



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'find-debuginfo.1'***

**\$ man find-debuginfo.1**

FIND-DEBUGINFO(1) User Commands FIND-DEBUGINFO(1)

#### NAME

find-debuginfo - finds debuginfo and processes it

#### SYNOPSIS

find-debuginfo [OPTION]... [builddir]

#### DESCRIPTION

automagically generates debug info and file lists

#### OPTIONS

[--strict-build-id] [-g] [-r] [-m] [-i] [-n] [--keep-section SECTION] [--remove-section SECTION] [--g-libs] [-j N] [--jobs N] [-o debugfiles.list] [-S debugsourcefiles.list] [--run-dwz] [--dwz-low-mem-die-limit N] [--dwz-max-die-limit N] [--dwz-single-file-mode] [--build-id-seed SEED] [--unique-debug-suffix SUFFIX] [--unique-debug-src-base BASE] [[-l filelist]... [-p 'pattern']] -o debuginfo.list] [builddir]

The -g flag says to use strip -g instead of full strip on DSOs or EXEs. The --g-libs flag says to use strip -g instead of full strip ONLY on DSOs. Options -g and --g-libs are mutually exclusive.

The -r flag says to use eu-strip --reloc-debug-sections.

Use --keep-section SECTION or --remove-section SECTION to explicitly keep a (non-allocated) section in the main executable or explicitly remove it into the .debug file. SECTION is an extended wildcard pattern. Both options can be given more than once.

The --strict-build-id flag says to exit with failure status if any ELF binary processed fails to contain a build-id note.

The -m flag says to include a .gnu\_debugdata section in the main binary.

The `-i` flag says to include a `.gdb_index` section in the `.debug` file.

The `-n` flag says to not recompute the build-id.

The `-j`, `--jobs N` option will spawn `N` processes to do the debuginfo extraction in parallel.

A single `-o` switch before any `-l` or `-p` switches simply renames the primary output file from `debugfiles.list` to something else. A `-o` switch that follows a `-p` switch or some `-l` switches produces an additional output file with the debuginfo for the files in the `-l` filelist file, or whose names match the `-p` pattern. The `-p` argument is an `grep -E` -style regexp matching the a file name, and must not use anchors (`^` or `$`).

The `--run-dwz` flag instructs `find-debuginfo` to run the `dwz` utility if available, and `--dwz-low-mem-die-limit` and `--dwz-max-die-limit` provide detailed limits. See `dwz(1)` -l and `-L` option for details. Use `--dwz-single-file-mode` to disable multi-file mode, see `dwz(1)` -m for more details.

If `--build-id-seed SEED` is given then `debugedit` is called to update the build-ids it finds adding the `SEED` as seed to recalculate the build-id hash. This makes sure the build-ids in the ELF files are unique between versions and releases of the same package. (Use `--build-id-seed "%{VERSION}-%{RELEASE}"`.)

If `--unique-debug-suffix SUFFIX` is given then the debug files created for `<FILE>` will be named `<FILE>-<SUFFIX>.debug`. This makes sure `.debug` are unique between package version, release and architecture. (Use `--unique-debug-suffix "-%{VERSION}-%{RELEASE}.%{_arch}"`.)

If `--unique-debug-src-base BASE` is given then the source directory will be called `/usr/de?bug/src/<BASE>`. This makes sure the debug source dirs are unique between package version, release and achitecture (Use `--unique-debug-src-base "%{name}-%{VERSION}-%{RELEASE}.%{_arch}"`)

All file names in switches are relative to `builddir` (`'.'` if not given).