Libtasn1

Abstract Syntax Notation One (ASN.1) library for the GNU system
part of the GnuTLS project
for version 1.5, 26 August 2008

Fabio Fiorina
Simon Josefsson (bug-gnutls@gnu.org)

This manual is for Libtasn1 (version 1.5, 26 August 2008), which is a library for Abstract Syntax Notation One (ASN.1) and Distinguish Encoding Rules (DER) manipulation.

Table of Contents

1 Introduction

This document describes the Libtasn1 library developed for ASN.1 (Abstract Syntax Notation One) structures management and DER (Distinguished Encoding Rules) encoding functions.

The main features of this library are:

- On line ASN1 structure management that doesn't require any C code file generation.
- Off line ASN1 structure management with C code file generation containing an array.
- DER (Distinguish Encoding Rules) encoding.
- No limits for INTEGER and ENUMERATED values.
- It's Free Software. Anybody can use, modify, and redistribute the library under the terms of the GNU Lesser General Public License version 2.1 (see Section A.2 [GNU LGPL], page 25). The command line tools, self-tests and build infrastructure are licensed under the GNU General Public License version 3.0 (see Section A.3 [GNU GPL], page 33).
- It's thread-safe. No global variables are used and multiple library handles and session handles may be used in parallel.
- It's portable. It should work on all Unix like operating systems, including Windows. The library itself should be portable to any C89 system, not even POSIX is required.

2 ASN.1 structure handling

2.1 ASN.1 syntax

The parser is case sensitive. The comments begin with "– " and end at the end of lines. An example is in "pkix.asn" file. ASN.1 definitions must have this syntax:

```
definitions_name {<object definition>}

DEFINITIONS <EXPLICIT or IMPLICIT> TAGS ::=

BEGIN

<type and constants definitions>

END
```

The token "::=" must be separate from others elements, so this is a wrong declaration:

```
;; INCORRECT
Version ::=INTEGER
```

the correct form is:

```
Version ::= INTEGER
```

Here is the list of types that the parser can manage:

- INTEGER
- ENUMERATED
- BOOLEAN
- OBJECT IDENTIFIER
- NULL
- BIT STRING
- OCTET STRING
- UTCTime
- GeneralizedTime
- GeneralString
- SEQUENCE
- SEQUENCE OF
- SET
- SET OF
- CHOICE
- ANY
- ANY DEFINED BY

This version doesn't manage REAL type. It doesn't allow the "EXPORT" and "IMPORT" sections too.

The SIZE constraints are allowed, but no check is done on them.

2.2 Naming

Consider this definition:

```
Example { 1 2 3 4 }

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

Group ::= SEQUENCE {
   id   OBJECT IDENTIFIER,
   value  Value
}

Value ::= SEQUENCE {
   value1  INTEGER,
   value2  BOOLEAN
}

END
```

To identify the type 'Group' you have to use the null terminated string "Example.Group". These strings are used in functions that are described below.

Others examples:

Field 'id' in 'Group' type : "Example.Group.id".

Field 'value1' in field 'value' in type 'Group': "Example.Group.value.value1".

Elements of structured types that don't have a name, receive the name "?1","?2", and so on.

The name "?LAST" indicates the last element of a `SET_OF` or `SEQUENCE_OF`.

2.3 Library Notes

The header file of this library is '`libtasn1.h`'.

The main type used in it is `ASN1_TYPE`, and it's used to store the ASN.1 definitions and structures (instances).

The constant ASN1_TYPE_EMPTY can be used for the variable initialization. For example:

```
ASN1_TYPE definitions=ASN1_TYPE_EMPTY;
```

Some functions require a parameter named errorDescription of char* type. The array must be already allocated and must have at least `MAX_ERROR_DESCRIPTION_SIZE` bytes (E.g, as in `char Description[MAX_ERROR_DESCRIPTION_SIZE];`).

`MAX_NAME_SIZE` indicates the maximum number of characters of a name inside a file with ASN1 definitions.

2.4 Future developments

- Add functions for a C code file generation containing equivalent data structures (not a single array like now).
- Type REAL.

