NAME

archive_read_data, **archive_read_data_block**, **archive_read_data_skip**, **archive_read_data_into_fd** — functions for reading streaming archives

LIBRARY

Streaming Archive Library (libarchive, -larchive)

SYNOPSIS

```
#include <archive.h>

la_ssize_t
archive_read_data(struct archive *, void *buff, size_t len);

int
archive_read_data_block(struct archive *, const void **buff, size_t *len,
    off_t *offset);

int
archive_read_data_skip(struct archive *);

int
archive_read_data_into_fd(struct archive *, int fd);
```

DESCRIPTION

archive_read_data()
: Read data associated with the header just read. Internally, this is a convenience function that calls **archive_read_data_block**() and fills any gaps with nulls so that callers see a single continuous stream of data.

archive_read_data_block()
: Return the next available block of data for this entry. Unlike **archive_read_data**(), the **archive_read_data_block**() function avoids copying data and allows you to correctly handle sparse files, as supported by some archive formats. The library guarantees that offsets will increase and that blocks will not overlap. Note that the blocks returned from this function can be much larger than the block size read from disk, due to compression and internal buffer optimizations.

archive_read_data_skip()
: A convenience function that repeatedly calls **archive_read_data_block**() to skip all of the data for this archive entry. Note that this function is invoked automatically by **archive_read_next_header2**() if the previous entry was not completely consumed.

archive_read_data_into_fd()
: A convenience function that repeatedly calls **archive_read_data_block**() to copy the entire entry to the provided file descriptor.

RETURN VALUES

Most functions return zero on success, non-zero on error. The possible return codes include: **ARCHIVE_OK** (the operation succeeded), **ARCHIVE_WARN** (the operation succeeded but a non-critical error was encountered), **ARCHIVE_EOF** (end-of-archive was encountered), **ARCHIVE_RETRY** (the operation failed but can be retried), and **ARCHIVE_FATAL** (there was a fatal error; the archive should be closed immediately).

archive_read_data() returns a count of bytes actually read or zero at the end of the entry. On error, a value of **ARCHIVE_FATAL**, **ARCHIVE_WARN**, or **ARCHIVE_RETRY** is returned.

ERRORS

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

SEE ALSO

tar(1), archive_read(3), archive_read_extract(3), archive_read_filter(3), archive_read_format(3), archive_read_header(3), archive_read_open(3), archive_read_set_options(3), archive_util(3), libarchive(3), tar(5)