

GNU Offloading and Multi Processing Runtime Library

The GNU OpenMP and OpenACC Implementation

Published by the Free Software Foundation
51 Franklin Street, Fifth Floor
Boston, MA 02110-1301, USA

Copyright © 2006-2025 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with the Invariant Sections being “Funding Free Software”, the Front-Cover texts being (a) (see below), and with the Back-Cover Texts being (b) (see below). A copy of the license is included in the section entitled “GNU Free Documentation License”.

(a) The FSF’s Front-Cover Text is:

A GNU Manual

(b) The FSF’s Back-Cover Text is:

You have freedom to copy and modify this GNU Manual, like GNU software. Copies published by the Free Software Foundation raise funds for GNU development.

Short Contents

1	Enabling OpenMP.....	1
2	OpenMP Implementation Status	3
3	OpenMP Runtime Library Routines	15
4	OpenMP Environment Variables	59
5	Enabling OpenACC.....	71
6	OpenACC Runtime Library Routines	73
7	OpenACC Environment Variables	93
8	CUDA Streams Usage.....	95
9	OpenACC Library Interoperability	97
10	OpenACC Profiling Interface	101
11	OpenMP-Implementation Specifics.....	107
12	Offload-Target Specifics.....	113
13	The libgomp ABI.....	119
14	Reporting Bugs	125
	GNU General Public License	127
	GNU Free Documentation License.....	139
	Funding Free Software	147
	Library Index	149

1 Enabling OpenMP

To activate the OpenMP extensions for C/C++ and Fortran, the compile-time flag `-fopenmp` must be specified. For C and C++, this enables the handling of the OpenMP directives using `#pragma omp` and the `[[omp::directive(...)]], [[omp::sequence(...)]]` and `[[omp::decl(...)]]` attributes. For Fortran, it enables for free source form the `!$omp` sentinel for directives and the `!$` conditional compilation sentinel and for fixed source form the `c$omp`, `*$omp` and `!$omp` sentinels for directives and the `c$`, `*$` and `!$` conditional compilation sentinels. The flag also arranges for automatic linking of the OpenMP runtime library (Chapter 3 [Runtime Library Routines], page 15).

The `-fopenmp-simd` flag can be used to enable a subset of OpenMP directives that do not require the linking of either the OpenMP runtime library or the POSIX threads library.

A complete description of all OpenMP directives may be found in the OpenMP Application Program Interface (<https://www.openmp.org>) manuals. See also Chapter 2 [OpenMP Implementation Status], page 3.

7 OpenACC Environment Variables

The variables `ACC_DEVICE_TYPE` and `ACC_DEVICE_NUM` are defined by section 4 of the OpenACC specification in version 2.0. The variable `ACC_PROFLIB` is defined by section 4 of the OpenACC specification in version 2.6.

7.1 `ACC_DEVICE_TYPE`

Description:

Control the default device type to use when executing compute regions. If unset, the code can be run on any device type, favoring a non-host device type.

Supported values in GCC (if compiled in) are

- `host`
- `nvidia`
- `radeon`

Reference: OpenACC specification v2.6 (<https://www.openacc.org>), section 4.1.

7.2 `ACC_DEVICE_NUM`

Description:

Control which device, identified by device number, is the default device. The value must be a nonnegative integer less than the number of devices. If unset, device number zero is used.

Reference: OpenACC specification v2.6 (<https://www.openacc.org>), section 4.2.

7.3 `ACC_PROFLIB`

Description:

Semicolon-separated list of dynamic libraries that are loaded as profiling libraries. Each library must provide at least the `acc_register_library` routine. Each library file is found as described by the documentation of `dlopen` of your operating system.

See also: Section 6.43 [`acc_register_library`], page 91, Chapter 10 [OpenACC Profiling Interface], page 101,

Reference: OpenACC specification v2.6 (<https://www.openacc.org>), section 4.3.

