

**Name** elfdump – dumps selected parts of an object file

**Synopsis** elfdump [-cCdegGHiklmnPrsSuvy] [-p | -w file] [-I *index-expr*] [-N *name*] [-T *type*] *filename...*

**Description** The `elfdump` utility symbolically dumps selected parts of the specified object file(s). The options allow specific portions of the file to be displayed.

The `elfdump` utility is similar in function to the `dump(1)` utility. The `dump` utility offers an older and less user-friendly interface than `elfdump`, although `dump` might be more appropriate for certain uses such as in shell scripts.

Archive files, produced by `ar(1)`, can also be inspected with `elfdump`. In this case, each object within the archive is processed using the options supplied.

`elfdump` can display the ELF header, program header array, and section header array for any ELF object. It is also able to display the data found in the following types of sections:

---

| Category                       | Option | ELF Section Type  |
|--------------------------------|--------|---|
| Dynamic                        | -d     | SHT_DYNAMIC   |
| Global Offset Table (GOT)      | -G     | (special, see below)  |
| Group                          | -g     | SHT_GROUP   |
| Hardware/Software Capabilities | -H     | SHT_SUNW_cap  |
| Hash Table                     | -h     | SHT_HASH  |
| Interpreter                    | -i     | (special, see below)  |
| Move                           | -m     | SHT_SUNW_move   |
| Note                           | -n     | SHT_NOTE  |
| Relocation                     | -r     | SHT_RELA<br>SHT_REL   |
| Stack Unwind                   | -u     | SHT_AMD64_UNWIND  |
| Syminfo                        | -y     | SHT_SUNW_syminfo  |
| Symbol Sort                    | -S     | SHT_SUNW_symsort<br>SHT_SUNW_tlssort                            |
| Symbol Table                   | -s     | SHT_SYMTAB<br>SHT_DYNSYM<br>SHT_SUNW_LDYNSYM<br>SHT_SUNW_versym |

---

| Category   | Option | ELF Section Type |
|------------|--------|------------------|
| Versioning | -v     | SHT_SUNW_verdef  |
|            |        | SHT_SUNW_verneed |

Interpreter and global offset table sections do not have a special ELF section type, but are instead implemented as SHT\_PROGBITS sections with well known names (`.interp` and `.got` respectively). `elfdump` is able to recognize and display these special sections.

When run without options to narrow the information displayed, `elfdump` will display all available information for each object.

For a complete description of the displayed information, refer to the *Linker and Libraries Guide*.

**Options** The following options are supported:

- c Dumps section header information.
- C Demangles C++ symbol names.
- d Dumps the contents of the `.dynamic` section.
- e Dumps the ELF header.
- g Dumps the contents of the `.group` section.
- G Dumps the contents of the `.got` section.
- h Dumps the contents of the `.hash` section.
- H Dumps the contents of the `.SUNW_cap` hardware and software capabilities section.
- i Dumps the contents of the `.interp` section.
- I *index-expr* Qualifies the sections or program headers to examine with a specific index or index range. For example, the third section header in a file can be displayed using:

```
example% elfdump -c -I 3 filename
```

An *index-expr* can be a single non-negative integer value that specifies a specific item, as shown in the previous example. Alternatively, an *index-expr* can consist of two such values separated by a colon (:), indicating a range of items. The following example displays the third, fourth, and fifth program headers in a file:

```
example% elfdump -p -I 3:5 filename
```

When specifying an index range, the second value can be omitted to indicate the final item in the file. For example, the following statement lists all section headers from the tenth to the end:

```
example% elfdump -c -I 10: filename
```

See Matching Options for additional information about the matching options (-I, -N, -T).

- k Calculates the ELF checksum. See `gelf_checksum(3ELF)`.
- l Displays long section names without truncation.
- m Dumps the contents of the `.SUNW_move` section.
- n Dumps the contents of the `.note` section.
- N *name* Qualifies the sections or program headers to examine with a specific name. For example, in a file that contains more than one symbol table, the `.dynsym` symbol table can be displayed by itself using:

```
example% elfdump -N .dynsym filename
```

ELF program headers do not have names. If the `-p` option is specified, *name* refers to the program header type, and the behavior of the `-N` option is identical to that of the `-T` option. For example, the program header that identifies an interpreter can be displayed using:

```
example% elfdump -p -N PT_INTERP filename
```

See Matching Options for additional information about the matching options (-I, -N, -T).

- p Dumps the program headers. Individual program headers can be specified using the matching options (-I, -N, -T). See Matching Options for additional information.

