NAME

readelf - Displays 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[lLiaprmfFsoRt]**| **--debug-**
dump[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=frames-
interp,=str,=loc,=Ranges,=pubtypes,=trace_info,=trace_abbrev,=trace_aranges,=gdb_index]]   [**--dwarf-**
depth=*n*] [**--dwarf-start**=*n*] [**-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** and **--version-info**.

-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 preceded 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.

-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.

-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[lLiaprmfFsoRt]
--debug-dump[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=frames-interp,=str,=loc,=Ranges,=pubtypes,=trace_info,=trace_abbrev,=trace_aranges,=gdb_index]

Displays the contents of the debug sections in the file, if any are present. If one of the optional letters or words follows the switch then only data found in those specific sections will be dumped.

Note that there is no single letter option to display the content of trace sections or .gdb_index.

Note: the **=decodedline** option will display the interpreted contents of a .debug_line section whereas the **=rawline** option dumps the contents in a raw format.

Note: the **=frames-interp** option will display the interpreted contents of a .debug_frame section whereas the **=frames** option dumps the contents in a raw format.

Note: the output from the **=info** option can also be affected by the options **--dwarf-depth** and **--dwarf-start**.

--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**.

-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

Copyright (c) 1991-2017 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled ''GNU Free Documentation License''.