## NAME

How to build SRecord

## SPACE REQUIREMENTS

You will need about 3MB to unpack and build the *SRecord* package. Your milage may vary.

## BEFORE YOU START

There are a few pieces of software you may want to fetch and install before you proceed with your installation of SRecord.

Libgcrypt Library

You will need the GNU Crypt library. If you are using a package based system, you will need the libgcrypt-devel package, or one named something very similar.
http://directory.fsf.org/project/libgcrypt/

GNU Libtool

You will need the GNU Libtool software, used to build shared libraries on a variety of systems.
http://www.gnu.org/software/libtool/

CMake You will need CMake to build, test install SRecord from source. depend on a build tool. CMake version 3.22 or later is required
http://cmake.org

GNU Groff

The documentation for the *SRecord* package was prepared using the GNU Groff package (version 1.14 or later). This distribution includes full documentation, which may be processed into PostScript or DVI files at install time – if GNU Groff has been installed.

GCC You may also want to consider fetching and installing the GNU C Compiler if you have not done so already. This is not essential. SRecord was developed using the GNU C++ compiler, and the GNU C++ libraries.

The GNU FTP archives may be found at `ftp.gnu.org`, and are mirrored around the world.

## BUILD ENVIRONMENT SETUP - LINUX

On systems using .deb packages such as Ubuntu or Debian, the following will install the necessary dependencies. Something similar will setup other distributions.

```
% sudo apt update
% sudo apt install build-essential g++ doxygen psutils libgcrypt20-dev
ghostscript groff cmake rpm
...lots of output...
%
```

## BUILD ENVIRONMENT SETUP - WINDOWS

Windows builds are based on the MSYS2 environment. While based on Cygwin, MSYS2's is geared toward building native applications. This will allow you to compile SRecord, build its documentation and run its test on Windows 7 and later.

Download and run the installer as described on the MSYS2 homepage `http://www.msys2.org`. Using the default installation locations is highly recommended.

Run the MINGW64 environment (giving a BASH prompt) and accept any updates. The first run will update the package database itself. Running it a second time will install updates flagged in the updated database.:

```
% pacman -Syu
...lots of output...
% pacman -Syu
...lots of output...
% pacman -S mingw-w64-x86_64-gcc groff \
     mingw-w64-x86_64-libgcrypt mingw-w64-x86_64-cmake \
     mingw-w64-x86_64-graphviz mingw-w64-x86_64-ninja \
```

```
        mingw-w64-x86_64-doxygen mingw-w64-i686-ghostscript \
...lots of output...
%
```

## SITE CONFIGURATION

The **SRecord** package is configured using the *cmake* program.

*cmake* attempts to guess correct values for various system-dependent variables used during compilation, and creates the *Makefile* and *lib/config.h* files. It also creates a shell script *config.status* that you can run in the future to recreate the current configuration.

Normally, from the top-leve directory of the source package, you just create and run cmake from a build directory

```
% mkdir build
% cd build
% cmake ..
...lots of output...
%
```

By default, on Linux systems, *cmake* will arrange to install the **SRecord** package's files in */usr/local/bin*, and */usr/local/man*. There are variables which allow you to control the placement of these files as well as many other options.

Here are the *cmake* variables that you might want to override with environment variables when running *configure*.

Variable: CXX
: C++ compiler programm. The default is determined by cmake according to the opeerating system and environment variables. */usr/local*.

Variable: CMAKE_INSTALL_PREFIX
: The common root of the installation tree for the files The default is */usr/local*.

If you need to do unusual things to compile the package, the author encourages you to figure out how *cmake* could check whether to do them, and mail diffs or instructions to the author so that they can be included in the next release.

## BUILDING SRECORD

All you should need to is ensure you are in the build directory and use the

```
% cmake --build .
...lots of output...
%
```

command and wait. When this finishes you should see directories called *srec_cat*, *srec_cmp* and *srec_info* containing executables by the same name.

**srec_cat**
: The *srec_cat* program is used to manipulate and convert EPROM load files. For more information, see *srec_cat*(1).

**srec_cmp**
: The *srec_cmp* program is used to compare EPROM load files. For more information, see *srec_cmp*(1).

**srec_info**
: The *srec_info* program is used to print information about EPROM load files. For more information, see *srec_info*(1).

If you have GNU Groff installed, the build will also create a *doc/REFERENCE.pdf* file. This contains the README file, this BUILDING file, and all of the man pages.

The build directory can be deleted at any time.

## TESTING SRECORD

The *SRecord* package comes with a test suite. To run this test suite, use the command

**%** `ctest`
*...lots of output...*
**%**

The tests take a few seconds each, with a few very fast, and a couple very slow, but it varies greatly depending on your CPU.

If all went well, the message

```
100% tests passed
```

should appear at the end.

## INSTALLING SRECORD

As explained in the *SITE CONFIGURATION* section, above, on Linux systems the *SRecord* package is installed under the */usr/local* tree by default. Use the `--prefix=`*PATH* option to *configure* if you want some other path. More specific installation locations are assignable, use the `--help` option to *configure* for details.

All that is required to install the *SRecord* package is to use the

**%** `cmake --install`
*...lots of output...*
**%**

command.

## PACKAGING SRECORD

Installation packages can be created. On Linux platforms .deb, .rpm and .tar.gz are supported and tested. Packages are created with the following command:

**%** `cpack -G DEB`
*...lots of output...*
**%**

or to build multiple packages:

**%** `cpack -G "DEB;RPM;TGZ"`
*...lots of output...*
**%**

On Windows, ZIP archive is supported and is created similarly:

**%** `cpack -G ZIP`
*...lots of output...*
**%**

## GETTING HELP

If you need assistance with the *SRecord* package, please post to the srecord-users mailing list

```
srecord-users@lists.sourceforge.net
```

For information obout the srecord-users mailing list. http://srecord.sourceforge.net/mailing-list.html

When reporting problems, please include the version number given by the

**%** `srec_cat -version`
**srecord version** *1.65.0*
*...warranty disclaimer...*
**%**

command. Please do not send this example; run the program for the exact version number.

## COPYRIGHT

*srecord* version 1.65
Copyright © 1998... Peter Miller
Copyright © 1998... Scott Finneran

The *SRecord* package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

It should be in the *LICENSE* file included with this distribution.

## AUTHOR

Scott Finneran E-Mail: scottfinneran@yahoo.com.au
Peter Miller E-Mail: pmiller@opensource.org.au