3 Utilities

3.1 Invoking asn1Parser

'`asn1Parser`' reads one file with ASN1 definitions and generates a file with an array to use with libtasn1 functions.

```
Usage:  asn1Parser [options] file

Options:
 -h : shows the help message.
 -v : shows version information and exit.
 -c : checks the syntax only.
 -o file : output file.
 -n name : array name.
```

3.2 Invoking asn1Coding

'`asn1Coding`' generates a DER encoding from a file with ASN1 definitions and another one with assignments.

The file with assignments must have this syntax:

```
InstanceName  Asn1Definition

nameString  value

nameString  value
...
```

The output file is a binary file with the DER encoding.

```
Usage:  asn1Coding [options] file1 file2
 file1 : file with ASN1 definitions.
 file2 : file with assignments.
Options:
 -h : shows the help message.
 -v : shows version information and exit.
 -c : checks the syntax only.
 -o file : output file.
```

3.3 Invoking asn1Decoding

'`asn1Decoding`' generates an ASN1 structure from a file with ASN1 definitions and a binary file with a DER encoding.

```
Usage:  asn1Decoding [options] file1 file2 type
 file1 : file with ASN1 definitions.
 file2 : binary file with a DER encoding.
 type : ASN1 definition name.
Options:
 -h : shows the help message.
```

```
-v : shows version information and exit.
-c : checks the syntax only.
-o file : output file.
```

4 Function reference

4.1 ASN.1 schema functions

asn1_parser2tree

`asn1_retCode asn1_parser2tree` (*const char * `file_name`*, *ASN1_TYPE * `definitions`*, *char * `errorDescription`*) [Function]

file_name: specify the path and the name of file that contains ASN.1 declarations.

definitions: return the pointer to the structure created from "file_name" ASN.1 declarations.

errorDescription: return the error description or an empty string if success.

Creates the structures needed to manage the definitions included in *FILE_NAME file.

Returns: ASN1_SUCCESS: The file has a correct syntax and every identifier is known.

ASN1_ELEMENT_NOT_EMPTY: *POINTER not ASN1_TYPE_EMPTY.

ASN1_FILE_NOT_FOUND: An error occured while opening FILE_NAME.

ASN1_SYNTAX_ERROR: The syntax is not correct.

ASN1_IDENTIFIER_NOT_FOUND: In the file there is an identifier that is not defined.

ASN1_NAME_TOO_LONG: In the file there is an identifier whith more than MAX_NAME_SIZE characters.

asn1_parser2array

`int asn1_parser2array` (*const char * `inputFileName`*, *const char * `outputFileName`*, *const char * `vectorName`*, *char * `errorDescription`*) [Function]

inputFileName: specify the path and the name of file that contains ASN.1 declarations.

outputFileName: specify the path and the name of file that will contain the C vector definition.

vectorName: specify the name of the C vector.

errorDescription: return the error description or an empty string if success.

Creates a file containing a C vector to use to manage the definitions included in *INPUTFILENAME file. If *INPUTFILENAME is "/aa/bb/xx.yy" and OUTPUTFILENAME is NULL, the file created is "/aa/bb/xx_asn1_tab.c". If VECTORNAME is NULL the vector name will be "xx_asn1_tab".

Returns: ASN1_SUCCESS: The file has a correct syntax and every identifier is known.

ASN1_FILE_NOT_FOUND: An error occured while opening FILE_NAME.

ASN1_SYNTAX_ERROR: The syntax is not correct.

ASN1_IDENTIFIER_NOT_FOUND: In the file there is an identifier that is not defined.

ASN1_NAME_TOO_LONG: In the file there is an identifier whith more than MAX_NAME_SIZE characters.

4.2 ASN.1 field functions

asn1_array2tree

`asn1_retCode asn1_array2tree` (*const ASN1_ARRAY_TYPE * array*, *ASN1_TYPE * definitions*, *char * errorDescription*) [Function]

array: specify the array that contains ASN.1 declarations

definitions: return the pointer to the structure created by *ARRAY ASN.1 declarations

errorDescription: return the error description.

