# The GNU libextractor Reference Manual

Version 1.0.0
8 September 2012

**Christian Grothoff** (christian@grothoff.org)

This manual is for GNU libextractor (version 1.0.0, 8 September 2012), a library for meta-data extraction.

# Short Contents

# Table of Contents

# 1 Introduction

GNU libextractor is GNU's library for extracting meta data from files. Meta data includes format information (such as mime type, image dimensions, color depth, recording frequency), content descriptions (such as document title or document description) and copyright information (such as license, author and contributors). Meta data extraction is an inherently uncertain business — a parse error can be a corrupt file, an incompatibility in the file format version, an entirely different file format or a bug in the parser. As a result of this uncertainty, GNU libextractor deliberately avoids to ever report any errors. Unexpected file contents simply result in less or possibly no meta data being extracted.

GNU libextractor uses plugins to handle various file formats. Technically a plugin can support multiple file formats; however, most plugins only support one particular format. By default, GNU libextractor will use all plugins that are available and found in the plugin installation directory. Applications can request the use of only specific plugins or the exclusion of certain plugins.

GNU libextractor is distributed with the `extract` command[1] which is a command-line tool for extracting meta data. `extract` is given a list of filenames and prints the resulting meta data to the console. The `extract` source code also serves as an advanced example for how to use GNU libextractor.

This manual focuses on providing documentation for writing software with GNU libextractor. The only relevant parts for end-users are the chapter on compiling and installing GNU libextractor (See Chapter 2 [Preparation], page 2.). Also, the chapter on existing plugins maybe of interest (See Chapter 7 [Existing Plugins], page 15.). Additional documentation for end-users can be find in the man page on `extract` (using `man extract`).

GNU libextractor is licensed under the GNU General Public License, specifically, since version 0.7, GNU libextractor is licensed under GPLv3 *or any later version*.

[1] Some distributions ship `extract` in a seperate package.

# 2 Preparation

This chapter first describes the general build instructions that should apply to all systems. Specific instructions for known problems for particular platforms are then described in individual sections afterwards.

Compiling GNU libextractor follows the standard GNU autotools build process using `configure` and `make`. For details on the GNU autotools build process, read the ‘`INSTALL`’ file and query `./configure --help` for additional options.

GNU libextractor has various dependencies, most of which are optional. Instead of specifying the names of the software packages, we will give the list in terms of the names of the respective Debian (wheezy) packages that should be installed.

You absolutely need:

- libtool
- gcc
- make
- g++
- libltdl7-dev

Recommended dependencies are:

- zlib1g-dev
- libbz2-dev
- libgif-dev
- libvorbis-dev
- libflac-dev
- libmpeg2-4-dev
- librpm-dev
- libgtk2.0-dev or libgtk3.0-dev
- libgsf-1-dev
- libqt4-dev
- libpoppler-dev
- libexiv2-dev
- libavformat-dev
- libswscale-dev
- libgstreamer1.0-dev

For Subversion access and compilation one also needs:

- subversion
- autoconf
- automake

Please notify us if we missed some dependencies (note that the list is supposed to only list direct dependencies, not transitive dependencies).

Once you have compiled and installed GNU libextractor, you should have a file '`extractor.h`' installed in your '`include/`' directory. This file should be the starting point for your C and C++ development with GNU libextractor. The build process also installs the '`extract`' binary and man pages for '`extract`' and GNU libextractor. The '`extract`' man page documents the '`extract`' tool. The GNU libextractor man page gives a brief summary of the C API for GNU libextractor.

When you install GNU libextractor, various plugins will be installed in the '`lib/libextractor/`' directory. The main library will be installed as '`lib/libextractor.so`'. Note that GNU libextractor will attempt to find the plugins relative to the path of the main library. Consequently, a package manager can move the library and its plugins to a different location later — as long as the relative path between the main library and the plugins is preserved. As a method of last resort, the user can specify an environment variable `LIBEXTRACTOR_PREFIX`. If GNU libextractor cannot locate a plugin, it will look in `LIBEXTRACTOR_PREFIX/lib/libextractor/`.

## 2.1 Installation on GNU/Linux

Should work using the standard instructions without problems.

## 2.2 Installation on FreeBSD

Should work using the standard instructions without problems.

## 2.3 Installation on OpenBSD

OpenBSD 3.8 also doesn't have CODESET in '`langinfo.h`'. CODESET is used in GNU libextractor in about three places. This causes problems during compilation.

