Journaled File LIBrary (libjf) FAQ

“Frequently Asked Questions”

Christian Ferrari

tiian@users.sourceforge.net

1. Global

1. Can I help fixing English language related mistakes of these FAQs?

Sure, please send me a better version at <`tiian@users.sourceforge.net`>

2. Can I suggest new FAQs?

Sure, please send me it at <`tiian@users.sourceforge.net`>

3. Is libjf “Yet Another Database Tool”?

No, libjf is a tool can be used to add application and/or system transactionality to your file based applications: it does not matter how your files are used (text, binary, search tree, heap, etc...). An interesting feature is the “transactional read” feature you can use with “R” and “R+” open modes: you can start reading from the point you leaved at previous execution without writing this information in some different place.

4. Is libjf “Yet Another Journaled File System”?

No, libjf API offers application level recovery and is independent from specific filesystem/operating system. The effort is to develop only “ANSI C” code, but some exceptions are necessary to avoid bugged or foolish code.

5. Is there a similar tool?

Alberto Bertogli noticed me he developed libjio. “Libjf” and “libjio” are *distinct original implementations* that solve slightly different problems in the same class: please feed-back me your opinion about “libjf vs. libjio”.

6. Is there a “step by step tutorial”?

Yes, it’s in `doc/tutorial` directory.

7. Is there an “API reference guide”?

Yes, it’s in `doc/api_reference` directory.

8. What did convince you to develop libjf?

I was interested in developing a search technology to manage impressive number of keys, from 10^12 up, with search time and insert time bounded by O(K) where K is the length of the key you are searching/inserting. After some early drafts I realized there's no usage for a huge index if recovery is not guaranteed: you can *not* insert again 10^12 keys because an application/system crash happened while updating the index and something "gone wrong". A journaling technology was necessary. Developing and improving libjf is a hard task so I don't think I'll be able to develop "tera index" technology too.

9. What's a "journal file"?

It's a special file handled by libjf library and used to store transactional data related to one or more journaled files.

10. What's a "journaled file"?

It's a normal file created and managed by libjf: using libjf API you can perform ACID transactions on one or more journaled files as you are used with relational databases. A journaled file can be read with any program like **cat**, **less**, etc..., but can be written *only* using libjf API. Writing a journaled file with different tools breaks recovery properties and the file may be damaged when journal file is opened.

11. What's a "tested environment"?

It's a tuple (operating system, libc, C compiler) with these properties:

- libjf compiles without warning
- **stress_test.sh** executes without errors.

12. Is there any warranty if I use libjf in a tested environment?

No. 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.

13. How can I add a new "tested environment" to `doc/tested_arch.xml` list?

Please be sure the new environment complies with these properties:

- libjf compiles without warnings
- **stress_test.sh** executes without errors.

Fill in a new `tested_arch_record.xml` record specifying *all* tags. Supposing you successfully tested the library distributed with package `libjf-1.2.3-4.2.1-20051231.tar.gz` you may retrieve the information you need using this procedure:

- package_version: first 3 numbers, `1.2.3`
- library_version: following 3 number, `4.2.1`
- date: execute command

  ```
  date +%Y-%m-%d
  ```

- description: take a look to original operating system package (CDs, DVDs, etc...)

- config_guess: execute command

  ```
  ./config.guess
  ```

  example:

  ```
  ./config.guess
  i686-suse-linux
  ```

  - host_cpu: "i686"
  - host_vendor: "suse"
  - host_os: "linux"

- libc_version: on GNU/Linux rpm-based systems, the command is

  ```
  rpm -qf /lib/libc.so.6
  ```

  on different operating systems you should use the specific packaging tool (“smit”, “sam”, “*pkg*”, ...)

- cc_version: many C compiler accepts `-v` option to retrieve the version:

  ```
  cc -v
  ```

  some C compilers may need something else...

Records lacking of any data will not be included in official documentation. Send to <`tiian@users.sourceforge.net`> your fresh `tested_arch_record.xml`

14. What’s “recovery pending” status for a journal?

“Recovery pending” means: “application and/or system crashed before a valid synchronization point and some operations must be performed in order to recover journaled files”. “Recovery pending” is a common status in a transactional system.

