

NAME

dwarf_producer_init, dwarf_producer_init_b - allocate a DWARF producer descriptor

LIBRARY

DWARF Access Library (libdwarf, -ldwarf)

SYNOPSIS

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

Dwarf_P_Debug

```
dwarf_producer_init(Dwarf_Unsigned flags, Dwarf_Callback_Func func, Dwarf_Handler errhand,  
                  Dwarf_Ptr errarg, Dwarf_Error *err);
```

Dwarf_P_Debug

```
dwarf_producer_init_b(Dwarf_Unsigned flags, Dwarf_Callback_Func_b func, Dwarf_Handler errhand,  
                  Dwarf_Ptr errarg, Dwarf_Error *error);
```

DESCRIPTION

These functions allocate and return a *Dwarf_P_Debug* descriptor representing a DWARF producer instance.

The argument *errhand* should contain the address of a function to be called in case of an error. If this argument is NULL, the default error handling scheme is used, see dwarf(3).

The argument *errarg* will be passed to the error handler function when it is invoked.

The argument *err* references a memory location that would hold a *Dwarf_Error* descriptor in case of an error.

The argument *flags* specifies additional characteristics of the DWARF producer instance. The following flags are recognized:

DW_DLC_ISA_IA64	(Deprecated) The target instruction set architecture is IA64. This flag is deprecated. Application code should use the dwarf_producer_set_isa(3) function to specify target instruction set architecture.
-----------------	---

DW_DLC_ISA_MIPS	(Deprecated) The target instruction set architecture is MIPS. This flag is deprecated. Application code should use the dwarf_producer_set_isa(3) function to specify target instruction set architecture.
-----------------	---

DW_DLC_SIZE_32 (Default) The target address size is 32-bit.

DW_DLC_SIZE_64 The target address size is 64-bit.

DW_DLC_STREAM_RELOCATIONS
 (Default) Generate stream relocations.

DW_DLC_SYMBOLIC_RELOCATIONS
 Generate symbolic relocations.

DW_DLC_TARGET_BIGENDIAN
 The target is big endian.

DW_DLC_TARGET_LITTLEENDIAN
 The target is little endian.

DW_DLC_WRITE (Required) Permit writing of DWARF information.

The following flags are mutually exclusive.

- ⊕ Flags DW_DLC_ISA_IA64 and DW_DLC_ISA_MIPS.
- ⊕ Flags DW_DLC_SIZE_32 and DW_DLC_SIZE_64.
- ⊕ Flags DW_DLC_STREAM_RELOCATIONS and DW_DLC_SYMBOLIC_RELOCATIONS.
- ⊕ Flags DW_DLC_TARGET_BIGENDIAN and DW_DLC_TARGET_LITTLEENDIAN.

If neither of the flags DW_DLC_TARGET_BIGENDIAN and DW_DLC_TARGET_LITTLEENDIAN is set, the target's endianness is assumed to be the same as the host's endianness.

Argument *func* should point to an application-provided callback function of type *Dwarf_Callback_Func_b*. The type *Dwarf_Callback_Func_b* is defined in the header file *<libdwarf.h>* as:

```
typedef int (*Dwarf_Callback_Func_b)(char *name, int size,
    Dwarf_Unsigned type, Dwarf_Unsigned flags, Dwarf_Unsigned link,
    Dwarf_Unsigned info, Dwarf_Unsigned *index, int *error);
```

This function is called by the DWARF Access Library (libdwarf, -ldwarf) once for each section in the object file that the library needs to create. The arguments to this callback function specify the values in the ELF section header for the section being created:

name The name of the section being created.
size The *sh_size* value in the section header.

type The *sh_type* value in the section header.
flags The *sh_flags* value in the section header.
link The *sh_link* value in the section header.
info The *sh_info* value in the section header.

On success, the callback function should return the section index value of the created section, and set the location pointed to by argument *index* to the symbol table index of the symbol that associated with the newly created section. This symbol table index will be used in relocation entries referring to the created section.

In case of failure, the callback function should return -1 and set the location pointed to by argument *error* to an application-defined error code. This application returned error code is currently ignored by the library.

Function **dwarf_producer_init()** is deprecated. Function **dwarf_producer_init()** is identical to function **dwarf_producer_init_b()** except that the callback function it expects can not properly handle arbitrary section symbol index values.

Memory Management

The *Dwarf_P_Debug* instance returned by these functions should be freed using the function **dwarf_producer_finish()**.

RETURN VALUES

On success, these functions return the created DWARF producer descriptor. In case of an error, they return DW_DLV_BADADDR and set the argument *err*.

EXAMPLES

To initialize a *Dwarf_P_Debug* instance for a MIPS32 big endian object, use:

```

Dwarf_P_Debug dbg;
Dwarf_Unsigned flags;
Dwarf_Error de;

/* ... assume cb_func points to the callback function ... */

flags = DW_DLC_WRITE | DW_DLC_SIZE_32 | DW_DLC_ISA_MIPS |
        DW_DLC_STREAM_RELOCATIONS | DW_DLC_TARGET_BIGENDIAN;
if ((dbg = dwarf_producer_init(flags, cb_func, NULL, NULL, &de)) ==
    DW_DLV_BADADDR)
    warnx("dwarf_producer_init failed: %s", dwarf_errmsg(-1));

```

ERRORS

These functions can fail with:

[DW_DLE_ARGUMENT] Argument *func* was NULL.

[DW_DLE_ARGUMENT] The flag DW_DLC_WRITE was not set in argument *flags*.

[DW_DLE_ARGUMENT] The flags DW_DLC_SIZE_32 and DW_DLC_SIZE_64 were both set in argument *flags*.

[DW_DLE_ARGUMENT] The flags DW_DLC_ISA_IA64 and DW_DLC_ISA_MIPS were both set in argument *flags*.

[DW_DLE_ARGUMENT] The flags DW_DLC_TARGET_BIGENDIAN and DW_DLC_TARGET_LITTLEENDIAN were both set in argument *flags*.

[DW_DLE_ARGUMENT] The flags DW_DLC_STREAM_RELOCATIONS and DW_DLC_SYMBOLIC_RELOCATIONS were both set in argument *flags*.

[DW_DLE_MEMORY] An out of memory condition was encountered.

SEE ALSO

dwarf(3), dwarf_errmsg(3), dwarf_producer_finish(3), dwarf_producer_set_isa(3), dwarf_transform_to_disk_form(3)