8 CUDA Streams Usage

This applies to the `nvptx` plugin only.

The library provides elements that perform asynchronous movement of data and asynchronous operation of computing constructs. This asynchronous functionality is implemented by making use of CUDA streams¹.

The primary means by that the asynchronous functionality is accessed is through the use of those OpenACC directives which make use of the `async` and `wait` clauses. When the `async` clause is first used with a directive, it creates a CUDA stream. If an `async-argument` is used with the `async` clause, then the stream is associated with the specified `async-argument`.

Following the creation of an association between a CUDA stream and the `async-argument` of an `async` clause, both the `wait` clause and the `wait` directive can be used. When either the clause or directive is used after stream creation, it creates a rendezvous point whereby execution waits until all operations associated with the `async-argument`, that is, stream, have completed.

Normally, the management of the streams that are created as a result of using the `async` clause, is done without any intervention by the caller. This implies the association between the `async-argument` and the CUDA stream is maintained for the lifetime of the program. However, this association can be changed through the use of the library function `acc_set_cuda_stream`. When the function `acc_set_cuda_stream` is called, the CUDA stream that was originally associated with the `async` clause is destroyed. Caution should be taken when changing the association as subsequent references to the `async-argument` refer to a different CUDA stream.

¹ See "Stream Management" in "CUDA Driver API", TRM-06703-001, Version 5.5, for additional information

9 OpenACC Library Interoperability

9.1 Introduction

The OpenACC library uses the CUDA Driver API, and may interact with programs that use the Runtime library directly, or another library based on the Runtime library, e.g., CUBLAS¹. This chapter describes the use cases and what changes are required in order to use both the OpenACC library and the CUBLAS and Runtime libraries within a program.

9.2 First invocation: NVIDIA CUBLAS library API

In this first use case (see below), a function in the CUBLAS library is called prior to any of the functions in the OpenACC library. More specifically, the function `cublasCreate()`.

When invoked, the function initializes the library and allocates the hardware resources on the host and the device on behalf of the caller. Once the initialization and allocation has completed, a handle is returned to the caller. The OpenACC library also requires initialization and allocation of hardware resources. Since the CUBLAS library has already allocated the hardware resources for the device, all that is left to do is to initialize the OpenACC library and acquire the hardware resources on the host.

Prior to calling the OpenACC function that initializes the library and allocate the host hardware resources, you need to acquire the device number that was allocated during the call to `cublasCreate()`. The invoking of the runtime library function `cudaGetDevice()` accomplishes this. Once acquired, the device number is passed along with the device type as parameters to the OpenACC library function `acc_set_device_num()`.

Once the call to `acc_set_device_num()` has completed, the OpenACC library uses the context that was created during the call to `cublasCreate()`. In other words, both libraries share the same context.

```
/* Create the handle */
s = cublasCreate(&h);
if (s != CUBLAS_STATUS_SUCCESS)
{
    fprintf(stderr, "cublasCreate failed %d\n", s);
    exit(EXIT_FAILURE);
}

/* Get the device number */
e = cudaGetDevice(&dev);
if (e != cudaSuccess)
{
    fprintf(stderr, "cudaGetDevice failed %d\n", e);
    exit(EXIT_FAILURE);
}

/* Initialize OpenACC library and use device 'dev' */
acc_set_device_num(dev, acc_device_nvidia);
```

Use Case 1

¹ See section 2.26, "Interactions with the CUDA Driver API" in "CUDA Runtime API", Version 5.5, and section 2.27, "VDPAU Interoperability", in "CUDA Driver API", TRM-06703-001, Version 5.5, for additional information on library interoperability.


```
        fprintf(stderr, "cublasSaxpy failed %d\n", s);
        exit(EXIT_FAILURE);
    }

    /* Copy the results from the device */
    acc_memcpy_from_device(&h_Y1[0], d_Y, N * sizeof (float));
```

Use Case 2

9.4 OpenACC library and environment variables

There are two environment variables associated with the OpenACC library that may be used to control the device type and device number: `ACC_DEVICE_TYPE` and `ACC_DEVICE_NUM`, respectively. These two environment variables can be used as an alternative to calling `acc_set_device_num()`. As seen in the second use case, the device type and device number were specified using `acc_set_device_num()`. If however, the aforementioned environment variables were set, then the call to `acc_set_device_num()` would not be required.

