-source.kamelet.yaml	Source
Azure Storage Queue Sink (Technology Preview)	azure-storage-queue-sink.kamelet.yaml	Sink
Cassandra sink	cassandra-sink.kamelet.yaml	Sink
Cassandra source	cassandra-source.kamelet.yaml	Source
Extract Field action	extract-field-action.kamelet.yaml	Action
FTP sink	ftp-sink.kamelet.yaml	Sink
FTP source	ftp-source.kamelet.yaml	Source
Has Header Key Filter action	has-header-filter-action.kamelet.yaml	Action (data transformation)
Hoist Field action	hoist-field-action.kamelet.yaml	Action
HTTP sink	http-sink.kamelet.yaml	Sink
Insert Field action	insert-field-action.kamelet.yaml	Action (data transformation)
Insert Header action	insert-header-action.kamelet.yaml	Action (data transformation)
Is Tombstone Filter action	is-tombstone-filter-action.kamelet.yaml	Action (data transformation)
Jira source	jira-source.kamelet.yaml	Source
JMS sink	jms-amqp-10-sink.kamelet.yaml	Sink

Kamelet	File name	Type (Sink, Source, Action)
JMS source	jms-amqp-10-source.kamelet.yaml	Source
JMS IBM MQ sink	jms-ibm-mq-sink.kamelet.yaml	Sink
JMS IBM MQ source	jms-ibm-mq-source.kamelet.yaml	Source
JSON Deserialize action	json-deserialize-action.kamelet.yaml	Action (data conversion)
JSON Serialize action	json-serialize-action.kamelet.yaml	Action (data conversion)
Kafka sink	kafka-sink.kamelet.yaml	Sink
Kafka source	kafka-source.kamelet.yaml	Source
Kafka Topic Name Filter action	topic-name-matches-filter-action.kamelet.yaml	Action (data transformation)
Log sink (for development and testing purposes)	log-sink.kamelet.yaml	Sink
MariaDB sink	mariadb-sink.kamelet.yaml	Sink
Mask Fields action	mask-field-action.kamelet.yaml	Action (data transformation)
Message TimeStamp Router action	message-timestamp-router-action.kamelet.yaml	Action (router)
MongoDB sink	mongodb-sink.kamelet.yaml	Sink
MongoDB source	mongodb-source.kamelet.yaml	Source
MySQL sink	mysql-sink.kamelet.yaml	Sink
PostgreSQL sink	postgresql-sink.kamelet.yaml	Sink
Predicate filter action	predicate-filter-action.kamelet.yaml	Action (router/filter)

Kamelet	File name	Type (Sink, Source, Action)
Protobuf Deserialize action	protobuf-deserialize-action.kamelet.yaml	Action (data conversion)
Protobuf Serialize action	protobuf-serialize-action.kamelet.yaml	Action (data conversion)
Regex Router action	regex-router-action.kamelet.yaml	Action (router)
Replace Field action	replace-field-action.kamelet.yaml	Action
Salesforce Create	salesforce-create-sink.kamelet.yaml	Sink
Salesforce Delete	salesforce-delete-sink.kamelet.yaml	Sink
Salesforce Update	salesforce-update-sink.kamelet.yaml	Sink
SFTP sink	sftp-sink.kamelet.yaml	Sink
SFTP source	sftp-source.kamelet.yaml	Source
Slack source	slack-source.kamelet.yaml	Source
SQL Server Database sink	sqlserver-sink.kamelet.yaml	Sink
Telegram source	telegram-source.kamelet.yaml	Source
Throttle action	throttle-action.kamelet.yaml	Action
Timer source (for development and testing purposes)	timer-source.kamelet.yaml	Source
TimeStamp Router action	timestamp-router-action.kamelet.yaml	Action (router)
Value to Key action	value-to-key-action.kamelet.yaml	Action (data transformation)

1.6. CAMEL K KNOWN ISSUES

The following known issues apply to the Camel K:

[ENTESB-15306](#) - CRD conflicts between Camel K and Fuse Online

If an older version of Camel K has ever been installed in the same OpenShift cluster, installing Camel K from the OperatorHub fails due to conflicts with custom resource definitions. For example, this includes older versions of Camel K previously available in Fuse Online.