15. What’s “damaged journal” status for a journal?

“Damaged journal” means: “the journal contains at least a record that’s not *well formed*”. Recovery can be performed only truncating the journal at last valid record. If damage appears in the middle of the journal - not at tail position - the journal itself is compromised and recovery is not suggested. “Damaged journal” must *not* be common status in a transactional system.

16. Is there a porting for Microsoft Windows operating systems?

A native porting is not yet available, but cygwin emulation has been successfully used to compile and test libjf. A native porting will be done, but there’s no a release date for this milestone.

17. How slow is libjf compared to C standard streams (stdio)?

libjf has to make a lot of work to supply transactional integrity so the comparison is not a 1:1 race. Take a look to Section 3.3 to read more about libjf benchmarking.

2. Development

1. Can I use relative paths instead of absolute paths when opening journaled files and journals with functions `jf_file_open` and `jf_journal_open`?

Yes, you may, but some precautions must be taken: when you re-use a journaled file created using a relative path you must start your program from the same current dir of creation time because journal file keeps original relative path. Some issues may happen if you have symbolic link in your path. As a general rule, it's strongly suggested to use absolute paths if journal file and related journaled files are not stored in the same directory.

2. How can I guess which options was selected at configure/compile time?

Commands:

```
strings /opt/libjf/lib/libjf | grep 'feature/'
```

retrieve which features are active in compiled library.

3. My project is using libjf, how can I register it?

Compile a `linked_proj_record.xml` record and send to <`tiian@users.sourceforge.net`>

4. I would like to port libjf to some different operating system, will you merge my fixes & patches?

After revision, your porting fixes and patches will be merged in main distribution and you will be cited as a contributor. A specific forum has been created on SourceForge.net for porting related discussions.

5. Can I use libjf in a multi-threaded program?

Yes, you may, but libjf objects are not multi-thread self protected: it's your responsibility avoid the same set of objects (a journal and its related journaled files) is accessed at the same time by two or more threads. At this time the support is guaranteed only by reentrant code: there is no static variable with a little exception can be ignored in most cases (it's a trivial counter used only when `--enable-crash-simul` build option is active).

6. How can I recover journaled files after a crash?

Your application must set *`recovery_enabled`* = `TRUE` in jf_journal_opts_s struct passed to `jf_journal_open` function. This type of recovery is called "automatic recovery" and is not enabled by default (you must set it).

7. How can I recover a damaged journal and its journaled files after a crash?

Sometimes, the crash may damage the journal itself, but some errors can be fixed with a "forced recovery". Your application must set *`recovery_enabled`* = `TRUE` and *`recovery_damaged_journal`* = `TRUE` in jf_journal_opts_s struct passed to `jf_journal_open` function. This type of recovery is called "automatic forced recovery" and is not enabled by default (you must set it).

8. Are text files supported?

Yes, libjf supports binary files as well as text files, but there are some differences between standard ANSI C streams and libjf journaled text files:

- UNIX text files, with records delimited by NL ("\n") must be opened like binary files, for example: "r", "w".
- DOS text files, with records delimited by CRLF ("\r\n") must be opened with a special ("DOS mode") flag, for example: "rD", "wD".

Please pay attention: libjf will translate CRLF ("\r\n") to NL ("\n") when reading and NL ("\n") to CRLF ("\r\n") when writing a file opened appending flag "D" to "open mode" string. If the journaled file is not opened specifying "DOS mode", the application has to deal with "\r\n" sequences. The semantic of libjf is slightly different from standard ANSI C streams: your application can deal with a file encoded for a different architecture without convert if before usage. For example, the configuration journaled file of your application might be encoded using "DOS mode" while running on DOS-like operating system or UNIX-like operating system. Additional encodings may be added in the future (MAC OS?) if someone will ask for them.

9. What's the difference between `fsync` and `jf_journal_commit`?

`fsync` synchronize the content of ONE file at a time, `jf_journal_commit` synchronize ALL the files at the same time preserving consistency in the event of an application or system crash.

2.1. Testing

1. Is there a case test procedure?

Yes, sure: I'm spending half of the time in developing case tests! Use **make check** command to start case test procedure.