## 2.4 Installation on NetBSD

No reports so far.

## 2.5 Installation using MinGW

Linking -lstdc++ with the provided libtool fails on Cygwin, this is a problem with libtool, there is unfortunately no flag to tell libtool how to do its job on Cygwin and it seems that it cannot be the default to set the library check to 'pass_all'. Patching libtool may help.

Note: this is a rather dated report and may no longer apply.

## 2.6 Installation on OS X

libextractor has two installation methods on Mac OS X: it can be installed as a Mac OS X framework or with the standard `./configure; make; make install` shell commands. The framework package is self-contained, but currently omits some of the extractor plugins that can be compiled in if libextractor is installed with `./configure; make; make install` (provided that the required dependencies exist.)

### 2.6.1 Installing and uninstalling the framework

The binary framework is distributed as a disk image ('`Extractor-x.x.xx.dmg`'). Installation is done by opening the disk image and clicking '`Extractor.pkg`' inside it. The Mac

OS X installer application will then run. The framework is installed to the root volume's '`/Library/Frameworks`' folder and installing will require admin privileges.

The framework can be uninstalled by dragging '`/Library/Frameworks/Extractor.framework`' to the '`Trash`'.

### 2.6.2 Using the framework

In the framework, the `extract` command line tool can be found at '`/Library/Frameworks/Extractor.framework/Versions/Current/bin/extract`'

The framework can be used in software projects as a framework or as a dynamic library.

When using the framework as a dynamic library in projects using autotools, one would most likely want to add "-I/Library/Frameworks/Extractor.framework/Versions/Current/include" to CPPFLAGS and "-L/Library/Frameworks/Extractor.framework/Versions/Current/lib" to LDFLAGS.

### 2.6.3 Example for using the framework

```
// hello.c
#include <Extractor/extractor.h>

int
main (int argc, char **argv)
{
  struct EXTRACTOR_PluginList *el;
  el = EXTRACTOR_plugin_load_defaults (EXTRACTOR_OPTION_DEFAULT_POLICY);
  // ...
  EXTRACTOR_plugin_remove_all (el);
  return 0;
}
```

You can then compile the example using

```
$ gcc -o hello hello.c -framework Extractor
```

### 2.6.4 Example for using the dynamic library

```
// hello.c
#include <extractor.h>
int main()
{
  struct EXTRACTOR_PluginList *el;
  el = EXTRACTOR_plugin_load_defaults (EXTRACTOR_OPTION_DEFAULT_POLICY);
  // ...
  EXTRACTOR_plugin_remove_all (el);
  return 0;
}
```

You can then compile the example using

```
$ gcc -I/Library/Frameworks/Extractor.framework/Versions/Current/include \
  -o hello hello.c \
```

```
-L/Library/Frameworks/Extractor.framework/Versions/Current/lib \
-lextractor
```

Notice the difference in the `#include` line.

## 2.7 Note to package maintainers

The suggested way to package GNU libextractor is to split it into roughly the following binary packages:

- libextractor (main library only, only hard dependency for other packages depending on GNU libextractor)
- extract (command-line tool and man page extract.1)
- libextractor-dev (extractor.h header and man page libextractor.3)
- libextractor-doc (this manual)
- libextractor-plugins (plugins without external dependencies; recommended but not required by extract and libextractor package)
- libextractor-plugin-XXX (plugin with dependency on libXXX, for example for XXX=mpeg this would be '`libextractor_mpeg.so`')
- libextractor-plugins-all (meta package that requires all plugins except experimental plugins)

This would enable minimal installations (i.e. for embedded systems) to not include any plugins, as well as moderate-size installations (that do not trigger GTK and X11) for systems that have limited resources. Right now, the MP4 plugin is experimental and does nothing and should thus never be included at all. The gstreamer plugin is experimental but largely works with the correct version of gstreamer and can thus be packaged (especially if the dependency is available on the target system) but should probably not be part of libextractor-plugins-all.

# 3 Generalities

## 3.1 Introduction to the "extract" command

The `extract` command takes a list of file names as arguments, extracts meta data from each of those files and prints the result to the console. By default, `extract` will use all available plugins and print all (non-binary) meta data that is found.

The set of plugins used by `extract` can be controlled using the "-l" and "-n" options. Use "-n" to not load all of the default plugins. Use "-l NAME" to specifically load a certain plugin. For example, specify "-n -l mime" to only use the MIME plugin.

