

NAME

dwarf_get_vars, dwarf_var_cu_offset, dwarf_var_die_offset, dwarf_var_name_offsets, dwarf_varname - retrieve information about static variables

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

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

int

```
dwarf_get_vars(Dwarf_Debug dbg, Dwarf_Var **vars, Dwarf_Signed *nvars, Dwarf_Error *err);
```

int

```
dwarf_var_cu_offset(Dwarf_Var var, Dwarf_Off *cu_offset, Dwarf_Error *err);
```

int

```
dwarf_var_die_offset(Dwarf_Var var, Dwarf_Off *die_offset, Dwarf_Error *err);
```

int

```
dwarf_var_name_offsets(Dwarf_Var var, char **name, Dwarf_Off *die_offset,  
    Dwarf_Off *cu_die_offset, Dwarf_Error *err);
```

int

```
dwarf_varname(Dwarf_Var var, char **name, Dwarf_Error *err);
```

DESCRIPTION

These functions retrieve information about the file scope static variables associated with a DWARF debug context. Information about these static variables is returned using opaque descriptors of type *Dwarf_Var*. Applications need to use the functions described below to retrieve the name and offset information contained in these descriptors.

Function **dwarf_get_vars()** retrieves descriptors for all the static variables associated with the DWARF debug context specified by argument *dbg*. The argument *vars* should point to a location that will be set to a pointer to an array of *Dwarf_Var* descriptors. The argument *nvars* should point to a location that will be set to the number of descriptors returned.

Function **dwarf_var_cu_offset()** returns the section-relative offset, relative to the ".debug_info" section, of the compilation unit that contains the debugging information entry associated with the argument *var*. Argument *cu_offset* should point to a location that will hold the returned offset.

Function **dwarf_var_die_offset()** retrieves the section-relative offset, relative to the ".debug_info" section, of the debugging information entry associated with the argument *var*, and stores it into the location pointed to by the argument *die_offset*.

Function **dwarf_var_name_offsets()** retrieves both the name and the associated offsets for the debugging information entry for argument *var*. Argument *name* should point to a location which will be set to a pointer to a NUL-terminated string containing the name of the associated debugging information entry. Argument *die_offset* should point to a location which will be set to a section-relative offset, relative to the ".debug_info" section, of the associated debugging information entry. Argument *cu_die_offset* should point to a location which will be set to a section-relative offset, relative to the ".debug_info" section, of the first debugging information entry in the compilation unit associated with argument *var*.

Function **dwarf_varname()** sets the location pointed to by argument *name* to a pointer to a NUL-terminated string holding the name of the debugging information entry associated with the argument *var*.

Memory Management

The memory area used for the array of *Dwarf_Var* descriptors returned in argument *vars* by function **dwarf_get_vars()** is owned by the DWARF Access Library (libdwarf, -ldwarf). Application code should not attempt to directly free this pointer. Portable code should instead use the function **dwarf_vars_dealloc(3)** to indicate that the memory area may be freed.

The memory area used for the string returned in the *name* argument to functions **dwarf_var_name_offsets()** and **dwarf_varname()** is owned by the DWARF Access Library (libdwarf, -ldwarf). Portable code should indicate that the memory area can be freed using the **dwarf_dealloc(3)** function.

Error Returns

If argument *err* is not NULL, these functions will use it to store error information, in case of an error.

RETURN VALUES

On success, these functions returns DW_DLV_OK. In case of an error, they return DW_DLV_ERROR and set the argument *err*.

ERRORS

These functions may fail with the following errors:

[DW_DLE_ARGUMENT] One of the arguments *cu_die_offset*, *cu_offset*, *dbg*, *die_offset*, *var*, *vars*, *name*, or *nvars* was NULL.

SEE ALSO

dwarf(3), dwarf_dealloc(3), dwarf_get_cu_die_offset_given_cu_header_offset(3),
dwarf_vars_dealloc(3)