NAME

dwarf_get_abbrev - retrieve abbreviation information

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_get_abbrev(*Dwarf_Debug dbg, Dwarf_Unsigned offset, Dwarf_Abbrev *ret_abbrev, Dwarf_Unsigned *length, Dwarf_Unsigned *attr_count, Dwarf_Error *err*);

DESCRIPTION

Function **dwarf_get_abbrev**() retrieves information about an abbreviation from the DWARF abbreviations section, ".debug_abbrev". Abbreviation information is returned using an opaque descriptor of type *Dwarf_Abbrev*. The returned *Dwarf_Abbrev* descriptor may then be passed to the other abbreviation related APIs in the DWARF(3) API to retrieve specific information about the abbreviation.

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

Argument *offset* should be an offset, relative to the ".debug_abbrev" section, to the start of an abbreviation entry.

Argument *ret_abbrev* should point to a location that will hold a pointer to the returned *Dwarf_Abbrev* descriptor.

Argument *length* should point to a location that will hold the number of bytes used by the abbrevation in the DWARF ".debug_abbrev" section.

Argument *attr_count* should point to a location that will hold the number of attributes in the abbrevation.

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

Memory Management

The memory area used for the *Dwarf_Abbrev* descriptor returned in argument *ret_abbrev* is allocated by the DWARF Access Library (libdwarf, -ldwarf). Application code should use function **dwarf_dealloc**() with the allocation type DW_DLA_ABBREV to free the memory area when the *Dwarf_Abbrev* descriptor is no longer needed.

Application Programming Notes

The last abbreviation entry in a standard DWARF abbreviation section will have a special length value of 1.

RETURN VALUES

Function **dwarf_get_abbrev**() returns DW_DLV_OK when it succeeds. It returns DW_DLV_NO_ENTRY if there is no abbreviation information at offset *offset*. In case of an error, it returns DW_DLV_ERROR and sets the argument *err*.

EXAMPLES

To loop through all the abbreviation information associated with a DWARF debug context, use:

```
Dwarf_Debug dbg;
Dwarf_Abbrev ab;
Dwarf_Off aboff;
Dwarf_Unsigned length, attr_count;
Dwarf_Half tag;
Dwarf_Error de;
int ret;

while ((ret = dwarf_next_cu_header(dbg, NULL, NULL, &aboff,
   NULL, NULL, &de)) ==  DW_DLV_OK) {
        while ((ret = dwarf_get_abbrev(re->dbg, aboff, &ab, &length,
           &attr_count, &de)) == DW_DLV_OK) {
                if (length == 1)       /* Last entry. */
                        break;
                aboff += length;
                if (dwarf_get_abbrev_tag(ab, &tag, &de) != DW_DLV_OK) {
                        warnx("dwarf_get_abbrev_tag failed: %s",
                           dwarf_errmsg(de));
                        continue;
                }
        if (ret != DW_DLV_OK)
                warnx("dwarf_get_abbrev: %s", dwarf_errmsg(de));
}
if (ret == DW_DLV_ERROR)
        warnx("dwarf_next_cu_header: %s", dwarf_errmsg(de));
```

ERRORS

Function **dwarf_get_abbrev**() can fail with:

[DW_DLE_ARGUMENT] One of the arguments *dbg*, *ret_abbrev*, *length* or *attr_count* was NULL.

[DW_DLE_NO_ENTRY] There is no abbreviation information at offset *offset*.

SEE ALSO

dwarf(3), dwarf_dealloc(3), dwarf_get_abbrev_children_flag(3), dwarf_get_abbrev_code(3), dwarf_get_abbrev_entry(3), dwarf_get_abbrev_tag(3)