For a workaround, you can install Camel K in a different OpenShift cluster, or enter the following command before installing Camel K:

```
$ oc get crds -l app=camel-k -o json | oc delete -f -
```

ENTESB-15858 - Added ability to package and run Camel integrations locally or as container images

Packaging and running Camel integrations locally or as container images is not currently included in the Camel K and has community-only support.

For more details, see the [Apache Camel K community](#).

ENTESB-16477 - Unable to download jira client dependency with productized build

When using Camel K operator, the integration is unable to find dependencies for jira client. The work around is to add the atlassian repo manually.

```
apiVersion: camel.apache.org/v1
kind: IntegrationPlatform
metadata:
  labels:
    app: camel-k
    name: camel-k
spec:
  configuration:
    - type: repository
      value: <atlassian repo here>
```

ENTESB-17033 - Camel-K ElasticsearchComponent options ignored

When configuring the Elasticsearch component, the Camel K ElasticsearchComponent options are ignored. The work around is to add `getContext().setAutowiredEnabled(false)` when using the Elasticsearch component.

ENTESB-17061 - Can't run mongo-db-source kamelet route with non-admin user - Failed to start route mongodb-source-1 because of null

It is not possible to run **mongo-db-source kamelet** route with non-admin user credentials. Some part of the component require admin credentials hence it is not possible run the route as a non-admin user.

1.7. CAMEL K FIXED ISSUES

The following sections list the issues that have been fixed in Red Hat Integration - Camel K 1.10.5.redhat-00016:

- [Section 1.7.1, "Bugs resolved in Camel K 1.10.5.redhat-00016"](#)

1.7.1. Bugs resolved in Camel K 1.10.5.redhat-00016

The following table lists the resolved bugs in Camel K 1.10.5.redhat-00016:

The following table lists the resolved bugs in Camel K 1.10.5.redhat-00016:

Table 1.2. Camel K 1.10.5.redhat-00016 Resolved Bugs

Issue	Description
CMLK-954	CVE-2023-34462 netty: SniHandler 16MB allocation leads to OOM [rhint-camel-k-1]
CMLK-911	CVE-2023-34455 snappy-java: Unchecked chunk length leads to DoS [rhint-camel-k-1]
CMLK-1386	CVE-2023-5072 JSON-java: parser confusion leads to OOM [rhint-camel-k-1.10]

The following table lists the resolved bugs in Camel K 1.10.4.redhat-00007.

Table 1.3. Camel K 1.10.4.redhat-00007 Resolved Bugs

Issue	Description
CMLK-1313	[Major Incident] CVE-2023-44487 undertow: HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable to a DDoS attack (Rapid Reset Attack) [rhint-camel-k-1.10]
CMLK-1312	[Major Incident] CVE-2023-44487 netty-codec-http2: HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable to a DDoS attack (Rapid Reset Attack) [rhint-camel-k-1.10]
CMLK-243	Camel-K uses io.quarkus.quarkus-maven-plugin but we should use com.quarkus.redhat.platform:quarkus-maven-plugin
CMLK-1314	Upgrade x/net to version 0.17.0

Camel K 1.10.3.redhat-00001 release, addresses underlying base images only, product is not changed.

The following table lists the resolved bugs in Camel K 1.10.2.redhat-00002:

Table 1.4. Camel K 1.10.2.redhat-00002 Resolved Bugs

Issue	Description
CMLK-1238	[Major Incident] CVE-2023-4853 quarkus-vertx-http: quarkus: HTTP security policy bypass [rhint-camel-k-1.10]