NAME

dwarf_get_fde_info_for_all_regs3 - retrieve register rule row

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_get_fde_info_for_all_regs3(*Dwarf_Fde fde, Dwarf_Addr pc, Dwarf_Regtable3 *reg_table, Dwarf_Addr *row_pc, Dwarf_Error *err*);

DESCRIPTION

Function **dwarf_get_fde_info_for_all_regs3**() retrieves a row from the register rule table associated with the given FDE descriptor.

Argument *fde* should reference a valid DWARF FDE descriptor.

Argument *pc* should hold the program counter address to be used to locate the desired table row.

Argument *reg_table* should point to a *Dwarf_Regtable3* descriptor which will hold the returned table row of register rules. The *Dwarf_Regtable3* descriptor is defined in the header file <*libdwarf.h*>:

```
typedef struct {
        Dwarf_Small     dw_offset_relevant;
        Dwarf_Small     dw_value_type;
        Dwarf_Half      dw_regnum;
        Dwarf_Unsigned  dw_offset_or_block_len;
        Dwarf_Ptr       dw_block_ptr;
} Dwarf_Regtable_Entry3;

typedef struct {
        Dwarf_Regtable_Entry3   rt3_cfa_rule;
        Dwarf_Half              rt3_reg_table_size;
        Dwarf_Regtable_Entry3   *rt3_rules;
} Dwarf_Regtable3;
```

The *rt3_reg_table_size* field specifies the maximum number of register rule columns to be returned, and should be set by the application before calling the function. The *rt3_rules* field should point to a

memory arena allocated by the application with space for at least *rt3_reg_table_size* descriptors of type *Dwarf_Regtable_Entry3*.

On a successful execution of this function, the *rt3_cfa_rule* field will be set to the CFA register rule associated with the table row, and the *rt3_rules* array will hold the returned register rules contained in the table row.

For each register rule descriptor returned, the *dw_offset_relevant* field will be set to 1 if the register rule has a offset value, the *dw_value_type* field will be set to the type code of the register rule and the *dw_regnum* field will be set to the register number associated with the register rule. If the register rule is of type DW_EXPR_OFFSET or DW_EXPR_VAL_OFFSET, the *dw_offset_or_block_len* field will be set to the offset value associated with the register rule. If the type is DW_EXPR_EXPRESSION or DW_EXPR_VAL_EXPRESSION, the *dw_offset_or_block_len* field will be set to the length in bytes of the DWARF expression block associated with the register rule. The *dw_block_ptr* field will be set to a pointer to the content of the DWARF expression block associated with the register rule.

Argument *row_pc* should point to a location which will be set to the lowest program counter address associated with the table row.

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

RETURN VALUES

Function **dwarf_get_fde_info_for_all_regs3**() returns DW_DLV_OK when it succeeds. In case of an error, it returns DW_DLV_ERROR and sets the argument *err*.

ERRORS

Function **dwarf_get_fde_info_for_all_regs3**() can fail with:

[DW_DLE_ARGUMENT]	One of the arguments *fde*, *reg_table* or *row_pc* was NULL.
[DW_DLE_PC_NOT_IN_FDE_RANGE]	The program counter value provided in argument *pc* did not fall in the range covered by argument *fde*.

SEE ALSO

dwarf(3), dwarf_get_fde_at_pc(3), dwarf_get_fde_info_for_all_regs(3), dwarf_get_fde_info_for_cfa_reg3(3), dwarf_get_fde_info_for_reg(3), dwarf_get_fde_info_for_reg3(3), dwarf_get_fde_n(3), dwarf_set_frame_cfa_value(3), dwarf_set_frame_rule_initial_value(3), dwarf_set_frame_rule_table_size(3), dwarf_set_frame_same_value(3), dwarf_set_frame_undefined_value(3)