NAME

archive_read_support_filter_all, **archive_read_support_filter_bzip2**, **archive_read_support_filter_compress**, **archive_read_support_filter_gzip**, **archive_read_support_filter_lz4**, **archive_read_support_filter_lzma**, **archive_read_support_filter_none**, **archive_read_support_filter_rpm**, **archive_read_support_filter_uu**, **archive_read_support_filter_xz**, **archive_read_support_filter_zstd**, **archive_read_support_filter_program**, **archive_read_support_filter_program_signature** — functions for reading streaming archives

LIBRARY

Streaming Archive Library (libarchive, -larchive)

SYNOPSIS

```
#include <archive.h>

int
archive_read_support_filter_all(struct archive *);

int
archive_read_support_filter_by_code(struct archive *, int);

int
archive_read_support_filter_bzip2(struct archive *);

int
archive_read_support_filter_compress(struct archive *);

int
archive_read_support_filter_grzip(struct archive *);

int
archive_read_support_filter_gzip(struct archive *);

int
archive_read_support_filter_lrzip(struct archive *);

int
archive_read_support_filter_lz4(struct archive *);

int
archive_read_support_filter_lzma(struct archive *);

int
archive_read_support_filter_lzop(struct archive *);

int
archive_read_support_filter_none(struct archive *);

int
archive_read_support_filter_rpm(struct archive *);

int
archive_read_support_filter_uu(struct archive *);

int
archive_read_support_filter_xz(struct archive *);
```

```
int
archive_read_support_filter_zstd(struct archive *);

int
archive_read_support_filter_program(struct archive *, const char *cmd);

int
archive_read_support_filter_program_signature(struct archive *,
    const char *cmd, const void *signature, size_t signature_length);
```

DESCRIPTION

archive_read_support_filter_bzip2(), **archive_read_support_filter_compress**(), **archive_read_support_filter_grzip**(), **archive_read_support_filter_gzip**(), **archive_read_support_filter_lrzip**(), **archive_read_support_filter_lz4**(), **archive_read_support_filter_lzma**(), **archive_read_support_filter_lzop**(), **archive_read_support_filter_none**(), **archive_read_support_filter_rpm**(), **archive_read_support_filter_uu**(), **archive_read_support_filter_xz**(), **archive_read_support_filter_zstd**(), Enables auto-detection code and decompression support for the specified compression. These functions may fall back on external programs if an appropriate library was not available at build time. Decompression using an external program is usually slower than decompression through built-in libraries. Note that "none" is always enabled by default.

archive_read_support_filter_all()
: Enables all available decompression filters.

archive_read_support_filter_by_code()
: Enables a single filter specified by the filter code. This function does not work with **ARCHIVE_FILTER_PROGRAM**. Note: In statically-linked executables, this will cause your program to include support for every filter. If executable size is a concern, you may wish to avoid using this function.

archive_read_support_filter_program()
: Data is fed through the specified external program before being dearchived. Note that this disables automatic detection of the compression format, so it makes no sense to specify this in conjunction with any other decompression option.

archive_read_support_filter_program_signature()
: This feeds data through the specified external program but only if the initial bytes of the data match the specified signature value.

RETURN VALUES

These functions return **ARCHIVE_OK** if the compression is fully supported, **ARCHIVE_WARN** if the compression is supported only through an external program.

archive_read_support_filter_none() always succeeds.

ERRORS

Detailed error codes and textual descriptions are available from the **archive_errno**() and **archive_error_string**() functions.

SEE ALSO

archive_read(3), archive_read_data(3), archive_read_format(3), archive_read_format(3), libarchive(3)