NAME

readelf − display information about ELF files

SYNOPSIS

readelf [**−a**|**−−all**]
[**−h**|**−−file−header**]
[**−l**|**−−program−headers**|**−−segments**]
[**−S**|**−−section−headers**|**−−sections**]
[**−g**|**−−section−groups**]
[**−t**|**−−section−details**]
[**−e**|**−−headers**]
[**−s**|**−−syms**|**−−symbols**]
[**−−dyn−syms**]
[**−n**|**−−notes**]
[**−r**|**−−relocs**]
[**−u**|**−−unwind**]
[**−d**|**−−dynamic**]
[**−V**|**−−version−info**]
[**−A**|**−−arch−specific**]
[**−D**|**−−use−dynamic**]
[**−x** <number or name>|**−−hex−dump=**<number or name>]
[**−p** <number or name>|**−−string−dump=**<number or name>]
[**−R** <number or name>|**−−relocated−dump=**<number or name>]
[**−z**|**−−decompress**]
[**−c**|**−−archive−index**]
[**−w[lLiaprmfFsoRtUuTgAckK]**|
−−debug−dump[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=frames−interp,=str,=loc,
[**−−dwarf−depth=***n*]
[**−−dwarf−start=***n*]
[**−−ctf=***section*]
[**−−ctf−parent=***section*]
[**−−ctf−symbols=***section*]
[**−−ctf−strings=***section*]
[**−I**|**−−histogram**]
[**−v**|**−−version**]
[**−W**|**−−wide**]
[**−H**|**−−help**]
elffile...

DESCRIPTION

readelf displays information about one or more ELF format object files. The options control what particular information to display.

elffile... are the object files to be examined. 32−bit and 64−bit ELF files are supported, as are archives containing ELF files.

This program performs a similar function to **objdump** but it goes into more detail and it exists independently of the BFD library, so if there is a bug in BFD then readelf will not be affected.

OPTIONS

The long and short forms of options, shown here as alternatives, are equivalent. At least one option besides **−v** or **−H** must be given.

−a
−−all
Equivalent to specifying **−−file−header**, **−−program−headers**, **−−sections**, **−−symbols**, **−−relocs**, **−−dynamic**, **−−notes**, **−−version−info**, **−−arch−specific**, **−−unwind**, **−−section−groups** and **−−histogram**.

Note – this option does not enable **--use-dynamic** itself, so if that option is not present on the command line then dynamic symbols and dynamic relocs will not be displayed.

-h
--file-header
Displays the information contained in the ELF header at the start of the file.

-l
--program-headers
--segments
Displays the information contained in the file's segment headers, if it has any.

-S
--sections
--section-headers
Displays the information contained in the file's section headers, if it has any.

-g
--section-groups
Displays the information contained in the file's section groups, if it has any.

-t
--section-details
Displays the detailed section information. Implies **-S**.

-s
--symbols
--syms
Displays the entries in symbol table section of the file, if it has one. If a symbol has version information associated with it then this is displayed as well. The version string is displayed as a suffix to the symbol name, preceeded by an @ character. For example **foo@VER_1**. If the version is the default version to be used when resolving unversioned references to the symbol then it is displayed as a suffix preceeded by two @ characters. For example **foo@@VER_2**.

--dyn-syms
Displays the entries in dynamic symbol table section of the file, if it has one. The output format is the same as the format used by the **--syms** option.

-e
--headers
Display all the headers in the file. Equivalent to **-h -l -S**.

-n
--notes
Displays the contents of the NOTE segments and/or sections, if any.

-r
--relocs
Displays the contents of the file's relocation section, if it has one.

-u
--unwind
Displays the contents of the file's unwind section, if it has one. Only the unwind sections for IA64 ELF files, as well as ARM unwind tables (`.ARM.exidx` / `.ARM.extab`) are currently supported. If support is not yet implemented for your architecture you could try dumping the contents of the *.eh_frames* section using the **--debug-dump=frames** or **--debug-dump=frames-interp** options.

-d
--dynamic
Displays the contents of the file's dynamic section, if it has one.

−V
−−version−info
Displays the contents of the version sections in the file, it they exist.

−A
−−arch−specific
Displays architecture-specific information in the file, if there is any.

−D
−−use−dynamic
When displaying symbols, this option makes **readelf** use the symbol hash tables in the file's dynamic section, rather than the symbol table sections.

When displaying relocations, this option makes **readelf** display the dynamic relocations rather than the static relocations.

−x <number or name>
−−hex−dump=<number or name>
Displays the contents of the indicated section as a hexadecimal bytes. A number identifies a particular section by index in the section table; any other string identifies all sections with that name in the object file.

−R <number or name>
−−relocated−dump=<number or name>
Displays the contents of the indicated section as a hexadecimal bytes. A number identifies a particular section by index in the section table; any other string identifies all sections with that name in the object file. The contents of the section will be relocated before they are displayed.

−p <number or name>
−−string−dump=<number or name>
Displays the contents of the indicated section as printable strings. A number identifies a particular section by index in the section table; any other string identifies all sections with that name in the object file.

−z
−−decompress
Requests that the section(s) being dumped by **x**, **R** or **p** options are decompressed before being displayed. If the section(s) are not compressed then they are displayed as is.

