

# LibRdKafka Version 2.12-1 for VSI OpenVMS

## Release Notes

**Publication Date:** March 2026

**Operating Systems:** VSI OpenVMS IA-64 Version 8.4-2L1 or higher  
VSI OpenVMS x86-64 Version 9.2-3 or higher

**Kit Names:** VSI-I64VMS-LIBRDKAFKA-V0212-1-1.PCSI  
VSI-X86VMS-LIBRDKAFKA-V0212-1-1.PCSI

## Table of Contents

1. Introduction .....	3
2. Acknowledgements .....	3
3. What's New in This Release .....	3
4. Requirements .....	3
5. Recommended Reading .....	4
6. Installing the Kit .....	4
6.1. Post-Installation Steps .....	5
6.2. Privileges and Quotas .....	5
7. Sample Applications .....	5

# 1. Introduction

Thank you for your interest in this port of the LibRdKafka API to VSI OpenVMS. The current release of LibRdKafka for OpenVMS is based on the 2.12.1 distribution.

LibRdKafka (<https://github.com/edenhill/librdkafka>) is a C/C++ library implementation of the Apache Kafka protocol and includes both producer and consumer support. Apache Kafka (<https://kafka.apache.org/>) is an open source message broker written in Java and Scala that aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. LibRdKafka provides a portable C/C++ API that can be used by client applications to interact with the message broker.

This OpenVMS port of LibRdKafka includes all functionality provided by the open source 2.12.1 release.

Additional information about LibRdKafka and Apache Kafka can be found at <https://github.com/edenhill/librdkafka> and <https://kafka.apache.org/>.

## 2. Acknowledgements

VMS Software Inc. would like to acknowledge the work of Magnus Edenhill for the creation and ongoing evolution of LibRdKafka.

## 3. What's New in This Release

For details on new features and bug fixes included in this release, review the open source project home page (<https://github.com/edenhill/librdkafka>).

It should be noted that this release of LibRdKafka represents a significant version jump from the previous release of this library for VSI OpenVMS, with many additions to the API, as well as deprecations (in part driven by enhancements to Apache Kafka itself). A significant enhancement available in this release of LibRdKafka for VSI OpenVMS is the inclusion of support for authentication using SASL/SCRAM.

This release of LibRdKafka for OpenVMS also includes the KCAT utility – a generic producer and consumer command line utility for Apache Kafka (for details, see <https://github.com/edenhill/kcat>).

## 4. Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. While it is highly likely that you will have no problems installing and using this kit on systems running higher versions of the products below, VSI does not recommend running the older versions.

- One of the following versions of VSI OpenVMS:
  - VSI OpenVMS IA-64 Version 8.4-2L1 or higher
  - VSI OpenVMS x86-64 Version 9.2-3 or higher
- VSI TCP/IP
- C or C++ compiler for application development

It is assumed that the reader has a good knowledge of OpenVMS and software development in the OpenVMS environment.

## 5. Recommended Reading

It is recommended that developers read the documentation available in the LibRdKafka GitHub repository (<https://github.com/edenhill/librdkafka>) and examine the sample programs provided with the VSI LibRdKafka for OpenVMS kit before using the software.

## 6. Installing the Kit

The kit is provided as an OpenVMS PCSI kit that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL LIBRDKAFKA
```

The installation will then proceed as follows (output may differ slightly from that shown depending on architecture and other factors):

```
Performing product kit validation of signed kits ...
```

```
The following product has been selected:
    VSI X86VMS LIBRDKAFKA V2.12-1          Layered Product
```

```
Do you want to continue? [YES]
```

```
Configuration phase starting ...
```

```
You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.
```

```
Configuring VSI X86VMS LIBRDKAFKA V2.12-1
```

```
    VMS Software Inc. & Magnus Edenhill
```

```
* This product does not have any configuration options.
```

```
Execution phase starting ...
```

```
The following product will be installed to destination:
    VSI X86VMS LIBRDKAFKA V2.12-1          DISK$X86SYS:[VMS$COMMON.]
```

```
Portion done: 0%...40%...90%...100%
```

```
The following product has been installed:
    VSI X86VMS LIBRDKAFKA V2.12-1          Layered Product
```

```
VSI X86VMS LIBRDKAFKA V2.12-1
```

```
Post-installation tasks are required.
```

```
To start librdkafka at system boot time, add the following lines
to SYS$MANAGER:SYSTARTUP_VMS.COM:
```

```
    $ file := SYS$STARTUP:LIBRDKAFKA$STARTUP.COM
    $ if f$search("''file'") .nes. "" then @'file'
```

To stop Kafka at system shutdown, add the following lines to SYS\$MANAGER:SYSHUTDOWN.COM:

```
$ file := SYS$STARTUP:LIBRDKAFKA$SHUTDOWN.COM
$ if f$search("'"file'") .nes. "" then @'file'
```

## 6.1. Post-Installation Steps

After the installation has successfully completed, include the commands displayed at the end of the installation procedure into the SYSTARTUP\_VMS.COM file to ensure that the required logical names are defined system-wide at start-up.

In addition to the logical name LIBRDKAFKA\$ROOT (that points to root directory of the LibRdKafka installation tree), the logical name LIBRDKAFKA\$SHR is also defined. This logical name points to the shareable image LIBRDKAFKA\$ROOT:[LIB]LIBRDKAFKA\$SHR.EXE that can be linked with the application code. Alternatively, it is possible to statically link the application code with the object library LIBRDKAFKA\$ROOT:[LIB]LIBRDKAFKA.OLB.

From a development perspective, the symbols in the shareable image and object library are mixed-case. To ensure that the symbols are correctly resolved when linking, application developers must use the C and C++ compiler option **/NAMES=(AS\_IS,SHORTENED)** or include the appropriate `#pragma` directives in their code. The application code must also include one or more of the header files found in LIBRDKAFKA\$ROOT:[INCLUDE].

## 6.2. Privileges and Quotas

Generally speaking, applications developed using LibRdKafka do not have any special quota or privilege requirements, although a high BYTLM is recommended. The following quotas should be more than adequate for most purposes:

Maxjobs:	0	Fillm:	256	Bytln:	128000
Maxacctjobs:	0	Shrfillm:	0	Pbytln:	0
Maxdetach:	0	BIolm:	150	JTquota:	4096
Prclm:	50	DIolm:	150	WSdef:	4096
Prio:	4	ASTlm:	300	WSquo:	8192
Queprio:	4	TQElm:	100	WSeextent:	16384
CPU:	(none)	Enqlm:	4000	Pgflquo:	256000

## 7. Sample Applications

The directory LIBRDKAFKA\$ROOT:[EXAMPLES] contains several simple example programs that can be used as a learning tool or as a source of inspiration. These examples can be compiled and linked using the provided build procedure (SAMPLES.COM). All examples are linked with the LIBRDKAFKA\$SHR shareable image, although they could also be statically linked with the corresponding object library.

---

### Note

If you are using the API with C++ on VSI OpenVMS x86-64, make sure to use the 32-bit pointers by specifying **/POINTER=32** with the **CXX** command (by default, 64-bit pointers are used).

---