2. Where do case test scripts write journals, journaled files and standard files?

Default path is `/tmp/libjf`, but you can change it setting environment var `TEST_TMP_PATH` before configure operation; this is an example:

```
> export TEST_TMP_PATH=/tmp/foo/bar
> ./configure
```

3. Is there a *stress test* procedure?

Yes, it runs case test trying all the available configurations. These are the commands to start it:

```
> ./configure
> cd tests
> ./stress_test.sh
```

4. Stress test procedure is very slow, can I speed it up?

Stress test procedure may be very slow, especially if you set environment variable `JF_JOURNAL_SYNC_TYPE` to "1" and use *safe* synchronization. If your operating system (and kernel) supports it, you can reduce elapsed time, using a "TMPFS" filesystem. If you are riding GNU/linux 2.4+ with "TMPFS" feature active, you can create a "TMPFS" filesystem for your `/tmp` mount point inserting something like this in your `/etc/fstab`:

```
none    /tmp    tmpfs    defaults    0 0
```

"TMPFS" is available on some flavor of UNIX systems.

3. System

1. How can I inspect the content of a journal file?

Use utility program **jf_report** to obtain an XML dump of journal file content. Some options are available to increase/reduce verbosity.

2. Can I change the type of synchronization for a journal file without changing the program?

If the program open the journal using flag `JF_JOURNAL_PROP_SYNC_DEFAULT` or `JF_JOURNAL_PROP_SYNC_ENV_VAR`, synchronization may be changed using environment var `JF_JOURNAL_SYNC_TYPE`:

export JF_JOURNAL_SYNC_TYPE=0

for fast synchronization (`fflush`)

export JF_JOURNAL_SYNC_TYPE=1

for safe synchronization (`fdatasync`)

3. Which type of synchronization should I choose?

`JF_JOURNAL_PROP_SYNC_FAST` is fast and works fine if system crash (hardware, operating system, power supply) is a seldom event you don't have to care of. `JF_JOURNAL_PROP_SYNC_SAFE` is slow and works fine when system crash is a frequent event or you can not deal with its consequences. Good I/O devices are necessary to achieve good performances.

4. Is there a list of environment vars recognized by libjf?

Yes, this is the list:

`JF_TRACE_MASK`

exadecimal value containing the mask of traced modules (purpose: development & debugging)

`JF_JOURNALED_FILE_CACHE_SIZE`

number of bytes used to cache a journaled file;

Note: the value is used only if:

- it's greater than hard wired constant `JF_CACHE_FILE_DEFAULT_LIMIT`
- program does not specify a specific value

`JF_CRASH_SIMUL_POINT`

place where crash must happen; note: the value is used only if library has been built with `--enable-crash-simul` option (purpose: development & debugging)

`JF_CRASH_SIMUL_COUNT`

number of times the process must cross the crash simulation point before a crash can happen; note: the value is used only if library has been built with `--enable-crash-simul` option (purpose: development & debugging)

`JF_JOURNAL_SYNC_TYPE`

type of synchronization (fast or safe) must be used; note: the value is used only if the program does not specify a specific value

`JF_JOURNAL_VIRTMEM`

virtual memory usage for internal buffering purposes; note: the value is used only if the program does not specify a specific value

`JF_JOURNAL_SIZE`

journal file size; note: the value is used at creation time only if the program does not specify a specific value

`JF_JOURNAL_NUM`

number of journal files must be kept in rotation process; note: the value is used at creation time only if the program does not specify a specific value

`JF_JOURNAL_ROTATION_THRESHOLD`

filling up ratio must be reached before a journal rotation can happen; note: the value is used at creation time only if the program does not specify a specific value.

3.1. Utilities

1. How can I create a journal without writing my own C program?

You can use utility **jf_create** to create a new journal file; journaled files may be added using utility **jf_join**.

2. How can I add a standard system file to a journal file without writing my own C program?

You can use utility **jf_join** where “join” means “a new standard file joins the group of files journaled using a specific journal file”.

3. How can I remove a journaled file from the control of a journal without writing my own C program?

You can use utility **jf_leave** where “leave” means “a journaled file leaves its journal and become a standard file”.

4. Can I rename a journaled file using standard system command **mv**?

No, renaming the file breaks the link between journal file and journaled files. To rename a journaled file, use utility **jf_rename**.

