



# Centreon Perl Connector Documentation

*Release 1.1.2*

**Centreon**

November 24, 2015







Centreon Perl Connector is a free software from Centreon available under the Apache Software License version 2 (ASL 2.0). It speeds up execution of Perl scripts when used along Centreon Engine.

Contents:



---

## Release notes

---

### 1.1 Centreon Perl Connector 1.0

#### 1.1.1 What's new

##### First release

Centreon Perl Connector allow you to execute script perl with the embedded perl system, to improve performance.





---

## Installation

---

Centreon recommends using its official packages from the Centreon Enterprise Server (CES) repository. Most of Centreon's 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 Enterprise Server (CES). Open source products are freely available from our repository.

These packages have been successfully tested with CentOS 5 and RedHat 5.

#### 2.1.1 Prerequisites

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

```
$ wget http://yum.centreon.com/standard/2.2/ces-standard.repo -O /etc/yum.repos.d/ces-standard.repo
```

The repo file is now installed. Don't forget to cleanup

```
$ yum clean all
```

#### 2.1.2 Install

Run the following commands as privileged user

```
$ yum install centreon-connector-perl
```

All dependencies are automatically installed from Centreon repositories.

### 2.2 Using sources

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

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

- Centreon Clib, The centreon Core library.
- Perl, the perl library to use embedded perl.

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

## 2.2.1 Prerequisites

### CentOS

In CentOS you need to add manually cmake. After that you can install binary packages. Either use the Package Manager or the yum tool to install them. You should check packages version when necessary.

Package required to build:

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.
Centreon Clib (>= 1.0)	centreon-clib-devel	Core library used by Centreon Connector.
Perl	perl perl-devel (CentOS 6) perl-ExtUtils-Embed	Scripting language. Header perl (only for CentOS 6). Embedded perl (only for CentOS 6).

#### 1. Install basic compilation tools

```
$ yum install gcc gcc-c++ make perl
$ yum install perl-devel perl-ExtUtils-Embed # Only for CentOS 6
```

#### 2. Install Centreon repository

You need to install Centreon Enterprise Server (CES) repos file as explained *Prerequisites* to use some specific package version.

#### 3. Install cmake

```
$ yum install cmake
```

#### 4. Install Centreon Clib

See the Centreon Clib [documentation](#).

### 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.

Package required to build:

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.
Centreon Clib	centreon-clib-dev	Core library used by Centreon Connector.
Perl	libperl-dev	Scripting language.

#### 1. Install compilation tools

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

## 2. Install Centreon Clib

See the Centreon Clib [documentation](#).

## 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	gcc gcc-c++ make	Mandatory tools to compile.
CMake (>= 2.8)	cmake	Read the build script and prepare sources for compilation.
Centreon Clib	centreon-clib-devel	Core library used by Centreon Connector.
Perl	perl	Scripting language.

### 1. Install compilation tools

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

### 2. Install Centreon Clib

See the Centreon Clib [documentation](#).

## 2.2.2 Build

### Get sources

Centreon Perl Connector can be checked out from GitHub at <https://github.com/centreon/centreon-connectors> . The Perl connector sources reside in the perl subdirectory. On a Linux box with git installed this is just a matter of

```
$ git clone https://github.com/centreon/centreon-connectors
```

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

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

### Configuration

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

```
$ cd /path_to_centreon_connector/perl/build
```

Your Centreon Perl Connector 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_CENTREON_CLIB_INCLUDE_DIR	Set include directory path of centreon-clib include.	auto detection
WITH_CENTREON_CLIB_LIBRARIES	Set libraries for centreon-clib library to use.	auto detection
WITH_CENTREON_CLIB_LIBRARY_DIR	Set library directory for centreon-clib library directory (don't use it if you use WITH_CENTREON_CLIB_LIBRARIES).	auto detection
WITH_PREFIX	Base directory for Centreon Perl Connector installation. If other prefixes are expressed as relative paths, they are relative to this path.	/usr/local
WITH_PREFIX_BINARY	Define specific directory for Centreon Connector Perl binary.	\${WITH_PREFIX}/bin
WITH_TESTING	Enable generation of unit tests. They can later OFF be run by typing <i>make test</i> .	

#### Example

```
$ cmake \
  -DWITH_PREFIX=/usr \
  -DWITH_PREFIX_BINARY=/usr/lib/centreon-connector \
  -DWITH_TESTING=0 .
```

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 *CMakeCache.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.

## 3.1 Configuration

Centreon Perl Connector itself does not require any configuration. It should only be configured as a connector of Centreon Engine.

To execute Perl scripts with Centreon Perl Connector from Centreon Engine, one might configure commands that relates to Perl scripts. Such commands must only contain the path to the Perl script to execute followed by its arguments, just like one would on the command line. To make it simple, you just have to add a connector property to your command definition.

### 3.1.1 Binary arguments

These arguments are `centreon_connector_perl` options.

Short name	Long name	Description
-d	-debug	If this flag is specified, print all logs messages.
-h	-help	Print help and exit.
-v	-version	Print software version and exit.

Exemple:

```
define connector{
    connector_name centreon_connector_perl
    connector_line /usr/bin/centreon-connector/centreon_connector_perl
}

define command{
    command_name check_ping
    command_line $USER1$/check_ping.pl -H $HOSTADDRESS$
    connector centreon_connector_perl
}

define command{
    command_name check_disk
    command_line $USER1$/check_disk.pl -H $HOSTADDRESS$ -D $ARG1$
    connector centreon_connector_perl
}
```

## 3.2 Technical details

This article describes how Centreon Perl Connector allow much gain on Perl script execution.

First of all let's examine how a Perl script is executed traditionnally by Centreon Engine.

- Centreon Engine forks, creating a new separate process.
- This new process executes the `execve` syscall to run the Perl interpreter. This step does not create a new process.
- The Perl interpreter parse the Perl script.
- Perl script get executed.

With Centreon Engine, the same script get executed multiple times but with different arguments. Therefore we took advantage of this fact to efficiently parse all the scripts once and get them executed. This was only possible because of the `fork()`ing system of Unix-like platform. If you read the reference page on Wikipedia you indeed remarked that once `fork()`ed the old and the new process are identical. Centreon Connector Perl's steps to execute scripts are as follow.

- Centreon Engine creates a resident process of Centreon Connector Perl once
- For all Perl scripts execution requests are forwarded to this process when requested to execute a script, Centreon Perl Connector checks if this script has already been parsed if not it parses it using the Embedded Perl interpreter.
- Centreon Perl Connector forks itself.
- The precompiled script gets executed

This way Perl scripts are only parsed once during the lifetime of the monitoring engine. This heavily relates to [prepared statements](#) in SQL.