

NAME

dwarf_add_arange, **dwarf_add_arange_b** - add address range information to a DWARF producer instance

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

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

Dwarf_Unsigned

```
dwarf_add_arange(Dwarf_P_Debug dbg, Dwarf_Addr start, Dwarf_Unsigned length,  
                 Dwarf_Signed symbol_index, Dwarf_Error *err);
```

Dwarf_Unsigned

```
dwarf_add_arange_b(Dwarf_P_Debug dbg, Dwarf_Addr start, Dwarf_Unsigned length,  
                  Dwarf_Unsigned symbol_index, Dwarf_Unsigned end_symbol_index,  
                  Dwarf_Addr offset_from_end_symbol, Dwarf_Error *err);
```

DESCRIPTION

Function **dwarf_add_arange_b()** adds an address range entry to a producer instance.

Argument *dbg* should reference a DWARF producer instance allocated using **dwarf_producer_init(3)** or **dwarf_producer_init_b(3)**.

Argument *start* specifies the relocatable start address of the address range.

Argument *length* specifies the length of the address range.

Argument *symbol_index* specifies the ELF symbol index of the first symbol to be used for relocation.

Argument *end_symbol_index* specifies the ELF symbol index of the second symbol to be used for relocation.

- If argument *end_symbol_index* is not 0, the DW_DLC_SYMBOLIC_RELOCATIONS flag should have been set on the DWARF producer instance. The address value specified by argument *start* will be treated as an offset value from the first symbol, and the argument *offset_from_end_symbol* should hold an offset value from the second symbol. Application code can retrieve the relocation entries for the symbol pair by calling function **dwarf_get_relocation_info(3)**. The relocation entry for the first symbol will have type **dwarf_drt_first_of_length_pair** and the relocation entry for the

second symbol will have type `dwarf_drt_second_of_length_pair`.

- ◆ If argument `end_symbol_index` is 0, argument `offset_from_end_symbol` will be ignored and only one symbol is used for relocation.

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

Function `dwarf_add_arange()` is deprecated. It is similar to function `dwarf_add_arange_b()` except that it cannot handle all possible symbol index values and supports only one relocation symbol.

RETURN VALUES

On success, these functions return a non-zero value. In case of an error, these functions return 0 and set the argument `err`.

ERRORS

These functions can fail with:

[DW_DLE_ARGUMENT] Argument `dbg` was NULL.

[DW_DLE_ARGUMENT] Argument `end_symbol_index` was non-zero, but the flag `DW_DLC_SYMBOLIC_RELOCATIONS` was not set on the producer instance.

[DW_DLE_MEMORY] An out of memory condition was encountered during the execution of the function.

SEE ALSO

`dwarf(3)`, `dwarf_get_relocation_info(3)`, `dwarf_producer_init(3)`, `dwarf_producer_init_b(3)`