NAME

dwarf_get_pubtypes, **dwarf_pubtype_cu_offset**, **dwarf_pubtype_die_offset**, **dwarf_pubtype_name_offsets**, **dwarf_pubtypename** - retrieve information about user-defined types

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

#include <libdwarf.h>

int
dwarf_get_pubtypes(*Dwarf_Debug dbg, Dwarf_Type **types, Dwarf_Signed *ntypes, Dwarf_Error *err*);

int
dwarf_pubtype_cu_offset(*Dwarf_Type type, Dwarf_Off *cu_offset, Dwarf_Error *err*);

int
dwarf_pubtype_die_offset(*Dwarf_Type type, Dwarf_Off *die_offset, Dwarf_Error *err*);

int
dwarf_pubtype_name_offsets(*Dwarf_Type type, char **name, Dwarf_Off *die_offset, Dwarf_Off *cu_die_offset, Dwarf_Error *err*);

int
dwarf_pubtypename(*Dwarf_Type type, char **name, Dwarf_Error *err*);

DESCRIPTION

These functions retrieve information about file-scope, user-defined types recorded in lookup tables in the ".debug_pubtypes" DWARF section. Information about these types is returned using opaque descriptors of type *Dwarf_Type*. Applications need to use the functions described below to retrieve the name and offset information contained in these descriptors.

Function **dwarf_get_pubtypes**() retrieves descriptors for all the user-defined types associated with the DWARF debug context specified by argument *dbg*. The argument *types* should point to a location that will be set to a pointer to an array of *Dwarf_Type* descriptors. The argument *ntypes* should point to a location that will be set to the number of descriptors returned.

Function **dwarf_pubtype_cu_offset**() returns the offset, relative to the ".debug_info" section, of the compilation unit that contains the debugging information entry associated with the argument *type*.

Argument *cu_offset* should point to a location that will hold the returned offset.

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

Function **dwarf_pubtype_name_offsets**() retrieves the name and offsets for the debugging information entry for argument *type*. 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 the 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 the offset, relative to the ".debug_info" section, of the first debugging information entry in the compilation unit associated with argument *type*.

Function **dwarf_pubtypename**() 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 *type*.

Memory Management

The memory area used for the array of *Dwarf_Type* descriptors returned in argument *types* by function **dwarf_get_pubtypes**() 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_types_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_pubtype_name_offsets**() and **dwarf_pubtypename**() 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*.

EXAMPLES

To retrieve the list of file scope user-defined types and print their names, use:

```
Dwarf_Debug dbg;
```

```
Dwarf_Signed ntypes;
Dwarf_Type *types;
Dwarf_Error err;
int n, result;
char *typename;

/* Initialize dbg etc. */;
result = dwarf_get_pubtypes(dbg, &types, &ntypes, &err);
if (result != DW_DLV_OK) /* Handle the error. */
        ;

/* Iterate over the returned array of descriptors. */
for (n = 0; n < ntypes; n++) {
        result = dwarf_pubtypename(types[n], &typename, &err);
        if (result != DW_DLV_OK) /* Handle the error. */
                ;
        printf("%s\n", typename);
}

/* Deallocate the returned array. */
dwarf_types_dealloc(dbg, types, ntypes);
```

ERRORS

These functions may fail with the following errors:

[DW_DLE_ARGUMENT] One of the arguments *cu_die_offset*, *cu_offset*, *dbg*, *die_offset*, *type*, *types*, *name*, or *ntypes* was NULL.

[DW_DLE_NO_ENTRY] The DWARF debugging context referenced by argument *dbg* did not contain information about user-defined types.

SEE ALSO

dwarf(3), dwarf_dealloc(3), dwarf_get_cu_die_offset_given_cu_header_offset(3), dwarf_pubtypes_dealloc(3)