

**NAME**

**dwarf\_attrlist** - retrieve DWARF attribute descriptors

**LIBRARY**

DWARF Access Library (libdwarf, -ldwarf)

**SYNOPSIS**

```
#include <libdwarf.h>
```

*int*

```
dwarf_attrlist(Dwarf_Die die, Dwarf_Attribute **attrbuf, Dwarf_Signed *attrcount, Dwarf_Error *err);
```

**DESCRIPTION**

Function **dwarf\_attrlist**() retrieves the DWARF attribute descriptors associated with a debugging information entry descriptor in argument *die*. The descriptors are returned as an array of values of the opaque type *Dwarf\_Attribute*. The data associated with each returned attribute descriptor may be queried using the form query functions in the dwarf(3) API set.

Argument *attrbuf* points to a location that will hold a pointer to the returned array of DWARF attribute descriptors. Argument *attrcount* points to a location that will hold the number of descriptors in the returned array.

If argument *err* is non-NULL, it is used to return an error descriptor in case of an error.

**Memory Management**

In the current implementation, the memory allocated for each DWARF attribute descriptor and for the returned array of descriptors is managed by the library and the application does not need to explicitly free the returned pointers. However, for compatibility with other implementations of the dwarf(3) API, the application is permitted to pass the pointers returned by to the **dwarf\_dealloc**() function.

**RETURN VALUES**

Function **dwarf\_attrlist**() returns DW\_DLV\_OK on success.

If the debugging information entry descriptor denoted by argument *die* does not contain any attribute, the function returns DW\_DLV\_NO\_ENTRY and sets argument *err*. For other errors, it returns DW\_DLV\_ERROR and sets argument *err*.

**EXAMPLES**

To retrieve the attribute list for a DWARF debugging information entry use:

```

Dwarf_Die dw_die;
Dwarf_Error dw_e;
Dwarf_Unsigned dw_count;
Dwarf_Attribute *dw_attributes;
int error, i;

```

... variable `dw_die` contains a reference to the DIE of interest ...

```

/* Retrieve the attribute list from the DIE. */
if ((error = dwarf_attrlist(dw_die, &dw_attributes, &dw_count,
    &dw_e)) != DW_DLV_OK)
    errx(EXIT_FAILURE, "dwarf_attrlist: %s", dwarf_errmsg(dw_e));

/* Process the attribute list. */
for (i = 0; i < dw_count; ++i) {
    /* Use the returned pointers in dw_attributes[i] here. */
}

```

## ERRORS

Function `dwarf_diename()` can fail with the following errors:

[DW\_DLE\_ARGUMENT] Arguments *die*, *attrbuf*, or *attrcount* were NULL.

[DW\_DLE\_NO\_ENTRY] Argument *die* had no attributes.

[DW\_DLE\_MEMORY] An out of memory condition was encountered during the execution of the function.

## SEE ALSO

`dwarf(3)`, `dwarf_attr(3)`, `dwarf_dealloc(3)`, `dwarf_hasattr(3)`, `dwarf_hasform(3)`, `dwarf_whatattr(3)`, `dwarf_whatform(3)`