Using the "-p" option the output of `extract` can be limited to only certain keyword types. Similarly, using the "-x" option, certain keyword types can be excluded. A list of all known keyword types can be obtained using the "-L" option.

The output format of `extract` can be influenced with the "-V" (more verbose, lists filenames), "-g" (grep-friendly, all meta data on a single line per file) and "-b" (bibTeX style) options.

## 3.2 Common usage examples for "extract"

```
$ extract test/test.jpg
comment - (C) 2001 by Christian Grothoff, using gimp 1.2 1
mimetype - image/jpeg

$ extract -V -x comment test/test.jpg
Keywords for file test/test.jpg:
mimetype - image/jpeg

$ extract -p comment test/test.jpg
comment - (C) 2001 by Christian Grothoff, using gimp 1.2 1

$ extract -nV -l png.so -p comment test/test.jpg test/test.png
Keywords for file test/test.jpg:
Keywords for file test/test.png:
comment - Testing keyword extraction
```

## 3.3 Introduction to the libextractor library

Each public symbol exported by GNU libextractor has the prefix `EXTRACTOR_`. All-caps names are used for constants. For the impatient, the minimal C code for using GNU libextractor (on the executing binary itself) looks like this:

```
#include <extractor.h>

int
main (int argc, char ** argv)
{
  struct EXTRACTOR_PluginList *plugins
```

```
    = EXTRACTOR_plugin_add_defaults (EXTRACTOR_OPTION_DEFAULT_POLICY);
  EXTRACTOR_extract (plugins, argv[1],
                     NULL, 0,
                     &EXTRACTOR_meta_data_print, stdout);
  EXTRACTOR_plugin_remove_all (plugins);
  return 0;
}
```

The minimal API illustrated by this example is actually sufficient for many applications. The full external C API of GNU libextractor is described in chapter See Chapter 4 [Extracting meta data], page 8. Bindings for other languages are described in chapter See Chapter 5 [Language bindings], page 12. The API for writing new plugins is described in chapter See Chapter 8 [Writing new Plugins], page 16.

# 4 Extracting meta data

In order to extract meta data with GNU libextractor you first need to load the respective plugins and then call the extraction API with the plugins and the data to process. This section documents how to load and unload plugins, the various types and formats in which meta data is returned to the application and finally the extraction API itself.

## 4.1 Plugin management

Using GNU libextractor from a multi-threaded parent process requires some care. The problem is that on most platforms GNU libextractor starts sub-processes for the actual extraction work. This is useful to isolate the parent process from potential bugs; however, it can cause problems if the parent process is multi-threaded. The issue is that at the time of the fork, another thread of the application may hold a lock (i.e. in gettext or libc). That lock would then never be released in the child process (as the other thread is not present in the child process). As a result, the child process would then deadlock on trying to acquire the lock and never terminate. This has actually been observed with a lock in GNU gettext that is triggered by the plugin startup code when it interacts with libltdl.

The problem can be solved by loading the plugins using the `EXTRACTOR_OPTION_IN_PROCESS` option, which will run GNU libextractor in-process and thus avoid the locking issue. In this case, all of the functions for loading and unloading plugins, including `EXTRACTOR_plugin_add_defaults` and `EXTRACTOR_plugin_remove_all`, are thread-safe and reentrant. However, using the same plugin list from multiple threads at the same time is not safe.

All plugin code is expected required to be reentrant and state-less, but due to the extensive use of 3rd party libraries this cannot be guaranteed.

`EXTRACTOR_PluginList` [C Struct]

A plugin list represents a set of GNU libextractor plugins. Most of the GNU libextractor API is concerned with either constructing a plugin list or using it to extract meta data. The internal representation of the plugin list is of no concern to users or plugin developers.

`void EXTRACTOR_plugin_remove_all` (*struct EXTRACTOR_PluginList *plugins*) [Function]

Unload all of the plugins in the given list.

`struct EXTRACTOR_PluginList * EXTRACTOR_plugin_remove` (*struct EXTRACTOR_PluginList *plugins, const char*name*) [Function]

Unloads a particular plugin. The given name should be the short name of the plugin, for example “mime” for the mime-type extractor or “mpeg” for the MPEG extractor.

`struct EXTRACTOR_PluginList * EXTRACTOR_plugin_add` (*struct EXTRACTOR_PluginList *plugins, const char* name,const char* options, enum EXTRACTOR_Options flags*) [Function]

Loads a particular plugin. The plugin is added to the existing list, which can be `NULL`. The second argument specifies the name of the plugin (i.e. “ogg”). The third argument can be NULL and specifies plugin-specific options. Finally, the last argument specifies if the plugin should be executed out-of-process (EXTRACTOR_OPTION_DEFAULT_POLICY) or not.

