

LibRdKafka for OpenVMS I64

February 2020

1. Introduction

Thank you for your interest in this port of the LibRdKafka API to OpenVMS. The current release of LibRdKafka for OpenVMS is based on the 1.3.0 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 1.3.0 release and a wrapper API that makes it easier to use LibRdKafka with OpenVMS programming languages other than C/C++, such as COBOL, FORTRAN, Pascal, and BASIC.

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

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 of new features and bug fixes included in this release, please read the Open Source project home page (<https://github.com/edenhill/librdkafka>) and links contained therein, and review the project repository change logs.

One significant enhancement available in this release of LibRdKafka for OpenVMS is the inclusion of support for authentication using SASL/SCRAM.

SSL/TLS support is statically linked into the shareable images `librdkafka$shr.exe` and uses OpenSSL 1.1.1b.

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

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 the kit on systems running higher versions of the products listed, we cannot say for sure that you will be so lucky if your system is running older versions.

- VSI OpenVMS Version 8.4-1H1 or higher

- VSI TCP/IP, HPE TCP/IP Services for OpenVMS, or MultiNet TCP/IP
- C compiler - VSI C V7.4-001 or comparable
- C++ compiler – HP C++ V7.3-023 or comparable

In addition to the above requirements, it is assumed that the reader has a good knowledge of OpenVMS and of software development in the OpenVMS environment.

5. Recommended reading

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

6. Installing the kit

The kit is provided as an OpenVMS PCSI kit (VSI-I64VMS-LIBRDKAFKA-V0103-0-1.PCSI) 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):

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

```
The following product has been selected:
```

```
VSI I64VMS LIBRDKAFKA V1.3-0          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 I64VMS LIBRDKAFKA V1.3-0
```

```
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 I64VMS LIBRDKAFKA V1.3-0          DISK$I64SYS:[VMS$COMMON.]
```

```
Portion done: 0%...30%...70%...90%...100%
```

```
The following product has been installed:
```

```
VSI I64VMS LIBRDKAFKA V1.3-0          Layered Product
```

```
VSI I64VMS LIBRDKAFKA V1.3-0
```

```
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 SYSTARTUP_VMS.COM to ensure that the logical names required in order for users to use the software are defined system-wide at start-up.

In addition to the logical name LIBRDKAFKA\$ROOT (which 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, which can be linked with application code. Alternatively, it is possible to statically link application code with the object library LIBRDKAFKA\$ROOT:[LIB]LIBRDKAFKA.OLB.

From a development perspective, it should be noted that symbols in the shareable image and object library for the C/C++ API are mixed-case, and application developers must therefore use the compiler option /NAMES=(AS_IS) or include in their code appropriate directives to ensure that symbols are correctly resolved when linking. Developers will also need to include in their application code one or more of the header files found in LIBRDKAFKA\$ROOT:[INCLUDE].

6.2. *Privileges and quotas*

Generally speaking there are no special quota or privilege requirements for applications developed using LibRdKafka, although a high BYTLM is recommended. The following quotas should be more than adequate for most purposes:

Maxjobs:	0	Fillm:	256	Bytlm:	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	WSextent:	16384
CPU:	(none)	Enqlm:	4000	Pgflquo:	256000

6.3. *Installing in an alternative location*

By default the software will be installed in SYS\$SYSDEVICE:[VMS\$COMMON]. If you wish to install the software in an alternative location this can be achieved using the /DESTINATION qualifier with the PRODUCT INSTALL command to specify the desired location; however it is important to note that an additional manual step will then be required to complete the installation. Specifically, when an alternative destination is specified, the start-up and shutdown procedures (LIBRDKAFKA\$STARTUP.COM and LIBRDKAFKA\$SHUTDOWN.COM) will be placed into a subdirectory [.SYS\$STARTUP] residing under the specified destination

directory. If you wish to run these files from your standard `SYS$STARTUP` directory they will need to be copied from the destination subdirectory into your systems `SYS$STARTUP` directory.

7. Sample applications

The directory `LIBRDKAFKA$ROOT:[EXAMPLES]` contains several simple example programs that can be used to learn about the API or as a source of inspiration for the development of new applications. These examples can be compiled and linked using the provided build procedure (`SAMPLES.COM`). Note that examples are provided for several programming languages, including C, C++, FORTRAN, and COBOL. All examples are linked with the `LIBRDKAFKA$SHR` shareable image; however they could alternatively be statically linked with the corresponding object library.

8. What's missing?

The supplied kit for OpenVMS includes all functionality supported by version 1.3.0 of the LibRdKafka API (<https://github.com/edenhill/librdkafka>).

As noted previously, this port of LibRdKafka to OpenVMS includes a language-agnostic API that makes it easier to write applications using 3GL languages such as COBOL and FORTRAN. However, the scope of the language-agnostic API is limited and represents only a small subset of the functionality that is provided by the C and C++ API's. It is anticipated that additional functionality will be added to the language-agnostic API in future releases.