



Centreon Clib Documentation

Release 1.4.2

Centreon

October 24, 2018

Centreon Clib is a common library for all Centreon products written in C/C++. It is licensed under the terms of the [Apache Software License Version 2](#).

Contents:

Release notes

1.1 Centreon Clib 1.0

1.1.1 What's new

First release

Centreon Clib is a common library for all Centreon products written in C/C++. This project provides high level implementation of many basic system mechanism. The target is to have a portable and powerful implementation.

Installation

Centreon recommends using its official packages from the Centreon Open Sources version available free of charge on our repository (ex CES). Most of Centreon endorsed software are available as RPM packages.

Alternatively, you can build and install your own version of this software by following the *Using sources*.

2.1 Using packages

Centreon provides RPM for its products through Centreon Open Sources (ex CES). Open source products are freely available from our repository. These packages are available for CentOS 6 or 7 architecture x86_64, which is the sole platform officially supported by Centreon.

2.1.1 Prerequisites

In order to use RPM from the COS repository, you have to install the appropriate repo file. Run the following command as privileged user

CentOS 6

Run the following commands as privileged user

```
$ wget http://yum.centreon.com/standard/3.4/el6/stable/noarch/RPMS/centreon-release-3.4-4.el6.noarch.rpm
$ yum install --nogpgcheck -y centreon-release-3.4-4.el6.noarch.rpm
$ rm -f centreon-release-3.4-4.el6.noarch.rpm
$ yum clean all
```

CentOS 7

Run the following commands as privileged user

```
$ wget http://yum.centreon.com/standard/3.4/el7/stable/noarch/RPMS/centreon-release-3.4-4.el7.centos.rpm
$ yum install --nogpgcheck -y centreon-release-3.4-4.el7.centos.noarch.rpm
$ rm -f centreon-release-3.4-4.el6.noarch.rpm
$ yum clean all
```

2.1.2 Install

Run the following commands as privileged user

```
$ yum install centreon-clib centreon-clib-devel
```

All dependencies are automatically installed from Centreon repositories.

2.2 Using sources

To build Centreon Clib, you will need the following external dependencies:

- a C++ compilation environment
- CMake (**>= 2.8**), a cross-platform build system

This program is compatible only with Unix-like platforms (Linux, FreeBSD, Solaris, ...).

2.2.1 Prerequisites

CentOS

Either use the Package Manager or the yum tool to install them. You should check package versions when necessary.

Required packages:

Software	Package Name	Description
C++ compilation environment	gcc gcc-c++ make	Mandatory tools to compile
CMake (>= 2.8)	cmake	Read the build script and prepare sources for compilation

1. Install basic compilation tools:

```
$ yum install gcc gcc-c++ make cmake
```

Debian/Ubuntu

In recent Debian/Ubuntu versions, necessary software is available as binary packages from distribution repositories. Either use the Package Manager or the apt-get tool to install them. You should check packages version when necessary.

Required packages:

Software	Package Name	Description
C++ compilation environment	build-essential	Mandatory tools to compile.
CMake (>= 2.8)	cmake	Read the build script and prepare sources for compilation.

1. Install compilation tools:

```
$ apt-get install build-essential cmake
```

OpenSUSE

In recent OpenSUSE versions, necessary software is available as binary packages from OpenSUSE repositories. Either use the Package Manager or the zypper tool to install them. You should check packages version when necessary.

Package required to build:

Software	Package Name	Description
C++ compilation environment CMake (>= 2.8)	gcc gcc-c++ make cmake	Mandatory tools to compile. Read the build script and prepare sources for compilation.

1. Install compilation tools

```
$ zypper install gcc gcc-c++ make cmake
```

2.2.2 Build

Get sources

Centreon Clib can be checked out from GitHub at <https://github.com/centreon/centreon-clib>. Currently, only the 1.4 branch works with Centreon components. On a Linux box with git installed this is just a matter of

```
$ git clone -b 1.4 https://github.com/centreon/centreon-clib
```

Or You can get the latest Centreon Clib's sources from its [download website](#) Once downloaded, extract it

```
$ tar xzf centreon-clib.tar.gz
```

Configuration

At the root of the project directory you'll find a build directory which holds build scripts. Generate the Makefile by running the following command

```
$ cd /path_to_centreon_clib/build
```

Your Centreon Clib can be tweaked to your particular needs using CMake's variable system. Variables can be set like this

```
$ cmake -D<variable1>=<value1> [-D<variable2>=<value2>] .
```

Here's the list of variables available and their description:

Variable	Description	Default value
WITH_PKGCONFIG_DIR	Define install pkg-config files.	\${WITH_PREFIX_LIB}/pkgconfig
WITH_PKGCONFIG_SCRIPT	Define install pkg-config files.	ON
WITH_PREFIX	Base directory for Centreon Clib installation. If other prefixes are expressed as relative paths, they are relative to this path.	/usr/local
WITH_PREFIX_INC	Define specific directory for Centreon Engine headers.	\${WITH_PREFIX}/include/centreon-engine
WITH_PREFIX_LIB	Define specific directory for Centreon Engine modules.	\${WITH_PREFIX}/lib/centreon-engine
WITH_SHARED_LIB	Create or not a shared library.	ON
WITH_STATIC_LIB	Create or not a static library.	OFF
WITH_TESTING	Build unit test.	OFF

Example

```
$ cmake \
  -DWITH_TESTING=0 \
  -DWITH_PREFIX=/usr \
  -DWITH_PREFIX_LIB=/usr/lib \
  -DWITH_PREFIX_INC=/usr/include/centreon-clib \
  -DWITH_SHARED_LIB=1 \
  -DWITH_STATIC_LIB=0 \
  -DWITH_PKGCONFIG_DIR=/usr/lib/pkgconfig .
```

At this step, the software will check for existence and usability of the prerequisites. If one cannot be found, an appropriate error message will be printed. Otherwise an installation summary will be printed.

Note: If you need to change the options you used to compile your software, you might want to remove the *CMake-Cache.txt* file that is in the *build* directory. This will remove cache entries that might have been computed during the last configuration step.

Compilation

Once properly configured, the compilation process is really simple:

```
$ make
```

And wait until compilation completes.

2.2.3 Install

Once compiled, the following command must be run as privileged user to finish installation

```
$ make install
```

And wait for its completion.