NAME

dwarf_add_frame_fde - add a call frame descriptor to a DWARF producer instance

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

Dwarf_Unsigned
dwarf_add_frame_fde(*Dwarf_P_Debug dbg, Dwarf_P_Fde fde, Dwarf_P_Die die, Dwarf_Unsigned cie, Dwarf_Addr virt_addr, Dwarf_Unsigned code_len, Dwarf_Unsigned symbol_index, Dwarf_Error *err*);

Dwarf_Unsigned
dwarf_add_frame_fde_b(*Dwarf_P_Debug dbg, Dwarf_P_Fde fde, Dwarf_P_Die die, Dwarf_Unsigned cie, Dwarf_Addr virt_addr, Dwarf_Unsigned code_len, Dwarf_Unsigned symbol_index, Dwarf_Unsigned end_symbol_index, Dwarf_Addr offset_from_end_sym, Dwarf_Error *err*);

DESCRIPTION

Function **dwarf_add_frame_fde_b**() adds the call frame descriptor referenced by argument *fde* 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 *fde* should reference a frame descriptor allocated using dwarf_new_fde(3).

Argument *die* is ignored by this implementation of the DWARF Access Library (libdwarf, -ldwarf).

Argument *cie* specifies the index of call frame common information entry for the frame descriptor. Valid indices are those returned by the function dwarf_add_frame_cie(3).

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

The meaning of the arguments *virt_addr*, *code_len* and *offset_from_end_sym* depend on the value of argument *end_symbol_index*:

- If the argument *end_symbol_index* is zero, the argument *virt_addr* specifies the relocatable address

of the start of the function associated with the frame descriptor, the argument *code_len* specifies the size in bytes of the machine instructions for this function, the argument *symbol_index* specifies the ELF symbol to be used for relocating the address in argument *virt_addr*, and the argument *offset_from_end_symbol* is ignored.

- If the argument *end_symbol_index* is non-zero, it specifies the ELF symbol index of the second symbol to be used for relocation. In this case, the argument *virt_addr* specifies an offset from the relocatable symbol specified by argument *symbol_index*, the argument *offset_from_end_symbol* should specify an offset from the symbol named by the argument *end_symbol_index*, and the argument *code_len* will be ignored. The DW_DLC_SYMBOLIC_RELOCATIONS flag should also have been set on the DWARF producer instance.

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

Function **dwarf_add_frame_fde**() is similar to function **dwarf_add_frame_fde_b**() except that it supports only one relocation symbol.

RETURN VALUES

On success, these functions return the index value for the added frame descriptor. In case of an error, these functions return DW_DLV_NOCOUNT and set the argument *err*.

ERRORS

These functions can fail with:

[DW_DLE_ARGUMENT]	One of the arguments *dbg* or *fde* was NULL.
[DW_DLE_ARGUMENT]	The frame descriptor referenced by argument *fde* did not belong to the producer instance referenced by argument *dbg*.
[DW_DLE_ARGUMENT]	The common information entry index specified by argument *cie* was invalid.
[DW_DLE_ARGUMENT]	Argument *end_symbol_index* was non-zero, but the flag DW_DLC_SYMBOLIC_RELOCATIONS was not set on the producer instance.

SEE ALSO

dwarf(3), dwarf_add_fde_inst(3), dwarf_add_frame_cie(3), dwarf_fde_cfa_offset(3), dwarf_get_relocation_info(3), dwarf_new_fde(3), dwarf_producer_init(3), dwarf_producer_init_b(3)