Creates the structures needed to manage the ASN.1 definitions. `array` is a vector created by `asn1_parser2array()`.

Returns: ASN1_SUCCESS: Structure created correctly.

ASN1_ELEMENT_NOT_EMPTY: *`definitions` not ASN1_TYPE_EMPTY.

ASN1_IDENTIFIER_NOT_FOUND: In the file there is an identifier that is not defined (see `errorDescription` for more information).

ASN1_ARRAY_ERROR: The array pointed by `array` is wrong.

asn1_delete_structure

`asn1_retCode asn1_delete_structure` (*ASN1_TYPE * structure*) [Function]

structure: pointer to the structure that you want to delete.

Deletes the structure *`structure`. At the end, *`structure` is set to ASN1_TYPE_EMPTY.

Returns: ASN1_SUCCESS: Everything OK.

ASN1_ELEMENT_NOT_FOUND: *`structure` was ASN1_TYPE_EMPTY.

asn1_delete_element

`asn1_retCode asn1_delete_element` (*ASN1_TYPE structure*, *const char * element_name*) [Function]

structure: pointer to the structure that contains the element you want to delete.

element_name: element's name you want to delete.

Deletes the element named *`element_name` inside *`structure`.

Returns: ASN1_SUCCESS: Everything OK.

ASN1_ELEMENT_NOT_FOUND: The name element was not found.

asn1_create_element

`asn1_retCode asn1_create_element` (*ASN1_TYPE definitions*, *const char * source_name*, *ASN1_TYPE * element*) [Function]

definitions: pointer to the structure returned by "parser_asn1" function

source_name: the name of the type of the new structure (must be inside p_structure).

element: pointer to the structure created.

Creates a structure of type `source_name`. Example using "pkix.asn":

rc = asn1_create_structure(cert_def, "PKIX1.Certificate", certptr);

Returns: ASN1_SUCCESS: Creation OK.

ASN1_ELEMENT_NOT_FOUND: SOURCE_NAME isn't known

asn1_print_structure

`void asn1_print_structure (`*FILE * out, ASN1_TYPE structure, const char * name, int mode*`)` [Function]

out: pointer to the output file (e.g. stdout).

structure: pointer to the structure that you want to visit.

name: an element of the structure

mode: specify how much of the structure to print, can be `ASN1_PRINT_NAME`, `ASN1_PRINT_NAME_TYPE`, `ASN1_PRINT_NAME_TYPE_VALUE`, or `ASN1_PRINT_ALL`.

Prints on the `out` file descriptor the structure's tree starting from the `name` element inside the structure `structure`.

asn1_number_of_elements

`asn1_retCode asn1_number_of_elements (`*ASN1_TYPE element, const char * name, int * num*`)` [Function]

element: pointer to the root of an ASN1 structure.

name: the name of a sub-structure of ROOT.

num: pointer to an integer where the result will be stored

Counts the number of elements of a sub-structure called NAME with names equal to "?1","?2", ...

Returns: ASN1_SUCCESS: Creation OK.

ASN1_ELEMENT_NOT_FOUND: NAME isn't known.

ASN1_GENERIC_ERROR: Pointer num equal to NULL.

asn1_find_structure_from_oid

`const char * asn1_find_structure_from_oid (`*ASN1_TYPE definitions, const char * oidValue*`)` [Function]

definitions: ASN1 definitions

oidValue: value of the OID to search (e.g. "1.2.3.4").

Search the structure that is defined just after an OID definition.

Returns: NULL when OIDVALUE not found, otherwise the pointer to a constant string that contains the element name defined just after the OID.

asn1_copy_node

`asn1_retCode asn1_copy_node (`*ASN1_TYPE dst, const char * dst_name, ASN1_TYPE src, const char * src_name*`)` [Function]

dst: Destination ASN1_TYPE node.

dst_name: Field name in destination node.

src: Source ASN1_TYPE node.

src_name: Field name in source node.

Create a deep copy of a ASN1_TYPE variable.

