

NAME

objcopy - copy and translate object files

SYNOPSIS

```
objcopy [-I objformat | -s objformat | --input-target=objformat]
        [-K symbolname | --keep-symbol=symbolname]
        [-L symbolname | --localize-symbol=symbolname]
        [-N symbolname | --strip-symbol=symbolname] [-O objformat | --output-target=objformat]
        [-R sectionname | --remove-section=sectionname] [-S | --strip-all] [-V | --version]
        [-W symbolname | --weaken-symbol=symbolname] [-X | --discard-locals]
        [-d | -g | --strip-debug] [-h | --help] [-j sectionname | --only-section=sectionname]
        [-p | --preserve-dates] [-w | --wildcard] [-x | --discard-all] [--add-gnu-debuglink=filename]
        [--add-section sectionname=filename] [--adjust-section-vma section{+|-|=}val |
        --change-section-address section{+|-|=}val] [--adjust-start=increment |
        --change-start=increment] [--adjust-vma=increment | --change-addresses=increment]
        [--adjust-warnings | --change-warnings] [--change-section-lma section{+|-|=}val]
        [--change-section-vma section{+|-|=}val] [--extract-dwo] [--gap-fill=val]
        [--globalize-symbol=ar symbolname] [--globalize-symbols=filename] [--localize-hidden]
        [--localize-symbols=filename] [--no-adjust-warnings | --no-change-warnings]
        [--only-keep-debug] [--pad-to=address] [--prefix-alloc-sections=string]
        [--prefix-sections=string] [--prefix-symbols=string]
        [--rename-section oldname=newname[flags]] [--set-section-flags sectionname=flags]
        [--set-start=address] [--srec-forceS3] [--srec-len=val] [--strip-dwo] [--strip-symbols=filename]
        [--strip-unneeded] infile [outfile]
```

DESCRIPTION

The **objcopy** utility copies the content of the binary object named by argument *infile* to that named by argument *outfile*, transforming it according to the command line options specified. If argument *outfile* is not specified, **objcopy** will create a temporary file and will subsequently rename it as *infile*.

The **objcopy** utility supports the following options:

-I *objformat* | **-s** *objformat* | --**input-target**=*objformat*

Specify that the input file named by the argument *infile* is in the object format specified by the argument *objformat*.

-K *symbolname* | --**keep-symbol**=*symbolname*

Copy the symbol named by argument *symbolname* to the output.

-L *symbolname* | --**localize-symbol**=*symbolname*

Make the symbol named by argument *symbolname* local to the output file.

-N *symbol* | **--strip-symbol=***symbolname*

Do not copy the symbol named by argument *symbolname* to the output.

-O *objformat* | **--output-target=***objformat*

Write the output file using the object format specified in argument *objformat*. The argument *objformat* should be one of the target names recognized by `elftc_bfd_find_target(3)`.

-R *sectionname* | **--remove-section=***sectionname*

Remove any section with name *sectionname* from the output file.

-S | **--strip-all**

Do not copy symbol and relocation information to the target file.

-V | **--version**

Print a version identifier and exit.

-W *symbolname* | **--weaken-symbol=***symbolname*

Mark the symbol named by argument *symbolname* as weak in the output.

-X | **--discard-locals**

Do not copy compiler generated local symbols to the output.

-d | **-g** | **--strip-debug**

Do not copy debugging information to the target file.

-h | **--help**

Display a help message and exit.

-j *sectionname* | **--only-section=***sectionname*

Copy only the section named by argument *sectionname* to the output.

-p | **--preserve-dates**

Set the access and modification times of the output file to the same as those of the input.

-w | **--wildcard**

Use shell-style patterns to name symbols. The following meta-characters are recognized in patterns:

! If this is the first character of the pattern, invert the sense of the pattern match.

- * Matches any string of characters in a symbol name.
- ? Matches zero or one character in a symbol name.
- [Mark the start of a character class.
- \ Remove the special meaning of the next character in the pattern.
-] Mark the end of a character class.

-x | --discard-all

Do not copy non-global symbols to the output.

--add-gnu-debuglink=*filename*

Create a `.gnu_debuglink` section in the output file that references the debug data in *filename*.

