VMS Software

cURL and libcURL V8.13-0 for OpenVMS

Release Notes

Table of Contents

Note

If you use GNV and wish to use cURL with GNV then it is recommended that you use the kit provided by VMSPORTS that is available at . If you do not use GNV, then it is recommended that you install the VSI kit, which is completely standalone and has no association with GNV. You cannot have both kits installed.

---

1. Introduction

Thank your for your interest in this port of cURL and libcURL to OpenVMS IA-64, Alpha, and x86-64. The current release of cURL and libcURL for OpenVMS is based on the 8.13.0 Open Source distribution.

cURL is a command line tool for getting or sending files using URL syntax and libcURL is an API that can be used by developers to implement cURL functionality in their application code. The API and command line utility support numerous web protocols, including HTTP, HTTPS, FTP, FTPS, LDAP, SFTP, HTTP/2, SMB, and others (see https://curl.haxx.se/ for more information).

2. 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.

- OpenVMS V8.4-2L1 or higher (IA-64 and Alpha), OpenVMS V9.2-3 or higher (x86-64)

- VSI TCP/IP

- C compiler or cross compiler for application development (optional)

- It is recommended that the software is installed on an ODS-5 file system.

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 (if intending to develop applications using LIBCURL).

3. Recommended Reading

It is recommended that users and developers review some of the excellent documentation and code examples available via the cURL website (https://curl.haxx.se/).

4. What's New in This Release?

This release of cURL for OpenVMS is based on the 8.13.0 Open Source distribution and has been built with OpenSSL 3.0. For a list of changes included in this version of cURL please see https://curl.se/changes.html#8_13_0. This release of cURL for VSI OpenVMS also includes large-file support, along with sftp, SMB, and HTTP/2 protocol support, and support for LDAP using the OpenLDAP client API.

This V8.13-0 release of cURL for VSI OpenVMS includes 64-bit versions of both the cURL utility and the libcURL API in addition to the standard 32-bit versions, providing additional flexibility for

---

application developers, and making possible certain operations that require the dynamic allocation of more memory than is possible with the 32-bit API and utility implementations.

5. Installing the Kit

The kit is provided as an OpenVMS PCSI kit (VSI-I64VMS-CURL-V0813-0-1.PCSI for IA-64, VSI-AXPVMS-CURL-V0813-0-1.PCSI for Alpha, or VSI-X86VMS-CURL-V0813-0-1.PCSI for x86-64) that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL CURL
```

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

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

The following product has been selected:
    VSI I64VMS CURL V8.13-0              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 CURL V8.13-0

    VMS Software Inc.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    VSI I64VMS CURL V8.13-0              DISK$I64SYS:[VMS$COMMON.]

Portion done: 0%...40%...80%...90%...100%

The following product has been installed:
    VSI I64VMS CURL V8.13-0              Layered Product

VSI I64VMS CURL V8.13-0

    Post-installation tasks are required.
To start cURL at system boot time, add the following lines to
    SYS$MANAGER:SYSTARTUP_VMS.COM:

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

    To stop cURL at system shutdown, add the following lines to
    SYS$MANAGER:SYSHUTDWN.COM:

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

5.1. 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.

Users will then be able to use the cURL command line utility by defining a foreign command as follows:

```
$ CURL :== $CURL$ROOT:[BIN]CURL.EXE
```

If you wish to use the 64-bit pointer version of the cURL command line utility, simply use CURL64.EXE in place of CURL.EXE in the above command.

Generally speaking, there are no special quota or privilege requirements required for users wishing use the cURL command line utility or to develop applications using the cURL API. Although, it should be noted that some applications may require a high BYTLM quota for efficient operation.

In addition to defining the logical name CURL$ROOT (which points to the top-level directory of the installation), the start-up procedure defines the logical names LIBCURL$SHR and LIBCURL$SHR64, which point to the 32-bit and 64-bit pointer implementations of the cURL API shareable images, respectively. These shareable images can be linked with application code that uses the cURL API. In order to use the API, developers will also need to include in their projects the header file CURL.H from CURL$ROOT:[INCLUDE].

Developers may alternatively wish to link application code with the object library CURL$ROOT:[LIB]LIBCURL.OLB; (or the 64-bit pointer implementation of the library, LIBCURL64.OLB), however it will then also be necessary to link in various additional libraries; when linking with the shareable image this is not necessary, as these additional components are statically linked into the shareable image.

Developers should also note that symbols in the shareable image and object library are mixed case. When compiling code, developers should therefore use the C compiler option **/NAMES=(AS_IS,SHORTENED)** or include appropriate #pragma directives in their code to ensure that symbols are correctly resolved when linking. In addition, when working with the 64-bit pointer API implementation, it is important to ensure that all application code is compiled with **/POINTER_SIZE=64** (or appropriate #pragma statements are used to manage pointer size around calls into the cURL API).