Return value: Return ASN1_SUCCESS on success.

asn1_write_value

`asn1_retCode asn1_write_value` (*ASN1_TYPE* `node_root`, *const char ** `name`, *const void ** `ivalue`, *int* `len`) [Function]

node_root: pointer to a structure

name: the name of the element inside the structure that you want to set.

ivalue: vector used to specify the value to set. If len is >0, VALUE must be a two's complement form integer. if len=0 *VALUE must be a null terminated string with an integer value.

len: number of bytes of *value to use to set the value: value[0]..value[len-1] or 0 if value is a null terminated string

Set the value of one element inside a structure.

If an element is OPTIONAL and you want to delete it, you must use the value=NULL and len=0. Using "pkix.asn":

result=asn1_write_value(cert, "tbsCertificate.issuerUniqueID", NULL, 0);

Description for each type: INTEGER: VALUE must contain a two's complement form integer.

value[0]=0xFF , len=1 -> integer=-1. value[0]=0xFF value[1]=0xFF , len=2 -> integer=-1. value[0]=0x01 , len=1 -> integer= 1. value[0]=0x00 value[1]=0x01 , len=2 -> integer= 1. value="123" , len=0 -> integer= 123.

ENUMERATED: As INTEGER (but only with not negative numbers).

BOOLEAN: VALUE must be the null terminated string "TRUE" or "FALSE" and LEN != 0.

value="TRUE" , len=1 -> boolean=TRUE. value="FALSE" , len=1 -> boolean=FALSE.

OBJECT IDENTIFIER: VALUE must be a null terminated string with each number separated by a dot (e.g. "1.2.3.543.1"). LEN != 0.

value="1 2 840 10040 4 3" , len=1 -> OID=dsa-with-sha.

UTCTime: VALUE must be a null terminated string in one of these formats: "YYMMDDhhmmssZ", "YYMMDDhhmmssZ", "YYMMDDhhmmss+hh'mm'", "YYMMDDhhmmss-hh'mm'", "YYMMDDhhmm+hh'mm'", or "YYMMDDhhmm-hh'mm'". LEN != 0.

value="9801011200Z" , len=1 -> time=Jannuary 1st, 1998 at 12h 00m Greenwich Mean Time

GeneralizedTime: VALUE must be in one of this format: "YYYYMMDDhhmmss.sZ", "YYYYMMDDhhmmss.sZ", "YYYYMMDDhhmmss.s+hh'mm'", "YYYYMMDDhhmmss.s-hh'mm'", "YYYYMMDDhhmm+hh'mm'", or

"YYYYMMDDhhmm-hh'mm'" where ss.s indicates the seconds with any precision like "10.1" or "01.02". LEN != 0

value="2001010112001.12-0700" , len=1 -> time=Jannuary 1st, 2001 at 12h 00m 01.12s Pacific Daylight Time

OCTET STRING: VALUE contains the octet string and LEN is the number of octets.

value="$\backslash$x01$\backslash$x02$\backslash$x03" , len=3 -> three bytes octet string

GeneralString: VALUE contains the generalstring and LEN is the number of octets.

value="$\backslash$x01$\backslash$x02$\backslash$x03" , len=3 -> three bytes generalstring

BIT STRING: VALUE contains the bit string organized by bytes and LEN is the number of bits.

value="$\backslash$xCF" , len=6 -> bit string="110011" (six bits)

CHOICE: if NAME indicates a choice type, VALUE must specify one of the alternatives with a null terminated string. LEN != 0. Using "pkix.asn"\:

result=asn1_write_value(cert, "certificate1.tbsCertificate.subject", "rdnSequence", 1);

ANY: VALUE indicates the der encoding of a structure. LEN != 0.

SEQUENCE OF: VALUE must be the null terminated string "NEW" and LEN != 0. With this instruction another element is appended in the sequence. The name of this element will be "?1" if it's the first one, "?2" for the second and so on.