--add-section *sectionname*=*filename*

Add a new section to the output file with name *sectionname*. The contents of the section are taken from the file named by argument *filename*. The size of the section will be the number of bytes in file *filename*.

--adjust-section-vma *section*{+|-|=}*val* | --change-section-address *section*{+|-|=}*val*

Depending on the operator specified, increase, decrease or set both the virtual memory address and the load memory address of the section named by the argument *section*. The argument *val* specifies the desired increment, decrement or new value for the address.

--adjust-start=*increment* | --change-start=*increment*

Increase the entry point address of the output ELF object by the value specified in the argument *increment*.

--adjust-vma=*increment* | --change-addresses=*increment*

Increase the virtual memory address and the load memory address of all sections by the value specified by the argument *increment*.

--adjust-warnings | --change-warnings

Issue a warning if the section specified by the options **--change-section-address**, **--change-section-lma** or **--change-section-vma** does not exist in the input object. This is the default.

--change-section-lma *section*{+|-|=}*val*

Change or set the load memory address of the section named by the argument *section*. Depending on the operator specified, the value in argument *val* will be used as an increment, a decrement or as the new value of the load memory address.

--change-section-vma *section*{+|-|=}*val*

Change or set the virtual memory address of the section named by the argument *section*. Depending on the operator specified, the value in argument *val* will be used as an increment, a decrement or as the new value of the virtual memory address.

--extract-dwo

Copy only .dwo debug sections to the output file.

--gap-fill=*val*

Fill the gaps between sections with the byte value specified by the argument *val*.

--globalize-symbol=*symbolname*

Make the symbol named by argument *symbolname* global, so that it is visible outside of the file in which it is defined.

--globalize-symbols=*filename*

Make each symbol listed in the file specified by *filename* global.

--localize-hidden

Make all hidden symbols local to the output file. This includes symbols with internal visibility.

--localize-symbols=*filename*

Make each symbol listed in the file specified by *filename* local to the output file.

--no-adjust-warnings | **--no-change-warnings**

Do not issue a warning if the section specified by the options **--change-section-address**, **--change-section-lma** or **--change-section-vma** is missing in the input object.

--only-keep-debug

Copy only debugging information to the output file.

--pad-to=*address*

Pad the load memory address of the output object to the value specified by the argument *address* by increasing the size of the section with the highest load memory address.

--prefix-alloc-sections=*string*

Prefix the section names of all the allocated sections with *string*.

--prefix-sections=*string*

Prefix the section names of all the sections with *string*.

--prefix-symbols=*string*

Prefix the symbol names of all the symbols with *string*.

--rename-section *oldname=newname*[*flags*]

Rename the section named by argument *oldname* to *newname*, optionally changing the sections flags to that specified by argument *flags*. Allowed values for the argument *flags* are as for option **--set-section-flags** below.

--set-section-flags *sectionname=flags*

Set the flags for the section named by argument *sectionname* to those specified by argument *flags*. Argument *flags* is a comma separated list of the following flag names:

alloc The section occupies space in the output file.
code The section contains machine instructions.
contents This flag is accepted but is ignored.
data The section contains writeable data.
debug The section holds debugging information.
load The section is loadable.
noload The section should not be loaded into memory.
readonly The section is not writable.
rom The section contains ROM'able contents.
share This flag is accepted but is ignored.

--set-start=*address*

Set the start address of the output ELF object to the value specified by the argument *address*.

--srec-forceS3

Only generate S-records of type "S3". This option is only meaningful when the output target is set to "srec".

--srec-len=*val*

Set the maximum length of an S-record line to *val*. This option is only meaningful when the output target is set to "srec".

--strip-dwo

Do not copy .dwo debug sections to the output file.

--strip-symbols=*filename*

Do not copy any of the symbols specified by *filename* to the output.

--strip-unneeded

Do not copy symbols that are not needed for relocation processing.

DIAGNOSTICS

The **objcopy** utility exits 0 on success, and >0 if an error occurs.

SEE ALSO

ar(1), ld(1), mcs(1), strip(1), elf(3), elftc_bfd_find_target(3), ar(5), elf(5)

COMPATIBILITY

The **objcopy** utility is expected to be option compatible with GNU **objcopy**.

HISTORY

objcopy has been implemented by Kai Wang <kaiwang27@users.sourceforge.net>.