The `-p` and `-w` options are mutually exclusive. Only one of these options may be used in a given `elfdump` invocation

- P Generate and use alternative section header information based on the information from the program headers, ignoring any section header information contained in the file. If the file has no section headers a warning message is printed and this option is automatically selected. Section headers are not used by the system to execute a program. As such, a malicious program can have its section headers stripped or altered to provide misleading information. In contrast the program headers must be accurate for the program to be runnable. The use of synthetic section header information derived from the program headers allows files with altered section headers to be examined.

- r Dumps the contents of the `.rel[a]` relocation sections.
- s Dumps the contents of the `.SUNW_ldynsym`, `.dynsym`, and `.symtab` symbol table sections. For archives, the archive symbol table is also dumped. Individual sections can be specified with the matching options (`-I`, `-N`, `-T`). An archive symbol table can be specified using the special section name `-N ARSYM`.

In the case of core files, the `shndx` field will have the value “unknown” since the field does not contain the valid values.

In addition to the standard symbol table information, the version definition index of the symbol is also provided under the *ver* heading.

See Matching Options for additional information about the matching options (`-I`, `-N`, `-T`).

- S Dumps the contents of the `.SUNW_ldynsym` and `.dynsym` symbol table sections sorted in the order given by the `.SUNW_dynsym`sort and `.SUNW_dyntls`sort symbol sort sections. Thread Local Storage (TLS) symbols are sorted by offset. Regular symbols are sorted by address. Symbols not referenced by the sort sections are not displayed.
- T *type* Qualifies the sections or program headers to examine with a specific type. For example, in a file that contains more than one symbol table, the `.dynsym` symbol table can be displayed by itself using:

```
example% elfdump -T SHT_DYNSYM filename
```

The value of *type* can be a numeric value, or any of the `SHT_` symbolic names defined in `/usr/include/sys/elf.h`. The `SHT_` prefix is optional, and *type* is case insensitive. Therefore, the above example can also be written as:

```
example% elfdump -T dynsym filename
```

If the `-p` option is specified, *type* refers to the program header type, which allows for the display of specific program headers. For example, the program header that identifies an interpreter can be displayed using:

```
example% elfdump -p -T PT_INTERP filename
```

The value of *type* can be a numeric value, or any of the `PT_` symbolic names defined in `/usr/include/sys/elf.h`. The `PT_` prefix is optional, and *type* is case insensitive. Therefore, the above example can also be written as:

```
example% elfdump -p -T interp filename
```

See Matching Options for additional information about the matching options (`-I`, `-N`, `-T`).

- u Dumps the contents of a frame unwind.
- v Dumps the contents of the `.SUNW_version` version sections.
- w *file* Writes the contents of sections which are specified with the matching options (-I, -N, -T) to the named file. For example, extracting the `.text` section of a file can be carried out with:  

```
example% elfdump -w text.out -N .text filename
```

See Matching Options for additional information about the matching options (-I, -N, -T).

The -p and -w options are mutually exclusive. Only one of these options may be used in a given `elfdump` invocation
- y Dumps the contents of the `.SUNW_syminfo` section.

**Operands** The following operand is supported:

*filename* The name of the specified object file.

### Usage

**Matching Options** The options -I, -N, and -T are collectively referred to as the *matching options*. These options are used to narrow the range of program headers or sections to examine, by index, name, or type.

The exact interpretation of the matching options depends on the other options used:

- When used with the -p option, the matching options reference program headers. -I refers to program header indexes. -T refers to program header types. As program headers do not have names, the -N option behaves identically to -T for program headers.
- The matching options are used to select sections by index, name, or type when used with any of the options -c, -g, -m, -n, -r, -s, -S, -u, or -w.
- If matching options are used alone without any of the options -c, -g, -m, -n, -p, -r, -s, -S, -u, or -w, then `elfdump` examines each object, and displays the contents of any sections matched.

Any number and type of matching option can be mixed in a given invocation of `elfdump`. In this case, `elfdump` will display the superset of all items matched by any of the matching options used. This feature allows for the selection of complex groupings of items using the most convenient form for specifying each item.

**Files** `liblddbg.so` linker debugging library

**Attributes** See `attributes(5)` for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | SUNWbtool       |
| Interface Stability | Stable          |

**See Also** `ar(1)`, `dump(1)`, `nm(1)`, `pvs(1)`, `elf(3ELF)`, `attributes(5)`

*Linker and Libraries Guide*

