

# Manual for OpenSCADA building from sources

## Table of context

<a href="#">Manual for OpenSCADA building from sources</a> .....	1
<a href="#">Introduction</a> .....	1
<a href="#">OpenSCADA configuration for building</a> .....	2
<a href="#">System OpenSCADA direct building</a> .....	3
<a href="#">Building packages of the OpenSCADA system for the Linux distribution</a> .....	5
<a href="#">Redhat-based distributions - RPM package</a> .....	5
<a href="#">Debian-based distributions - Deb package</a> .....	6
<a href="#">Gentoo - metadistribution</a> .....	7

## Introduction

This manual made for help on OpenSCADA building from sources. In process of accumulation of experience of building for different OS Linux distributions and others platforms generally, this document will be complement for tips and details of building for different environments.

The document also contain information about preset configuration of built system of OpenSCADA, for complete start with using of present, included to sources, demo DB (model "AGLKS").

 For build OpenSCADA from sources you need experience in opensource program build, and also generic knowledges in OS Linux and used Linux distributive, that allow correct resolve dependences and solve possible build problems.

## OpenSCADA configuration for building

By OpenSCADA project, as a build system, using a set of utilities [AutoTools](#). For any project building under control "AutoTools" precedes the stage configuration, and sometimes the formation of the build system under environment of the OS. The stage configuration, as well as directly build, made always and not depend from is a direct build or build the package.

To form the build system must be installed utility "AutoTools": "autoconf", "automake" and "libtool". Common forming you can made by command "**\$ autoreconf -if**". Earlier, formed build the system may sometimes be necessary to update, for example, in the case of significant differences in operating environments where the build system was formed and the current. Reconfiguring the build system is carried by previously specified formation command.

The build system configuration carried by command "**\$ configure**". By specifying the arguments to this command, you can enable/disable a function or module OpenSCADA for build. Fro get all allowed configure parameters and OpenSCADA modules you can specify argument **--help**.

Enabling and disabling modules OpenSCADA made by passing arguments **--enable-*{ModName}*** and **--disable-*{ModName}***. To modules select you can specify a policy of "Enable all modules" **--enable-AllModuls** or "Disable all modules" **--disable-AllModuls**, and then enable/disable individual modules, depending on it would be easier. Most modules OpenSCADA can be linked (embedded) directly into the OpenSCADA, core library and not build separate library, for which you need to specify the argument **--enable-*{ModName}*=incl**.

As a result, the configuration will be formed summary table settings, which can be used to verify the desired configuration.

## System OpenSCADA direct building

Build system must be executed from normal system user name. Thus described variants of installing and execution how global and local, to user directory. Get what user login is - "user".

Before building enter as common user and change user login to your, get sources (from ftp/http-server or SVN-repository), select install method (global or local) and follow instruction:

Steps of preparation of the source tree to build:

1. Make directory for project building:  

```
$ mkdir ~/projects; cd ~/projects
```
2. Download sources package. It may be made from two ways:
  - download tar archive from ftp-server <ftp://ftp.oscada.org/OpenSCADA/> and unpacked it, for example:  

```
$ wget ftp://ftp.oscada.org/OpenSCADA/0.8.0/openscada-0.8.0.tar.lzma  
$ wget ftp://ftp.oscada.org/OpenSCADA/0.8.0/openscada-res-0.8.0.tar.lzma  
$ tar --lzma -xvf openscada-0.8.0.tar.lzma  
$ cd openscada-0.8.0  
$ tar --lzma -xvf ../openscada-res-0.8.0.tar.lzma
```
  - get project's source tree from SVN-repository:  

```
$ svn co svn://oscada.org/trunk/OpenSCADA  
$ cd OpenSCADA
```
3. Resolve OpenSCADA dependents for build, on according [demands](#).
4. Generate of build system, when download from the SVN-repository:  

```
$ autoreconf -ivf
```

Steps of global building:

1. Configuring building system of OpenSCADA:  

```
$ ./configure
```
2. Make OpenSCADA project:  

```
$ make
```
3. Install OpenSCADA:  

```
$ su; make install
```
4. Copy start script and configuration file of demonstration:  

```
$ cp data/ModelsDB/AGLKS/openscada_demo /usr/bin  
$ cp data/ModelsDB/AGLKS/oscada_AGLKS.xml /etc
```
5. Make data directory and fill it:  

```
$ mkdir /var/spool/openscada  
$ mkdir /var/spool/openscada/{DATA,icons,LibsDB,AGLKS}  
$ cp data/LibsDB/*.db /var/spool/openscada/LibsDB  
$ cp data/ModelsDB/AGLKS/*.db /var/spool/openscada/AGLKS  
$ cp data/icons/* /var/spool/openscada/icons  
$ install -m 777 -d /var/spool/openscada/ARCHIVES/{MESS,VAL}
```
6. System ready to start:  

```
$ exit; openscada_demo
```