Using "pkix.asn"\:

result=asn1_write_value(cert, "certificate1.tbsCertificate.subject.rdnSequence", "NEW", 1);

SET OF: the same as SEQUENCE OF. Using "pkix.asn":

result=asn1_write_value(cert, "tbsCertificate.subject.rdnSequence.?LAST", "NEW", 1);

Returns: ASN1_SUCCESS: Set value OK.

ASN1_ELEMENT_NOT_FOUND: NAME is not a valid element.

ASN1_VALUE_NOT_VALID: VALUE has a wrong format.

asn1_read_value

`asn1_retCode asn1_read_value` (*ASN1_TYPE* `root`, *const char ** `name`, *void ** `ivalue`, *int ** `len`) [Function]

root: pointer to a structure.

name: the name of the element inside a structure that you want to read.

ivalue: vector that will contain the element's content, must be a pointer to memory cells already allocated.

len: number of bytes of *value: value[0]..value[len-1]. Initialy holds the sizeof value.

Returns the value of one element inside a structure.

If an element is OPTIONAL and the function "read_value" returns ASN1_ELEMENT_NOT_FOUND, it means that this element wasn't present in the der encoding that created the structure. The first element of a SEQUENCE_OF or SET_OF is named "?1". The second one "?2" and so on.

INTEGER: VALUE will contain a two's complement form integer.

integer=-1 -> value[0]=0xFF , len=1. integer=1 -> value[0]=0x01 , len=1.

ENUMERATED: As INTEGER (but only with not negative numbers).

BOOLEAN: VALUE will be the null terminated string "TRUE" or "FALSE" and LEN=5 or LEN=6.

OBJECT IDENTIFIER: VALUE will be a null terminated string with each number separated by a dot (i.e. "1.2.3.543.1").

LEN = strlen(VALUE)+1

UTCTime: VALUE will be a null terminated string in one of these formats: "YYMMDDhhmmss+hh'mm'" or "YYMMDDhhmmss-hh'mm'". LEN=strlen(VALUE)+1.

GeneralizedTime: VALUE will be a null terminated string in the same format used to set the value.

OCTET STRING: VALUE will contain the octet string and LEN will be the number of octets.

GeneralString: VALUE will contain the generalstring and LEN will be the number of octets.

BIT STRING: VALUE will contain the bit string organized by bytes and LEN will be the number of bits.

CHOICE: If NAME indicates a choice type, VALUE will specify the alternative selected.

ANY: If NAME indicates an any type, VALUE will indicate the DER encoding of the structure actually used.

Returns: ASN1_SUCCESS: Set value OK.

ASN1_ELEMENT_NOT_FOUND: NAME is not a valid element.

ASN1_VALUE_NOT_FOUND: There isn't any value for the element selected.

ASN1_MEM_ERROR: The value vector isn't big enough to store the result. In this case LEN will contain the number of bytes needed.

asn1_read_tag

`asn1_retCode asn1_read_tag (`*node_asn * **root***, *const char * **name***, *int * **tagValue***, *int * **classValue***)` [Function]

root: pointer to a structure

name: the name of the element inside a structure.

tagValue: variable that will contain the TAG value.

classValue: variable that will specify the TAG type.

Returns the TAG and the CLASS of one element inside a structure.

CLASS can have one of these constants: ASN1_CLASS_APPLICATION, ASN1_CLASS_UNIVERSAL, ASN1_CLASS_PRIVATE or ASN1_CLASS_CONTEXT_SPECIFIC.

Returns: ASN1_SUCCESS: Set value OK.
ASN1_ELEMENT_NOT_FOUND: NAME is not a valid element.

4.3 DER functions

asn1_length_der

`void asn1_length_der (`*unsigned long int* `len`, *unsigned char ** `ans`, *int ** `ans_len``)` [Function]

len: value to convert.

ans: string returned.

ans_len: number of meaningful bytes of ANS (ans[0]..ans[ans_len-1]).

Creates the DER coding for the LEN parameter (only the length). The `ans` buffer is pre-allocated and must have room for the output.

asn1_octet_der

`void asn1_octet_der (`*const unsigned char ** `str`, *int* `str_len`, *unsigned char ** `der`, *int ** `der_len``)` [Function]

str: OCTET string.

str_len: STR length (str[0]..str[str_len-1]).

der: string returned.

der_len: number of meaningful bytes of DER (der[0]..der[ans_len-1]).