`struct EXTRACTOR_PluginList * EXTRACTOR_plugin_add_config` (*struct EXTRACTOR_PluginList *plugins, const char* config, enum EXTRACTOR_Options flags*) [Function]

Loads and unloads plugins based on a configuration string, modifying the existing list, which can be NULL. The string has the format "[-]NAME(OPTIONS){:[-]NAME(OPTIONS)}*". Prefixing the plugin name with a "-" means that the plugin should be unloaded.

`struct EXTRACTOR_PluginList * EXTRACTOR_plugin_add_defaults` (*enum EXTRACTOR_Options flags*) [Function]

Loads all of the plugins in the plugin directory. This function is what most GNU libextractor applications should use to setup the plugins.

## 4.2 Meta types

enum EXTRACTOR_MetaType is a C enum which defines a list of over 100 different types of meta data. The total number can differ between different GNU libextractor releases; the maximum value for the current release can be obtained using the EXTRACTOR_metatype_get_max function. All values in this enumeration are of the form EXTRACTOR_METATYPE_XXX.

`const char * EXTRACTOR_metatype_to_string` (*enum EXTRACTOR_MetaType type*) [Function]

The function EXTRACTOR_metatype_to_string can be used to obtain a short English string 's' describing the meta data type. The string can be translated into other languages using GNU gettext with the domain set to GNU libextractor (dgettext("libextractor", s)).

`const char * EXTRACTOR_metatype_to_description` (*enum EXTRACTOR_MetaType type*) [Function]

The function EXTRACTOR_metatype_to_description can be used to obtain a longer English string 's' describing the meta data type. The description may be empty if the short description returned by EXTRACTOR_metatype_to_string is already comprehensive. The string can be translated into other languages using GNU gettext with the domain set to GNU libextractor (dgettext("libextractor", s)).

## 4.3 Meta formats

enum EXTRACTOR_MetaFormat is a C enum which defines on a high level how the extracted meta data is represented. Currently, the library uses three formats: UTF-8 strings, C strings and binary data. A fourth value, EXTRACTOR_METAFORMAT_UNKNOWN is defined but not used. UTF-8 strings are 0-terminated strings that have been converted to UTF-8. The format code is EXTRACTOR_METAFORMAT_UTF8. Ideally, most text meta data will be of this format. Some file formats fail to specify the encoding used for the text. In this case, the text cannot be converted to UTF-8. However, the meta data is still known to be 0-terminated and presumably human-readable. In this case, the format code used is `EXTRACTOR_METAFORMAT_C_STRING`; however, this should not be understood to mean that the encoding is the same as that used by the C compiler. Finally, for binary data (mostly images), the format `EXTRACTOR_METAFORMAT_BINARY` is used.

Naturally this is not a precise description of the meta format. Plugins can provide a more precise description (if known) by providing the respective mime type of the meta data. For example, binary image meta data could be also tagged as "image/png" and normal text would typically be tagged as "text/plain".

## 4.4 Extracting

`int (*EXTRACTOR_MetaDataProcessor)(void` **cls, const char *plugin_name, enum EXTRACTOR_MetaType type, enum EXTRACTOR_MetaFormat format, const char *data_mime_type, const char *data, size_t data_len*) [Function Pointer]

Type of a function that libextractor calls for each meta data item found.

*cls* closure (user-defined)

*plugin_name*
name of the plugin that produced this value; special values can be used (i.e. '<zlib>' for zlib being used in the main libextractor library and yielding meta data);

*type* libextractor-type describing the meta data;

*format basic*
format information about data

*data_mime_type*
mime-type of data (not of the original file); can be `NULL` (if mime-type is not known);

*data* actual meta-data found

*data_len* number of bytes in data

Return 0 to continue extracting, 1 to abort.