The use of the environment variables is only relevant when an OpenACC function is called prior to a call to `cudaCreate()`. If `cudaCreate()` is called prior to a call to an OpenACC function, then you must call `acc_set_device_num()`²

² More complete information about `ACC_DEVICE_TYPE` and `ACC_DEVICE_NUM` can be found in sections 4.1 and 4.2 of the OpenACC (<https://www.openacc.org>) Application Programming Interface, Version 2.6.

- `acc_update_device`, `acc_update_device_async`
- `acc_update_self`, `acc_update_self_async`
- `acc_map_data`, `acc_unmap_data`
- `acc_memcpy_to_device`, `acc_memcpy_to_device_async`
- `acc_memcpy_from_device`, `acc_memcpy_from_device_async`

the memory; on Linux, this is in particular the case when the memory placement policy is set to preferred.

- The `access` trait has no effect such that memory is always accessible by all threads. (Except on supported no-host devices.)
- The `sync_hint` trait has no effect.

See also: Chapter 12 [Offload-Target Specifics], page 113,


```

    use data;
    body;
}

setup data;
GOMP_parallel_start (subfunction, &data, num_threads);
subfunction (&data);
GOMP_parallel_end ();

void GOMP_parallel_start (void (*fn)(void *), void *data, unsigned num_threads)

```

The *FN* argument is the subfunction to be run in parallel.

The *DATA* argument is a pointer to a structure used to communicate data in and out of the subfunction, as discussed above with respect to *FIRSTPRIVATE* et al.

The *NUM_THREADS* argument is 1 if an *IF* clause is present and false, or the value of the *NUM_THREADS* clause, if present, or 0.

The function needs to create the appropriate number of threads and/or launch them from the dock. It needs to create the team structure and assign team ids.

```
void GOMP_parallel_end (void)
```

Tears down the team and returns us to the previous `omp_in_parallel()` state.

13.11 Implementing FOR construct

```

#pragma omp parallel for
for (i = lb; i <= ub; i++)
    body;

```

becomes

```

void subfunction (void *data)
{
    long _s0, _e0;
    while (GOMP_loop_static_next (&_s0, &_e0))
    {
        long _e1 = _e0, i;
        for (i = _s0; i < _e1; i++)
            body;
    }
    GOMP_loop_end_nowait ();
}

GOMP_parallel_loop_static (subfunction, NULL, 0, lb, ub+1, 1, 0);
subfunction (NULL);
GOMP_parallel_end ();

#pragma omp for schedule(runtime)
for (i = 0; i < n; i++)
    body;

```

becomes

```

{
    long i, _s0, _e0;
    if (GOMP_loop_runtime_start (0, n, 1, &_s0, &_e0))
        do {
            long _e1 = _e0;
            for (i = _s0, i < _e0; i++)
                body;
        } while (GOMP_loop_runtime_next (&_s0, &_e0));
}

```

```
GOMP_loop_end ();
}
```

Note that while it looks like there is trickiness to propagating a non-constant STEP, there isn't really. We're explicitly allowed to evaluate it as many times as we want, and any variables involved should automatically be handled as PRIVATE or SHARED like any other variables. So the expression should remain evaluable in the subfunction. We can also pull it into a local variable if we like, but since its supposed to remain unchanged, we can also not if we like.

If we have SCHEDULE(STATIC), and no ORDERED, then we ought to be able to get away with no work-sharing context at all, since we can simply perform the arithmetic directly in each thread to divide up the iterations. Which would mean that we wouldn't need to call any of these routines.