5. Can I move a journaled file to a different filesystem using utility **jf_rename**?

If your `rename` (**man 2 rename**) implementation supports move across filesystems yes, otherwise no. GNU/Linux 2.6.x and glibc 2.3.3 do not support this type of operation (`EXDEV` error is returned).

3.2. Recovery

1. How can I recover journaled files after a crash?

You can use utility **jf_recover** specifically designed for this purpose. This is a quite flexible utility can be used:

- to check if a journal needs recovery
- to create an XML report with the operations would be performed if a recovery phase was performed
- to perform a recovery phase.

These functions apply to “recovery pending” journals and “damaged journals” too.

2. How can I verify the “recovery pending” status of a journal?

Use utility **jf_recover** with `-t` flag:

```
jf_recover -j <my journal name> -t
```

and check exit code:

- 0: journal needs recovery
- 1: journal is damaged

- 2: error
- 3: journal is not in “recovery pending” status.

3. How can I recover from a “recovery pending” status?

Use utility “jf_recover” and check exit code is 0. Example:

```
jf_recover -j <my journal name>
```

4. How can I establish if a journal is damaged?

Use utility **jf_recover** with `-t` and `-f` flags:

```
jf_recover -j <my journal name> -t
```

and check exit code is 1, then

```
jf_recover -j <my journal name> -t -f
```

and check exit code is 0

5. How can I recover a damaged journal?

Use utility **jf_recover** specifying `-f` (“force”) flag and check exit code is 0. Example:

```
jf_recover -j <my journal name> -f
```

6. How can I guess the operations would be performed if recovery phase was performed?

Use utility **jf_recover** with options `-t` (“test”) and `-d` (“dump”). Example:

```
jf_recover -j <my journal name> -t -dh
```

Type **jf_recover -h** to see the list of sub-options related to `-d`

7. How can I see the operations performed in a recovery phase?

Use utility **jf_recover** with option `-d` (“dump”). Example:

```
jf_recover -j <my journal name> -dh
```

Type **jf_recover -h** to see the list of sub-options related to `-d`.

3.3. Tuning

1. What is the maximum size of a journal?

Journal maximum size is determined at creation time:

- the program can specify it setting *`file_size`* field of struct jf_journal_file_opts_s
- if the program does not specify it, the value of environment var `JF_JOURNAL_SIZE` is used
- if environment var `JF_JOURNAL_SIZE` is not set, the default value (see hard wired constant `JF_JOURNAL_DEFAULT_FILE_SIZE`) is used.

The maximum size of a journal file can be retrieved using utility **jf_report**.

2. How many backup journals will be kept?

Journal file can NOT be expanded indefinitely by design. A full journal is archived appending suffix “1” and a new journal file is created: this is called “rotation process”. The number of old journals must be kept is determined at creation time:

- the program can specify it setting *`file_num`* field of struct jf_journal_file_opts_s
- if the program does not specify it, the value of environment var `JF_JOURNAL_NUM` is used
- if environment var `JF_JOURNAL_NUM` is not set, the default value (see hard wired constant `JF_JOURNAL_DEFAULT_FILE_NUM`) is used.

The number of old journals kept by a journal file can be retrieved using utility **jf_report**.

3. At what time does “rotation process” happen?

A journal file rotates when “rotation threshold” is reached and a *global sync point* is asked by the application. “Rotation threshold” is a number in range (0 , 1.0)

Example 1. rotation does not happen because journal file is not full

file_size = 4 Mbytes
rotation_threshold = 0.75
journal file current size = 2.3 Mbytes
application calls `jf_journal_commit()`

Example 2. rotation does happen because journal file is “quite” full

file_size = 4 Mbytes
rotation_threshold = 0.75
journal file current size = 3.2 Mbytes
application calls `jf_journal_commit()`

4. How can rotation threshold be set?

Journal rotation threshold is determined at creation time:

- the program can specify it setting *`rotation_threshold`* field of struct jf_journal_file_opts_s
- if the program does not specify it, the value of environment var `JF_JOURNAL_ROTATION_THRESHOLD` is used
- if environment var `JF_JOURNAL_ROTATION_THRESHOLD` is not set, the default value (see hard wired constant `JF_JOURNAL_DEFAULT_ROTATION_THRESHOLD`) is used.