Creates the DER coding for an OCTET type (length included).

asn1_bit_der

`void asn1_bit_der (`*const unsigned char ** `str`, *int* `bit_len`, *unsigned char ** `der`, *int ** `der_len``)` [Function]

str: BIT string.

bit_len: number of meaningful bits in STR.

der: string returned.

der_len: number of meaningful bytes of DER (der[0]..der[ans_len-1]).

Creates the DER coding for a BIT STRING type (length and pad included).

asn1_der_coding

`asn1_retCode asn1_der_coding (`*ASN1_TYPE* `element`, *const char ** `name`, *void ** `ider`, *int ** `len`, *char ** `ErrorDescription``)` [Function]

element: pointer to an ASN1 element

name: the name of the structure you want to encode (it must be inside *POINTER).

ider: vector that will contain the DER encoding. DER must be a pointer to memory cells already allocated.

len: number of bytes of *`ider`: `ider`[0]..`ider`[len-1], Initialy holds the sizeof of der vector.

Creates the DER encoding for the NAME structure (inside *POINTER structure).

Returns: ASN1_SUCCESS: DER encoding OK.

ASN1_ELEMENT_NOT_FOUND: NAME is not a valid element.

ASN1_VALUE_NOT_FOUND: There is an element without a value.

ASN1_MEM_ERROR: `ider` vector isn't big enough. Also in this case LEN will contain the length needed.

asn1_get_length_der

`signed long asn1_get_length_der` (*const unsigned char * `der`, int `der_len`, int * `len`*) [Function]

der: DER data to decode.

der_len: Length of DER data to decode.

len: Output variable containing the length of the DER length field.

Extract a length field from DER data.

Return value: Return the decoded length value, or -1 on indefinite length, or -2 when the value was too big.

asn1_get_tag_der

`int asn1_get_tag_der` (*const unsigned char * `der`, int `der_len`, unsigned char * `cls`, int * `len`, unsigned long * `tag`*) [Function]

der: DER data to decode.

der_len: Length of DER data to decode.

cls: Output variable containing decoded class.

len: Output variable containing the length of the DER TAG data.

tag: Output variable containing the decoded tag.

Decode the class and TAG from DER code.

Return value: Returns ASN1_SUCCESS on success, or an error.

asn1_get_octet_der

`int asn1_get_octet_der` (*const unsigned char * `der`, int `der_len`, int * `ret_len`, unsigned char * `str`, int `str_size`, int * `str_len`*) [Function]

der: DER data to decode containing the OCTET SEQUENCE.

der_len: Length of DER data to decode.

ret_len: Output variable containing the length of the DER data.

str: Pre-allocated output buffer to put decoded OCTET SEQUENCE in.

str_size: Length of pre-allocated output buffer.

str_len: Output variable containing the length of the OCTET SEQUENCE.

Extract an OCTET SEQUENCE from DER data.

Return value: Returns ASN1_SUCCESS on success, or an error.

asn1_get_bit_der

`int asn1_get_bit_der` (*const unsigned char * der, int der_len, int * ret_len, unsigned char * str, int str_size, int * bit_len*) [Function]

der: DER data to decode containing the BIT SEQUENCE.

der_len: Length of DER data to decode.

ret_len: Output variable containing the length of the DER data.

str: Pre-allocated output buffer to put decoded BIT SEQUENCE in.

str_size: Length of pre-allocated output buffer.

bit_len: Output variable containing the size of the BIT SEQUENCE.

Extract a BIT SEQUENCE from DER data.

Return value: Return ASN1_SUCCESS on success, or an error.

asn1_der_decoding

`asn1_retCode asn1_der_decoding` (*ASN1_TYPE * element, const void * ider, int len, char * errorDescription*) [Function]

element: pointer to an ASN1 structure.

ider: vector that contains the DER encoding.

len: number of bytes of *`ider`: `ider`[0]..`ider`[len-1].

errorDescription: null-terminated string contains details when an error occurred.

