

NAME

dwarf_get_arange_info - extract address range information from a descriptor

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

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

int

```
dwarf_get_arange_info(Dwarf_Arange ar, Dwarf_Addr *start, Dwarf_Unsigned *length,  
    Dwarf_Off *cu_die_offset, Dwarf_Error *err);
```

DESCRIPTION

Function **dwarf_get_arange_info**() extracts address range information from a *Dwarf_Arange* descriptor.

Argument *ar* should reference a valid *Dwarf_Arange* descriptor returned by function **dwarf_get_ranges**(3).

Argument *start* should point to a location which will hold the start value of the address range associated with the descriptor.

Argument *length* should point to a location which will hold the length in bytes of the address range associated with the descriptor.

Argument *cu_die_offset* should point to a location which will be set to an offset, relative to the ".debug_info" section, of the first debugging information entry in the compilation unit associated with argument *ar*.

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

RETURN VALUES

Function **dwarf_get_arange_info**() returns DW_DLV_OK when it succeeds. In case of an error, it returns DW_DLV_ERROR and sets the argument *err*.

EXAMPLES

To loop through all the address lookup table entries, use:

```
Dwarf_Debug dbg;  
Dwarf_Addr start;
```

```

Dwarf_Arange *aranges;
Dwarf_Off die_off;
Dwarf_Signed i, cnt;
Dwarf_Unsigned length;
Dwarf_Error de;

if (dwarf_get_aranges(dbg, &aranges, &cnt, &de) != DW_DLV_OK)
    errx(EXIT_FAILURE, "dwarf_get_aranges: %s",
        dwarf_errmsg(de));
for (i = 0; i < cnt; i++) {
    if (dwarf_get_arange_info(aranges[i], &start, &length,
        &die_off, &de) != DW_DLV_OK) {
        warnx("dwarf_get_arange_info: %s",
            dwarf_errmsg(de));
        continue;
    }
    /* Do something with the returned information. */
}

```

ERRORS

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

[DW_DLE_ARGUMENT] One of the arguments *ar*, *start*, *length* or *cu_die_offset* was NULL.

SEE ALSO

dwarf(3), dwarf_get_arange(3), dwarf_get_arange_cu_header_offset(3), dwarf_get_aranges(3), dwarf_get_cu_die_offset(3)