There are separate routines for handling loops with an ORDERED clause. Bookkeeping for that is non-trivial...

13.12 Implementing ORDERED construct

```
void GOMP_ordered_start (void)
void GOMP_ordered_end (void)
```

13.13 Implementing SECTIONS construct

A block as

```
#pragma omp sections
{
    #pragma omp section
    stmt1;
    #pragma omp section
    stmt2;
    #pragma omp section
    stmt3;
}
```

becomes

```
for (i = GOMP_sections_start (3); i != 0; i = GOMP_sections_next ())
    switch (i)
    {
        case 1:
            stmt1;
            break;
        case 2:
            stmt2;
            break;
        case 3:
            stmt3;
            break;
    }
GOMP_barrier ();
```

13.14 Implementing SINGLE construct

A block like

```
#pragma omp single
```

```

{
    body;
}

```

becomes

```

if (GOMP_single_start ())
    body;
GOMP_barrier ();

```

while

```

#pragma omp single copyprivate(x)
    body;

```

becomes

```

datap = GOMP_single_copy_start ();
if (datap == NULL)
{
    body;
    data.x = x;
    GOMP_single_copy_end (&data);
}
else
    x = datap->x;
GOMP_barrier ();

```

13.15 Implementing OpenACC's PARALLEL construct

```

void GOACC_parallel ()

```


14 Reporting Bugs

Bugs in the GNU Offloading and Multi Processing Runtime Library should be reported via Bugzilla (<https://gcc.gnu.org/bugzilla/>). Please add "openacc", or "openmp", or both to the keywords field in the bug report, as appropriate.

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

- 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

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 <https://www.gnu.org/copyleft/>.

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

11. RELICENSING

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

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

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

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

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

ADDENDUM: How to use this License for your documents

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

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

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

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

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

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

Funding Free Software

If you want to have more free software a few years from now, it makes sense for you to help encourage people to contribute funds for its development. The most effective approach known is to encourage commercial redistributors to donate.

Users of free software systems can boost the pace of development by encouraging for-a-fee distributors to donate part of their selling price to free software developers—the Free Software Foundation, and others.

The way to convince distributors to do this is to demand it and expect it from them. So when you compare distributors, judge them partly by how much they give to free software development. Show distributors they must compete to be the one who gives the most.

To make this approach work, you must insist on numbers that you can compare, such as, “We will donate ten dollars to the Frobnitz project for each disk sold.” Don’t be satisfied with a vague promise, such as “A portion of the profits are donated,” since it doesn’t give a basis for comparison.

Even a precise fraction “of the profits from this disk” is not very meaningful, since creative accounting and unrelated business decisions can greatly alter what fraction of the sales price counts as profit. If the price you pay is \$50, ten percent of the profit is probably less than a dollar; it might be a few cents, or nothing at all.

Some redistributors do development work themselves. This is useful too; but to keep everyone honest, you need to inquire how much they do, and what kind. Some kinds of development make much more long-term difference than others. For example, maintaining a separate version of a program contributes very little; maintaining the standard version of a program for the whole community contributes much. Easy new ports contribute little, since someone else would surely do them; difficult ports such as adding a new CPU to the GNU Compiler Collection contribute more; major new features or packages contribute the most.

By establishing the idea that supporting further development is “the proper thing to do” when distributing free software for a fee, we can assure a steady flow of resources into making more free software.

Copyright © 1994 Free Software Foundation, Inc.

Verbatim copying and redistribution of this section is permitted without royalty; alteration is not permitted.

Library Index

A

acc_get_property 74
acc_get_property_string 74

E

Environment Variable ... 59, 60, 61, 62, 63, 64, 65,
66, 67, 68, 69

F

FDL, GNU Free Documentation License 139

I

Implementation specific setting.. 63, 66, 68, 69, 107