Fill the structure *ELEMENT with values of a DER encoding string. The sructure must just be created with function 'create_stucture'. If an error occurs during the decoding procedure, the *ELEMENT is deleted and set equal to `ASN1_TYPE_EMPTY`.

Returns: ASN1_SUCCESS: DER encoding OK.

ASN1_ELEMENT_NOT_FOUND: ELEMENT is ASN1_TYPE_EMPTY.

ASN1_TAG_ERROR,ASN1_DER_ERROR: The der encoding doesn't match the structure NAME. *ELEMENT deleted.

asn1_der_decoding_element

`asn1_retCode asn1_der_decoding_element` (*ASN1_TYPE * structure, const char * elementName, const void * ider, int len, char * errorDescription*) [Function]

structure: pointer to an ASN1 structure

elementName: name of the element to fill

ider: vector that contains the DER encoding of the whole structure.

len: number of bytes of *der: der[0]..der[len-1]

errorDescription: null-terminated string contains details when an error occurred.

Fill the element named ELEMENTNAME with values of a DER encoding string. The sructure must just be created with function 'create_stucture'. The DER vector must contain the encoding string of the whole STRUCTURE. If an error occurs during the decoding procedure, the *STRUCTURE is deleted and set equal to `ASN1_TYPE_EMPTY`.

Returns: ASN1_SUCCESS: DER encoding OK.

ASN1_ELEMENT_NOT_FOUND: ELEMENT is ASN1_TYPE_EMPTY or elementName == NULL.

ASN1_TAG_ERROR,ASN1_DER_ERROR: The der encoding doesn't match the structure STRUCTURE. *ELEMENT deleted.

asn1_der_decoding_startEnd

`asn1_retCode asn1_der_decoding_startEnd` (*ASN1_TYPE* `element`, *const void ** `ider`, *int* `len`, *const char ** `name_element`, *int ** `start`, *int ** `end`) [Function]

element: pointer to an ASN1 element

ider: vector that contains the DER encoding.

len: number of bytes of *`ider`: `ider`[0]..`ider`[len-1]

name_element: an element of NAME structure.

start: the position of the first byte of NAME_ELEMENT decoding (`ider`[*start])

end: the position of the last byte of NAME_ELEMENT decoding (`ider`[*end])

Find the start and end point of an element in a DER encoding string. I mean that if you have a der encoding and you have already used the function "asn1_der_decoding" to fill a structure, it may happen that you want to find the piece of string concerning an element of the structure.

Example: the sequence "tbsCertificate" inside an X509 certificate.

Returns: ASN1_SUCCESS: DER encoding OK.

ASN1_ELEMENT_NOT_FOUND: ELEMENT is ASN1_TYPE EMPTY or NAME_ELEMENT is not a valid element.

ASN1_TAG_ERROR,ASN1_DER_ERROR: the der encoding doesn't match the structure ELEMENT.

asn1_expand_any_defined_by

`asn1_retCode asn1_expand_any_defined_by` (*ASN1_TYPE* `definitions`, *ASN1_TYPE ** `element`) [Function]

definitions: ASN1 definitions

element: pointer to an ASN1 structure

Expands every "ANY DEFINED BY" element of a structure created from a DER decoding process (asn1_der_decoding function). The element ANY must be defined by an OBJECT IDENTIFIER. The type used to expand the element ANY is the first one following the definition of the actual value of the OBJECT IDENTIFIER.

Returns: ASN1_SUCCESS: Substitution OK.

ASN1_ERROR_TYPE_ANY: Some "ANY DEFINED BY" element couldn't be expanded due to a problem in OBJECT_ID -> TYPE association.

other errors: Result of der decoding process.

asn1_expand_octet_string

`asn1_retCode asn1_expand_octet_string` (*ASN1_TYPE* `definitions`, *ASN1_TYPE* * `element`, *const char* * `octetName`, *const char* * `objectName`) [Function]

definitions: ASN1 definitions

element: pointer to an ASN1 structure

octetName: name of the OCTECT STRING field to expand.

objectName: name of the OBJECT IDENTIFIER field to use to define the type for expansion.