Steps of local building:

1. Configuring building system of OpenSCADA:  

```
$ mkdir ~/OScadaW; ./configure --prefix=/home/user/OScadaW
```
2. Make OpenSCADA project:  

```
$ make
```
3. Install OpenSCADA:  

```
$ make install
```
4. Copy start script and configuration file of demonstration:  

```
$ cp data/ModelsDB/AGLKS/openscada_demo_local ~/OScadaW/bin/openscada_demo  
$ mkdir ~/OScadaW/etc  
$ cp data/ModelsDB/AGLKS/oscada_demo_local.xml  
~/OScadaW/etc/oscada_demo.xml
```
5. Make data directory and fill it:  

```
$ mkdir ~/OScadaW/share/openscada
```

```
$ mkdir ~/OScadaW/share/openscada/{DATA,icons,LibsDB,AGLKS}  
$ cp data/LibsDB/*.db ~/OScadaW/share/openscada/LibsDB  
$ cp data/ModelsDB/AGLKS/*.db ~/OScadaW/share/openscada/AGLKS  
$ cp data/icons/* ~/OScadaW/share/openscada/icons  
$ install -m 777 -d ~/OScadaW/share/openscada/ARCHIVES/{MESS,VAL}
```

**6. System ready to start:**

```
$ cd ~/OScadaW/bin; ./openscada_demo
```

# Building packages of the OpenSCADA system for the Linux distribution

## Redhat-based distributions - RPM package

To build the RPM-package you'll need OpenSCADA sources, resources tarballs and RPM package specification file (opencada.spec, opencada\_mod.spec) or SRPM-package of all these source materials (opencada-0.8.0-1.src.rpm), which can be downloaded from [ftp](#) or [http-server](#) of the project.

Specification file of the RPM-packages and SRPM-package may be different for distributions of Linux, in view of their features, so you should try to load them in accordance with the distribution. If the specific SRPM package is missing you can try to build from the unified specification file in the main SRPM package.

RPM-package building is made in the directory with a special structure, which is normally created when you install the appropriate distribution package (for example: rpm-build). This directory, depending on the distribution, is located in the user's home directory or in the directory `/usr/src`. The known facts of location the build directory of RPM-packages for different distributions:

- `/usr/src/packages`: [Open-SUSE](#)
- `~/RPM`: [ALTLinux](#)
- `~/rpmbuild`: [Mandriva](#), [Fedora and branches](#)

If, after the installation of a specialized package, the directory of RPM packages creation is missed, you need to create it with the following command:

```
$ install -m 755 -d Distribution directory/SPECS,SOURCES}
```

Build steps in the RPM specification file:

1. Placement of OpenSCADA archives and the specification file in the RPM build directory (the source files pre-placed in the current directory):

```
$ cp opencada-0.8.0.tar.lzma opencada-res-0.8.0.tar.lzma Distribution directory/SOURCES
```

```
$ cp opencada.spec Distribution directory/SPECS
```
2. Resolving of the OpenSCADA project's dependencies for building in accordance with [the requirements](#). Arguments for command "configure" you can specify into file "opencada.spec".
3. Building of packages
  - build from the beginning:

```
$ rpmbuild -bb Distribution directory/SPECS/opencada.spec
```
  - only the installation and packaging from the etersoft rpm-utilities package (useful when configuring opencada.spec file):

```
$ rpmbb -i Distribution directory/SPECS/opencada.spec
```
4. Packages are ready and available in the directory: **Distribution directory/RPMS** for the appropriate architecture.

Build steps from the SRPMS package:

1. Resolving of the OpenSCADA project's dependencies for building in accordance with [the requirements](#).
2. Building of the packages (the source files pre-placed in the current directory):

```
$ rpmbuild --rebuild opencada-0.8.0-1.src.rpm
```
3. Packages are ready and available in the directory **Distribution directory/RPMS** for the appropriate architecture.

## Debian-based distributions - Deb package

To build the RPM-package you'll need OpenSCADA sources, resources tarballs, which can be downloaded from [ftp](#) or [http](#)-server of the project.

