

**NAME**

**archive\_read\_set\_filter\_option**, **archive\_read\_set\_format\_option**, **archive\_read\_set\_option**, **archive\_read\_set\_options** — functions controlling options for reading archives

**LIBRARY**

Streaming Archive Library (libarchive, -larchive)

**SYNOPSIS**

```

int
archive_read_set_filter_option(struct archive *, const char *module,
    const char *option, const char *value);

int
archive_read_set_format_option(struct archive *, const char *module,
    const char *option, const char *value);

int
archive_read_set_option(struct archive *, const char *module,
    const char *option, const char *value);

int
archive_read_set_options(struct archive *, const char *options);

```

**DESCRIPTION**

These functions provide a way for libarchive clients to configure specific read modules.

**archive\_read\_set\_filter\_option()**, **archive\_read\_set\_format\_option()**

Specifies an option that will be passed to currently-registered filters (including decompression filters) or format readers.

If *option* and *value* are both NULL, these functions will do nothing and **ARCHIVE\_OK** will be returned. If *option* is NULL but *value* is not, these functions will do nothing and **ARCHIVE\_FAILED** will be returned.

If *module* is not NULL, *option* and *value* will be provided to the filter or reader named *module*. The return value will be that of the module. If there is no such module, **ARCHIVE\_FAILED** will be returned.

If *module* is NULL, *option* and *value* will be provided to every registered module. If any module returns **ARCHIVE\_FATAL**, this value will be returned immediately. Otherwise, **ARCHIVE\_OK** will be returned if any module accepts the option, and **ARCHIVE\_FAILED** in all other cases.

**archive\_read\_set\_option()**

Calls **archive\_read\_set\_format\_option()**, then **archive\_read\_set\_filter\_option()**. If either function returns **ARCHIVE\_FATAL**, **ARCHIVE\_FATAL** will be returned immediately. Otherwise, greater of the two values will be returned.

**archive\_read\_set\_options()**

*options* is a comma-separated list of options. If *options* is NULL or empty, **ARCHIVE\_OK** will be returned immediately.

Calls **archive\_read\_set\_option()** with each option in turn. If any **archive\_read\_set\_option()** call returns **ARCHIVE\_FATAL**, **ARCHIVE\_FATAL** will be returned immediately.

Individual options have one of the following forms:

*option=value*

The option/value pair will be provided to every module. Modules that do not accept an option with this name will ignore it.

*option* The option will be provided to every module with a value of “1”.

*!option*

The option will be provided to every module with a NULL value.

*module:option=value, module:option, module:!option*

As above, but the corresponding option and value will be provided only to modules whose name matches *module*.

## OPTIONS

Format iso9660

**joliet** Support Joliet extensions. Defaults to enabled, use **!joliet** to disable.

**rockridge**

Support RockRidge extensions. Defaults to enabled, use **!rockridge** to disable.

Format tar

**compat-2x**

Libarchive 2.x incorrectly encoded Unicode filenames on some platforms. This option mimics the libarchive 2.x filename handling so that such archives can be read correctly.

**hdrcharset**

The value is used as a character set name that will be used when translating filenames.

**mac-ext**

Support Mac OS metadata extension that records data in special files beginning with a period and underscore. Defaults to enabled on Mac OS, disabled on other platforms. Use **!mac-ext** to disable.

**read\_concatenated\_archives**

Ignore zeroed blocks in the archive, which occurs when multiple tar archives have been concatenated together. Without this option, only the contents of the first concatenated archive would be read.

## ERRORS

Detailed error codes and textual descriptions are available from the **archive\_errno()** and **archive\_error\_string()** functions.

## SEE ALSO

tar(1), libarchive(3), archive\_write\_set\_options(3), archive\_read(3)