The rotation threshold of a journal file can be retrieved using utility **jf_report**.

5. How can the cache of a journaled file be tuned?

Cache associated to a journaled file is determined at open time:

- the program can specify it setting field *`cache_size_limit`* of struct jf_journal_file_opts_s of struct jf_journal_opts_s of struct jf_file_open_opts_s
- if the program does not specify it, the value of environment var `JF_JOURNALED_FILE_CACHE_SIZE` is used
- if environment var `JF_JOURNALED_FILE_CACHE_SIZE` is not set, the default value (see hard wired constant `JF_CACHE_FILE_DEFAULT_LIMIT`) is used.

Cache size can not be set at value lower then `JF_CACHE_FILE_MIN_LIMIT` (it's a hard wired constant).

6. Is there an official libjf benchmark tool?

Yes, utility **jf_bench** is the official tool.

7. Why is libjf slower than stdio?

First, libjf is a recent experimental piece of code, while stdio is a well tested one. Second, libjf is based on stdio to avoid wheel design. Last but not least, libjf has a reacher semantic that allows transactionality, while stdio does not support transactions.

8. How fast/slow is an application developed with libjf in comparison with stdio?

Applications that use "safe" synchronization (disk synchronization) tend to use twice the time when moved from stdio to libjf. Applications that use "fast" synchronization (buffer flush) tend to slow down a lot (4-5 times) when moved from stdio to libjf.

9. Why should I use libjf if it is "so slow"?

Because stdio does not provide a rollback function. Because stdio does not provide a synchronization function for more than one stream/file descriptor: your system may crash after synchronization of first stream/file descriptor and before synchronization of second stream/file descriptor.

10. Which factors do influence **jf_bench** results?

There are many:

- filesystem type
- storage type (EIDE, SATA, SCSI, etc...)

- device type (native, RAID, DRBD, crypto, etc...)
- RAM and cache size and speed
- CPU type and speed.

11. Can I guess how slow will become my application after migrating it from stdio to libjf?

If your application is CPU intensive and occasionally write something to disk, don't mind about libjf slow down. If your application writes a significative amount of data on disk, you can expect a slow down between 5% and 20%. If your application is I/O intensive, the results of **jf_bench** utility can help you guessing how elapsed time will increase and how CPU, user and system, time will increase. **jf_bench** does nothing with data written and read to/from disk so it should be the worst case your application might reach.

12. Is there any difference between "append" and "update" from a performance point of view?

Yes, the slow-down introduced by libjf is correlated to the type of write an application requires. **jf_bench** utility program can help you with average results and pattern specific results.

13. Can I obtain raw test data from **jf_bench**?

Yes, **jf_bench** prints on terminal average results, but you can ask it to supply all the raw data in CSV and/or XML format.

14. Can I use different block size, number of records, number of files, etc... with **jf_bench**?

Yes, you may specify different parameters changing the constants defined in the source code and recompiling it. The official version of **jf_bench** might change these values in the future: no one know "the right values"...

15. Is libjf already optimized or should we expect better results for **jf_bench** in the future?

At this time libjf is "alpha" software and no optimization work has been done.

16. **jf_bench** utility program stops and shows this error:

```
jf_bench/bench_test_results_compute: error while computing
benchmark results: -20/ERROR: journal file exceeds maximum
desired size and operation can NOT be performed; a global
sync/rollback is necessary to activate journal rotation
```

Sometimes default journal size is not sufficient for benchmark execution and environment variable `JF_JOURNAL_SIZE` must be tuned to a value higher than default; you may use this command before execution to set journal size at 256 Mbytes:

```
export JF_JOURNAL_SIZE=268435456
```

A. GNU Free Documentation License

A.1. 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.

A.2. 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.

A.3. 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.

A.4. 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.

A.5. 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:

GNU FDL Modification Conditions

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.

A.6. 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".

A.7. 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.

A.8. 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.

A.9. 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.

A.10. 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.

A.11. 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.

A.12. 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:

Sample Invariant Sections list

> 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:

Sample Invariant Sections list

> 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.