NAME

dwarf_get_fde_list - retrieve frame information

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_get_fde_list(*Dwarf_Debug dbg, Dwarf_Cie **cie_list, Dwarf_Signed *cie_count, Dwarf_Fde **fde_list, Dwarf_Signed *fde_count, Dwarf_Error *err*);

int
dwarf_get_fde_list_eh(*Dwarf_Debug dbg, Dwarf_Cie **cie_list, Dwarf_Signed *cie_count, Dwarf_Fde **fde_list, Dwarf_Signed *fde_count, Dwarf_Error *err*);

DESCRIPTION

These functions retrieve frame related information for the specified DWARF debug context.

Function **dwarf_get_fde_list**() retrieves frame information from the DWARF section named ".debug_frame". For objects containing GNU style C++ exception handling information, the function **dwarf_get_fde_list_eh**() retrieves frame information from the section named ".eh_frame".

Frame information is returned using opaque descriptors of type *Dwarf_Cie* and *Dwarf_Fde*. Applications need to use the other frame related functions in the DWARF(3) API set to retrieve the information contained in these descriptors.

Argument *dbg* should reference a DWARF debug context allocated using dwarf_init(3).

Argument *cie_list* should point to a location that will be set to a pointer to an array of *Dwarf_Cie* descriptors.

Argument *cie_count* should point to a location that will be set to the number of *Dwarf_Cie* descriptors returned.

Argument *fde_list* should point to a location that will be set to a pointer to an array of *Dwarf_Fde* descriptors.

Argument *fde_count* should point to a location that will be set to the number of *Dwarf_Fde* descriptors

returned.

If argument *err* is not NULL, it will be used to store error information in case of an error.

Memory Management

The memory areas used for the arrays returned in arguments *cie_list* and *fde_list* are owned by the DWARF Access Library (libdwarf, -ldwarf). Application code should not attempt to directly free these areas. Portable applications should instead use the dwarf_fde_cie_list_dealloc(3) function to indicate that these memory areas may be freed.

RETURN VALUES

On success, these functions returns DW_DLV_OK. They return DW_DLV_NO_ENTRY if there is no frame information associated with the given DWARF debug context. In case of an error, they return DW_DLV_ERROR and set the argument *err*.

EXAMPLES

To obtain frame information from the ".debug_frame" section, use:

```
Dwarf_Debug dbg;
Dwarf_Cie *cie_list, cie;
Dwarf_Fde *fde_list, fde;
Dwarf_Off fde_offset, cie_offset;
Dwarf_Unsigned func_len, fde_length, fde_instlen;
Dwarf_Signed cie_count, fde_count, cie_index;
Dwarf_Addr low_pc;
Dwarf_Ptr fde_addr, fde_inst, cie_inst;
Dwarf_Error de;
int i;

if (dwarf_get_fde_list(dbg, &cie_list, &cie_count,
   &fde_list, &fde_count, &de) != DW_DLV_OK) {
        errx(EXIT_FAILURE, "dwarf_get_fde_list failed: %s",
           dwarf_errmsg(de));
}

for (i = 0; i < fde_count; i++) {
        if (dwarf_get_fde_n(fde_list, i, &fde, &de) != DW_DLV_OK) {
                warnx("dwarf_get_fde_n failed: %s",
                   dwarf_errmsg(de));
                continue;
```

```
                }
                if (dwarf_get_cie_of_fde(fde, &cie, &de) != DW_DLV_OK) {
                        warnx("dwarf_get_fde_n failed: %s",
                           dwarf_errmsg(de));
                        continue;
                }
                if (dwarf_get_fde_range(fde, &low_pc, &func_len, &fde_addr,
                   &fde_length, &cie_offset, &cie_index, &fde_offset,
                   &de) != DW_DLV_OK) {
                        warnx("dwarf_get_fde_range failed: %s",
                           dwarf_errmsg(de));
                        continue;
                }
                if (dwarf_get_fde_instr_bytes(fde, &fde_inst, &fde_instlen,
                   &de) != DW_DLV_OK) {
                        warnx("dwarf_get_fde_instr_bytes failed: %s",
                           dwarf_errmsg(de));
                        continue;
                }

                /* ... Use the retrieved frame information ... */
        }

        /* Indicate that the returned arrays may be freed. */
        dwarf_fde_cie_list_dealloc(dbg, cie_list, cie_count, fde_list,
           fde_count);
```

ERRORS

These functions may fail with the following errors:

[DW_DLE_ARGUMENT] One of the arguments *dbg*, *cie_list*, *cie_count*, *fde_list* or *fde_count* was NULL.

[DW_DLE_NO_ENTRY] There is no frame information associated with the giving DWARF debug context.

SEE ALSO

dwarf(3), dwarf_fde_cie_list_dealloc(3), dwarf_get_cie_index(3), dwarf_get_cie_of_fde(3), dwarf_get_fde_at_pc(3), dwarf_get_fde_instr_bytes(3), dwarf_get_fde_n(3), dwarf_get_fde_range(3), dwarf_set_frame_cfa_value(3), dwarf_set_frame_rule_initial_value(3),

dwarf_set_frame_rule_table_size(3), dwarf_set_frame_same_value(3), dwarf_set_frame_undefined_value(3)