NAME

dwarf_get_aranges - retrieve program address space mappings

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_get_aranges(*Dwarf_Debug dbg*, *Dwarf_Arange **ar_list*, *Dwarf_Signed *ar_cnt*,
*Dwarf_Error *err*);

DESCRIPTION

The function **dwarf_get_aranges**() retrieves address range information from the ".debug_aranges"
DWARF section. Information about address ranges is returned using opaque descriptors of type
Dwarf_Arange,

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

Argument *ar_list* should point to a location which will be set to a pointer to an array of *Dwarf_Arange*
descriptors.

Argument *ar_cnt* should point to a location which will be set to the number of 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 area used for the array returned in argument *ar_list* is owned by DWARF Access Library
(libdwarf, -ldwarf). Application code should not attempt to directly free this area. Portable applications
should instead use dwarf_dealloc(3) to indicate that the memory area may be freed.

RETURN VALUES

Function **dwarf_get_aranges**() returns DW_DLV_OK when it succeeds. It returns
DW_DLV_NO_ENTRY if there is no ".debug_aranges" section associated with the specified debugging
context. 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_aranges**() can fail with:

[DW_DLE_ARGUMENT]   One of the arguments *dbg*, *ar_list* or *ar_cnt* was NULL.

[DW_DLE_NO_ENTRY]   The debugging context *dbg* did not contain a ".debug_aranges" string
                    section.

SEE ALSO
dwarf(3), dwarf_get_arange(3), dwarf_get_arange_cu_header_offset(3), dwarf_get_arange_info(3),
dwarf_get_cu_die_offset(3)