Deb-package building is made in the "debian" directory with the building scripts, which is contained in the archives of the OpenSCADA sources.

Building steps:

1. Creation the directory to build the project:  

```
$ mkdir ~/build; cd ~/build
```
2. Download the source package. Download the sources and resources tarball from ftp-server of project <ftp://ftp.oscada.org/OpenSCADA/> and unpacking, like this:  

```
$ wget ftp://ftp.oscada.org/OpenSCADA/0.8.0/openscada-0.8.0.tar.lzma
$ wget ftp://ftp.oscada.org/OpenSCADA/0.8.0/openscada-res-0.8.0.tar.lzma
$ tar --lzma -xvf openscada-0.8.0.tar.lzma
$ cd openscada-0.8.0
$ tar --lzma -xvf ../openscada-res-0.8.0.tar.lzma
```
3. Resolving of the OpenSCADA project's dependencies for building in accordance with [the requirements](#). Arguments for command "configure" you can specify into file `openscada-0.8.0/debian/rules`.
4. The transition to the OpenSCADA directory and start the building:  

```
$ cd openscada-0.8.0
```

  - build from the beginning:  

```
$ dpkg-buildpackage -rfakeroot
```
  - skip the building and repack it is convenient for debugging **openscada-0.8.0/debian/rules**:  

```
$ fakeroot debian/rules binary
$ fakeroot debian/rules build
```
5. Packages are ready and available in the level above directory.

## Gentoo - metadistribution

OpenSCADA is not included to official portage Gentoo tree, so the building is made from the overlay. Details about the work with the overlay can be found in the Gentoo documentation, for example: [http://en.gentoo-wiki.com/wiki/Portage\\_Overlay](http://en.gentoo-wiki.com/wiki/Portage_Overlay). Before the building of the OpenSCADA package itself, portage will satisfy dependencies, ie build the missing packages (if necessary).

1. Creation of the portage overlay reduces to the describing the path to the overlay tree, ie in **/etc/make.conf** it is necessary to add the line:

```
PORTDIR_OVERLAY="/usr/local/portage"
```

2. Create the directory **/usr/local/portage/dev-util/openscada/**, then download the ebuild archive file from the official site and unpack it in the created folder.

```
$ cd /usr/local/portage/dev-util/openscada
$ wget ftp://ftp.openscada.org/OpenSCADA/0.8.0/Gentoo/openscada-0.8.0.ebuild.tar.gz
$ tar -xzf openscada-0.8.0.ebuild.tar.gz
```

3. You must specify the portage, that OpenSCADA is allowed to build on your platform. To do this, put the following line to the file **/etc/portage/package.keywords**:

```
dev-util/openscada
```

4. Check:

```
$ emerge -vp openscada
These are the packages that would be merged, in order:
Calculating dependencies... done!
[ebuild R ~] dev-util/openscada-0.8.0 USE="mysql ssl -bfm -blockcalc
-daqqate -dbarch -dbase -dcon -demo -diamondboards -doc (-firebird) -
flibcomplex1 -flibmath -flibsys -fsarch -http -icp_das -javalikecalc -
logiclev -modbus -portaudio -qtcfg -qtstarter -selfsystem -serial -simens
-snmpp -sockets -sqlite -system -systemtest -vcaengine -vision -webcfg -
webcfgd -webvision" 79,647 kB [1]
```

```
Total: 1 package (1 reinstall), Size of downloads: 79,647 kB
```

```
Portage tree and overlays:
```

```
[0] /usr/portage
[1] /usr/local/portage
```

5. If you don't have connection to the Internet, you have to download the sources and resources tarball from ftp-server of the project, and place them in advance in the sources directory of the portage (**/usr/portage/distfiles**). If you have an Internet connection, then everything that is necessary for building will be taken from the project's server automatically.
6. It is necessary to put the required USE-flags, with which the OpenSCADA will be built. This can be done in the file **/etc/portage/package.use**, for example:

```
dev-util/openscada dcon demo http javalikecalc mysql qtcfg qtstarter ssl
vision webcfg webcfgd webvision blockcalc dbarch dbase doc flibcomplex1
flibmath flibsys fsarch logiclev modbus serial sockets sqlite vcaengine
```

7. Directly building:

```
$ emerge -q openscada
```

8. Database files and resources of OpenSCADA are ready and placed in the directory **/var/spool/openscada/**, executable files - **/usr/bin/**, documentation - **/usr/share/doc/openscada-0.8.0/**.