`void EXTRACTOR_extract` (*struct EXTRACTOR_PluginList *plugins, const char *filename, const void *data, size_t size, EXTRACTOR_MetaDataProcessor proc, void *proc_cls*) [Function]

This is the main function for extracting keywords with GNU libextractor. The first argument is a plugin list which specifies the set of plugins that should be used for extracting meta data. The '`filename`' argument is optional and can be used to specify the name of a file to process. If '`filename`' is `NULL`, then the '`data`' argument must point to the in-memory data to extract meta data from. If '`filename`' is non-`NULL`, '`data`' can be `NULL`. If '`data`' is non-null, then '`size`' is the size of '`data`' in bytes. Otherwise '`size`' should be zero. For each meta data item found, GNU libextractor will call the '`proc`' function, passing '`proc_cls`' as the first argument to '`proc`'. The other arguments to '`proc`' depend on the specific meta data found.

Meta data extraction should never really fail — at worst, GNU libextractor should not call '`proc`' with any meta data. By design, GNU libextractor should never crash or leak memory, even given corrupt files as input. Note however, that running GNU libextractor on a corrupt file system (or incorrectly `mmap`ed files) can result in the operating system sending a SIGBUS (bus error) to the process. While GNU libextractor runs plugins out-of-process, it first maps the file into memory and then attempts to decompress it. During decompression it is possible to encounter a SIGBUS. GNU libextractor will *not* attempt to catch this signal and your application is likely to crash. Note again that this should only happen if the file *system* is corrupt (not if individual files are corrupt). If this is not acceptable, you might want to consider running GNU libextractor itself also out-of-process (as done, for example, by doodle (`http://grothoff.org/christian/doodle/`)).

# 5 Language bindings

GNU libextractor works immediately with C and C++ code. Bindings for Java, Mono, Ruby, Perl, PHP and Python are available for download from the main GNU libextractor website. Documentation for these bindings (if available) is part of the downloads for the respective binding. In all cases, a full installation of the C library is required before the binding can be installed.

## 5.1 Java

Compiling the GNU libextractor Java binding follows the usual process of running `configure` and `make`. The result will be a shared C library '`libextractor_java.so`' with the native code and a JAR file (installed to '`$PREFIX/share/java/libextractor.java`').

A minimal example for using GNU libextractor's Java binding would look like this:

```
import org.gnu.libextractor.*;
import java.util.ArrayList;

public static void main(String[] args) {
  Extractor ex = Extractor.getDefault();
  for (int i=0;i<args.length;i++) {
    ArrayList keywords = ex.extract(args[i]);
    System.out.println("Keywords for " + args[i] + ":");
    for (int j=0;j<keywords.size();j++)
      System.out.println(keywords.get(j));
  }
}
```

The GNU libextractor library and the '`libextractor_java.so`' JNI binding have to be in the library search path for this to work. Furthermore, the '`libextractor.jar`' file should be on the classpath.

Note that the API does not use Java 5 style generics in order to work with older versions of Java.

## 5.2 Mono

his binding is undocumented at this point.

## 5.3 Perl

This binding is undocumented at this point.

## 5.4 Python

This binding is undocumented at this point.

## 5.5 PHP

This binding is undocumented at this point.

## 5.6 Ruby

This binding is undocumented at this point.

# 6 Utility functions

This chapter describes various utility functions for GNU libextractor usage. All of the functions are reentrant.

## 6.1 Utility Constants

The constant `EXTRACTOR_VERSION` is a hexadecimal representation of the version number of the installed libextractor header. The hexadecimal format is 0xAABBCCDD where AA is the major version (so far always 0), BB is the minor version, CC is the revision and DD the patch number. For example, for version 0.5.18, we would have AA=0, BB=5, CC=18 and DD=0. Minor releases such as 0.5.18a or significant changes in unreleased versions would be marked with DD=1 or higher.

## 6.2 Meta data printing

The `EXTRACTOR_meta_data_print` is a simple function which prints the meta data found with libextractor to a file. The function is mostly useful for debugging and as an example for how to manipulate the keyword list and can be passed as the '`proc`' argument to `EXTRACTOR_extract`. The file to print to should be passed as '`proc_cls`' (which must be of type `FILE *`), for example `stdout`.

# 7 Existing Plugins

- ARCHIVE (using libarchive)
- DVI
- EXIV2 (using libexiv2, 0.23 or later preferred)
- FLAC (using libFLAC)
- GIF (using libgif)
- GSTREAMER (using libgstreamer v1.0 or later)
- HTML (using libtidy)
- IT
- JPEG (using libjpeg v8 or later)
- MAN
- MIDI (using libsmf)
- MIME (using libmagic)
- MPEG (using libmpeg2)
- NSF
- NSFE
- ODF
- OLE2 (with libgsf)
- OGG (with libogg)
- PNG
- PS
- RIFF
- RPM (using librpm)
- S3M
- SID
- ThumbnailFFMPEG (using libavformat and related libav-libraries, including libswscale)
- ThumbnailGtk (using libgtk)
- TIFF (with libtiff, tested with v4)
- WAV
- XM
- ZIP

'`gzip`' and '`bzip2`' compressed versions of these formats are also supported (as well as meta data embedded by '`gzip`' itself) if zlib or libbz2 are available.

