

GNU libiberty

Phil Edwards et al.

Copyright © 2001-2025 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 no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

1 Using

To date, `libiberty` is generally not installed on its own. It has evolved over years but does not have its own version number nor release schedule.

Possibly the easiest way to use `libiberty` in your projects is to drop the `libiberty` code into your project's sources, and to build the library along with your own sources; the library would then be linked in at the end. This prevents any possible version mismatches with other copies of `libiberty` elsewhere on the system.

Passing `--enable-install-libiberty` to the `configure` script when building `libiberty` causes the header files and archive library to be installed when `make install` is run. This option also takes an (optional) argument to specify the installation location, in the same manner as `--prefix`.

For your own projects, an approach which offers stability and flexibility is to include `libiberty` with your code, but allow the end user to optionally choose to use a previously-installed version instead. In this way the user may choose (for example) to install `libiberty` as part of GCC, and use that version for all software built with that compiler. (This approach has proven useful with software using the GNU `readline` library.)

Making use of `libiberty` code usually requires that you include one or more header files from the `libiberty` distribution. (They will be named as necessary in the function descriptions.) At link time, you will need to add `-liberty` to your link command invocation.

`void obstack_blank_fast (struct obstack *obstack_ptr, size_t size)`
Add *size* uninitialized bytes to a growing object without checking that there is enough room. See Section 2.3.1.7 [Extra Fast Growing], page 8.

`void obstack_1grow_fast (struct obstack *obstack_ptr, char data_char)`
Add one byte containing *data_char* to a growing object without checking that there is enough room. See Section 2.3.1.7 [Extra Fast Growing], page 8.

`size_t obstack_room (struct obstack *obstack_ptr)`
Get the amount of room now available for growing the current object. See Section 2.3.1.7 [Extra Fast Growing], page 8.

`size_t obstack_alignment_mask (struct obstack *obstack_ptr)`
The mask used for aligning the beginning of an object. This is an lvalue. See Section 2.3.1.9 [Obstacks Data Alignment], page 10.

`size_t obstack_chunk_size (struct obstack *obstack_ptr)`
The size for allocating chunks. This is an lvalue. See Section 2.3.1.10 [Obstack Chunks], page 11.

`void *obstack_base (struct obstack *obstack_ptr)`
Tentative starting address of the currently growing object. See Section 2.3.1.8 [Status of an Obstack], page 10.

`void *obstack_next_free (struct obstack *obstack_ptr)`
Address just after the end of the currently growing object. See Section 2.3.1.8 [Status of an Obstack], page 10.

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

A.2 BSD

Copyright © 1990 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. [rescinded 22 July 1999]
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

