

NAME

dwarf_dealloc, **dwarf_fde_cie_list_dealloc**, **dwarf_funcs_dealloc**, **dwarf_globals_dealloc**, **dwarf_pubtypes_dealloc**, **dwarf_ranges_dealloc**, **dwarf_srclines_dealloc**, **dwarf_types_dealloc**, **dwarf_vars_dealloc**, **dwarf_weak_dealloc** - release resources

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

void

dwarf_dealloc(*Dwarf_Debug* dbg, *Dwarf_Ptr* ptr, *Dwarf_Unsigned* type);

dwarf_fde_cie_list_dealloc(*Dwarf_Debug* dbg, *Dwarf_Cie* *cie_list, *Dwarf_Signed* cie_count, *Dwarf_Fde* *fde_list, *Dwarf_Signed* fde_count);

void

dwarf_funcs_dealloc(*Dwarf_Debug* dbg, *Dwarf_Func* *funcs, *Dwarf_Signed* funccount);

void

dwarf_globals_dealloc(*Dwarf_Debug* dbg, *Dwarf_Global* *globals, *Dwarf_Signed* globalcount);

void

dwarf_pubtypes_dealloc(*Dwarf_Debug* dbg, *Dwarf_Type* *pubtypes, *Dwarf_Signed* pubtypecount);

void

dwarf_ranges_dealloc(*Dwarf_Debug* dbg, *Dwarf_Ranges* *ranges, *Dwarf_Signed* rangecount);

void

dwarf_srclines_dealloc(*Dwarf_Debug* dbg, *Dwarf_Line* *lines, *Dwarf_Signed* linecount);

void

dwarf_types_dealloc(*Dwarf_Debug* dbg, *Dwarf_Type* *types, *Dwarf_Signed* typecount);

void

dwarf_vars_dealloc(*Dwarf_Debug* dbg, *Dwarf_Var* *vars, *Dwarf_Signed* varcount);

void

dwarf_weak_dealloc(*Dwarf_Debug* dbg, *Dwarf_Weak* *weak, *Dwarf_Signed* weakcount);

DESCRIPTION

The function **dwarf_dealloc()** is used by applications to indicate that memory areas returned by DWARF Access Library (`libdwarf`, `-ldwarf`) may be safely disposed off. Due to the way memory is managed in the current implementation, the use of **dwarf_dealloc()** is only necessary for a small set of DWARF types.

Argument *dbg* should reference a valid debugging context allocated using `dwarf_init(3)`.

Argument *ptr* should point to an object or memory area obtained by a prior call to a DWARF(3) function.

Argument *type* indicates the type of object being deallocated. The indicated type must match that of the object being passed in argument *ptr*. Valid values for the *type* argument are:

DW_DLA_ABBREV An object of type *Dwarf_Abbrev*, as returned by a call to the function `dwarf_get_abbrev(3)`.

DW_DLA_DIE An object of type *Dwarf_Die*, as returned by calls to the functions `dwarf_child(3)`, `dwarf_offdie(3)` or `dwarf_siblingof(3)`.

DW_DLA_FRAME_BLOCK An array of objects of type *Dwarf_Frame_op*, as returned by a call to the function `dwarf_expand_frame_instructions(3)`.

Calls to **dwarf_dealloc()** with other values for argument *type* are no-ops in this implementation.

The functions **dwarf_fde_cie_list_dealloc()**, **dwarf_funcs_dealloc()**, **dwarf_globals_dealloc()**, **dwarf_pubtypes_dealloc()**, **dwarf_ranges_dealloc()**, **dwarf_srclines_dealloc()**, **dwarf_types_dealloc()**, **dwarf_vars_dealloc()** and **dwarf_weeks_dealloc()** are provided for compatibility with other implementations of the DWARF(3) API. Due to the way memory is managed in the current implementation, these functions are effectively no-ops.

See `dwarf(3)` for more information about the memory management scheme in this implementation of the DWARF(3) API.

RETURN VALUES

Functions **dwarf_dealloc()**, **dwarf_fde_cie_list_dealloc()**, **dwarf_funcs_dealloc()**, **dwarf_globals_dealloc()**, **dwarf_pubtypes_dealloc()**, **dwarf_ranges_dealloc()**, **dwarf_srclines_dealloc()**, **dwarf_types_dealloc()**, **dwarf_vars_dealloc()** and **dwarf_weeks_dealloc()** have no return value.

SEE ALSO

dwarf(3), dwarf_child(3), dwarf_expand_frame_instructions(3), dwarf_get_abbrev(3), dwarf_offdie(3), dwarf_siblingof(3)