Expands an "OCTET STRING" element of a structure created from a DER decoding process (asn1_der_decoding function). The type used for expansion is the first one following the definition of the actual value of the OBJECT IDENTIFIER indicated by OBJECTNAME.

Returns: ASN1_SUCCESS: Substitution OK.

ASN1_ELEMENT_NOT_FOUND: OBJECTNAME or OCTETNAME are not correct.

ASN1_VALUE_NOT_VALID: Wasn't possible to find the type to use for expansion.

other errors: result of der decoding process.

4.4 Error handling functions

libtasn1_perror

`void libtasn1_perror` (*asn1_retCode* `error`) [Function]

error: is an error returned by a libtasn1 function.

This function is like `perror()`. The only difference is that it accepts an error returned by a libtasn1 function.

libtasn1_strerror

`const char * libtasn1_strerror` (*asn1_retCode* `error`) [Function]

error: is an error returned by a libtasn1 function.

This function is similar to `strerror()`. The only difference is that it accepts an error (number) returned by a libtasn1 function.

Returns: Pointer to static zero-terminated string describing error code.

4.5 Auxilliary functions

asn1_find_node

`ASN1_TYPE asn1_find_node` (*ASN1_TYPE* `pointer`, *const char* * `name`) [Function]

pointer: NODE_ASN element pointer.

name: null terminated string with the element's name to find.

Searches for an element called NAME starting from POINTER. The name is composed by differents identifiers separated by dots. When *POINTER has a name, the first identifier must be the name of *POINTER, otherwise it must be the name of one child of *POINTER.

Return value: the searching result. NULL if not found.

asn1_check_version

`const char * asn1_check_version` (*const char * `req_version`*) [Function]

req_version: Required version number, or NULL.

Check that the the version of the library is at minimum the requested one and return the version string; return `NULL` if the condition is not satisfied. If a `NULL` is passed to this function, no check is done, but the version string is simply returned.

See `LIBTASN1_VERSION` for a suitable `req_version` string.

Return value: Version string of run-time library, or `NULL` if the run-time library does not meet the required version number.

Appendix A Copying Information

A.1 GNU Free Documentation License

Version 1.2, November 2002

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 non-commercially. 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.

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 for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

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.

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 1.2
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.

A.2 GNU Lesser General Public License

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the *Lesser* General Public License because it does *Less* to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to

use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

   A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

   The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

   "Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a. The modified work must itself be a software library.

b. You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c. You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d. If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

   Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

   This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code,

which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

   However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

   When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

   If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

   Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

   You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

   a. Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions

files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d. If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e. Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

   a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

   b. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative

works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

    If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

    It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

    This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public 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.

    Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published

by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the library's name and an idea of what it does.
Copyright (C) year  name of author

This library is free software; you can redistribute it and/or modify it
under the terms of the GNU Lesser General Public License as published by
the Free Software Foundation; either version 2.1 of the License, or (at
your option) any later version.

This library 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
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
USA.
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the library
'Frob' (a library for tweaking knobs) written by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

A.3 GNU General Public License

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is

intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

   "This License" refers to version 3 of the GNU General Public License.

   "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

   The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

   A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

   The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

   The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition

files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

   All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

   You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

   Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

   No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

   When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

   You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

   You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

   You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

   a. The work must carry prominent notices stating that you modified it, and giving a relevant date.
   b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
   c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
   d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

   A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

   You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

   a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
   b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
   c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself

materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
d. Limiting the use for publicity purposes of names of licensors or authors of the material; or
e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

   You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

   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, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

   You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

    Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

    An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of

distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

    If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

    Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU General Public 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.

    Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

    If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

    Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

    THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN

WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

    If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or (at
your option) any later version.

This program 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.

You should have received a copy of the GNU General Public License
along with this program.  If not, see http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
program Copyright (C) year name of author
This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands ‘`show w`’ and ‘`show c`’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html.

Concept Index

Function and Data Index

A

L