# 8 Writing new Plugins

Writing a new plugin for libextractor usually requires writing of or interfacing with an actual parser for a specific format. How this is can be accomplished depends on the format and cannot be specified in general. However, care should be taken for the code to be reentrant and highly fault-tolerant, especially with respect to malformed inputs.

Plugins should start by verifying that the header of the data matches the specific format and immediately return if that is not the case. Even if the header matches the expected file format, plugins must not assume that the remainder of the file is well formed.

The plugin library must be called libextractor_XXX.so, where XXX denotes the file format of the plugin. The library must export a method `libextractor_XXX_extract_method`, with the following signature:

```
void
EXTRACTOR_XXX_extract_method (struct EXTRACTOR_ExtractContext *ec);
```

'`ec`' contains various information the plugin may need for its execution. Most importantly, it contains functions for reading ("read") and seeking ("seek") the input data and for returning extracted data ("proc"). The "config" member can contain additional configuration options. "proc" should be called on each meta data item found. If "proc" returns non-zero, processing should be aborted (if possible).

In order to test new plugins, the '`extract`' command can be run with the options "-ni" and "-l XXX" . This will run the plugin in-process (making it easier to debug) and without any of the other plugins.

## 8.1 Example for a minimal extract method

The following example shows how a plugin can return the mime type of a file.

```
void
EXTRACTOR_mymime_extract (struct EXTRACTOR_ExtractContext *ec)
{
  void *data;
  ssize_t data_size,

  if (-1 == (data_size = ec->read (ec->cls, &data, 4)))
    return; /* read error */
  if (data_size < 4)
    return; /* file too small */
  if (0 != memcmp (data, "\177ELF", 4))
    return; /* not ELF */
  if (0 != ec->proc (ec->cls,
                     "mymime",
                     EXTRACTOR_METATYPE_MIMETYPE,
                     EXTRACTOR_METAFORMAT_UTF8,
                     "text/plain",
                     "application/x-executable",
                     1 + strlen("application/x-executable")))
    return;
```

```
  /* more calls to 'proc' here as needed */
}
```

# 9 Internal utility functions

Some plugins link against the `libextractor_common` library which provides common abstractions needed by many plugins. This section documents this internal API for plugin developers. Note that the headers for this library are (intentionally) not installed: we do not consider this API stable and it should hence only be used by plugins that are build and shipped with GNU libextractor. Third-party plugins should not use it.

'`convert_numeric.h`' defines various conversion functions for numbers (in particular, byte-order conversion for floating point numbers).

'`unzip.h`' defines an API for accessing compressed files.

'`pack.h`' provides an interpreter for unpacking structs of integer numbers from streams and converting from big or little endian to host byte order at the same time.

'`convert.h`' provides a function for character set conversion described below.

`char * EXTRACTOR_common_convert_to_utf8` (*const char *input, size_t len, const char *charset*) [Function]

Various GNU libextractor plugins make use of the internal '`convert.h`' header which defines a function

`EXTRACTOR_common_convert_to_utf8` which can be used to easily convert text from any character set to UTF-8. This conversion is important since the linked list of keywords that is returned by GNU libextractor is expected to contain only UTF-8 strings. Naturally, proper conversion may not always be possible since some file formats fail to specify the character set. In that case, it is often better to not convert at all.

The arguments to `EXTRACTOR_common_convert_to_utf8` are the input string (which does *not* have to be zero-terminated), the length of the input string, and the character set (which *must* be zero-terminated). Which character sets are supported depends on the platform, a list can generally be obtained using the `iconv -l` command. The return value from `EXTRACTOR_common_convert_to_utf8` is a zero-terminated string in UTF-8 format. The responsibility to free the string is with the caller, so storing the string in the keyword list is acceptable.

# 10 Reporting bugs

GNU libextractor uses the Mantis bugtracking system (`https://gnunet.org/bugs/`). If possible, please report bugs there. You can also e-mail the GNU libextractor mailinglist at `libextractor@gnu.org`.

# Appendix A GNU Free Documentation License



0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with... Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

# Index