−c
−−archive−index
Displays the file symbol index information contained in the header part of binary archives. Performs the same function as the **t** command to **ar**, but without using the BFD library.

−w[lLiaprmfFsoRtUuTgAckK]
−−debug−dump[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=frames−interp,=str,=loc
Displays the contents of the DWARF debug sections in the file, if any are present. Compressed debug sections are automatically decompressed (temporarily) before they are displayed. If one or more of the optional letters or words follows the switch then only those type(s) of data will be dumped. The letters and words refer to the following information:

`a`
`=abbrev`
Displays the contents of the **.debug_abbrev** section.

`A`
`=addr`
Displays the contents of the **.debug_addr** section.

`c`

=cu_index
Displays the contents of the **.debug_cu_index** and/or **.debug_tu_index** sections.

f
=frames
Display the raw contents of a **.debug_frame** section.

F
=frame-interp
Display the interpreted contents of a **.debug_frame** section.

g
=gdb_index
Displays the contents of the **.gdb_index** and/or **.debug_names** sections.

i
=info
Displays the contents of the **.debug_info** section. Note: the output from this option can also be restricted by the use of the **--dwarf-depth** and **--dwarf-start** options.

k
=links
Displays the contents of the **.gnu_debuglink** and/or **.gnu_debugaltlink** sections. Also displays any links to separate dwarf object files (dwo), if they are specified by the DW_AT_GNU_dwo_name or DW_AT_dwo_name attributes in the **.debug_info** section.

K
=follow-links
Display the contents of any selected debug sections that are found in linked, separate debug info file(s). This can result in multiple versions of the same debug section being displayed if it exists in more than one file.

In addition, when displaying DWARF attributes, if a form is found that references the separate debug info file, then the referenced contents will also be displayed.

l
=rawline
Displays the contents of the **.debug_line** section in a raw format.

L
=decodedline
Displays the interpreted contents of the **.debug_line** section.

m
=macro
Displays the contents of the **.debug_macro** and/or **.debug_macinfo** sections.

o
=loc
Displays the contents of the **.debug_loc** and/or **.debug_loclists** sections.

p
=pubnames
Displays the contents of the **.debug_pubnames** and/or **.debug_gnu_pubnames** sections.

r
=aranges
Displays the contents of the **.debug_aranges** section.

R
=Ranges
Displays the contents of the **.debug_ranges** and/or **.debug_rnglists** sections.

`s`
`=str`
Displays the contents of the **.debug_str**, **.debug_line_str** and/or **.debug_str_offsets** sections.

`t`
`=pubtype`
Displays the contents of the **.debug_pubtypes** and/or **.debug_gnu_pubtypes** sections.

`T`
`=trace_aranges`
Displays the contents of the **.trace_aranges** section.

`u`
`=trace_abbrev`
Displays the contents of the **.trace_abbrev** section.

`U`
`=trace_info`
Displays the contents of the **.trace_info** section.

Note: displaying the contents of **.debug_static_funcs**, **.debug_static_vars** and **debug_weaknames** sections is not currently supported.

−−dwarf−depth=*n*
Limit the dump of the `.debug_info` section to *n* children. This is only useful with **−−debug−dump=info**. The default is to print all DIEs; the special value 0 for *n* will also have this effect.

With a non-zero value for *n*, DIEs at or deeper than *n* levels will not be printed. The range for *n* is zero-based.

−−dwarf−start=*n*
Print only DIEs beginning with the DIE numbered *n*. This is only useful with **−−debug−dump=info**.

If specified, this option will suppress printing of any header information and all DIEs before the DIE numbered *n*. Only siblings and children of the specified DIE will be printed.

This can be used in conjunction with **−−dwarf−depth**.

−−ctf=*section*
Display the contents of the specified CTF section. CTF sections themselves contain many subsections, all of which are displayed in order.

−−ctf−parent=*section*
Specify the name of another section from which the CTF dictionary can inherit types. (If none is specified, we assume the CTF dictionary inherits types from the default-named member of the archive contained within this section.)

−−ctf−symbols=*section*
−−ctf−strings=*section*
Specify the name of another section from which the CTF file can inherit strings and symbols. By default, the `.symtab` and its linked string table are used.

If either of **−−ctf−symbols** or **−−ctf−strings** is specified, the other must be specified as well.

−I
−−histogram
Display a histogram of bucket list lengths when displaying the contents of the symbol tables.

−v
−−version
Display the version number of readelf.

−W
−−wide
Don't break output lines to fit into 80 columns. By default **readelf** breaks section header and segment listing lines for 64−bit ELF files, so that they fit into 80 columns. This option causes **readelf** to print each section header resp. each segment one a single line, which is far more readable on terminals wider than 80 columns.

−H
−−help
Display the command-line options understood by **readelf**.

@*file*
Read command-line options from *file*. The options read are inserted in place of the original @*file* option. If *file* does not exist, or cannot be read, then the option will be treated literally, and not removed.

Options in *file* are separated by whitespace. A whitespace character may be included in an option by surrounding the entire option in either single or double quotes. Any character (including a backslash) may be included by prefixing the character to be included with a backslash. The *file* may itself contain additional @*file* options; any such options will be processed recursively.

SEE ALSO

objdump (1), and the Info entries for *binutils*.

COPYRIGHT