

NAME

archive_write_set_format, **archive_write_set_format_7zip,**
archive_write_set_format_ar, **archive_write_set_format_ar_bsd,**
archive_write_set_format_ar_svr4, **archive_write_set_format_by_name,**
archive_write_set_format_cpio, **archive_write_set_format_cpio_newc,**
archive_write_set_format_filter_by_ext,
archive_write_set_format_filter_by_ext_def,
archive_write_set_format_gnutar, **archive_write_set_format_iso9660,**
archive_write_set_format_mtree, **archive_write_set_format_mtree_classic,**
archive_write_set_format_mtree_default, **archive_write_set_format_pax,**
archive_write_set_format_pax_restricted, **archive_write_set_format_raw,**
archive_write_set_format_shar, **archive_write_set_format_shar_dump,**
archive_write_set_format_ustar, **archive_write_set_format_v7tar,**
archive_write_set_format_warc, **archive_write_set_format_xar,**
archive_write_set_format_zip, — functions for creating archives

LIBRARY

Streaming Archive Library (libarchive, -larchive)

SYNOPSIS

```

#include <archive.h>

int
archive_write_set_format(struct archive *, int code);

int
archive_write_set_format_7zip(struct archive *);

int
archive_write_set_format_ar(struct archive *);

int
archive_write_set_format_ar_bsd(struct archive *);

int
archive_write_set_format_ar_svr4(struct archive *);

int
archive_write_set_format_by_name(struct archive *, const char *name);

int
archive_write_set_format_cpio(struct archive *);

int
archive_write_set_format_cpio_newc(struct archive *);

int
archive_write_set_format_filter_by_ext(struct archive *,
    const char *filename);

int
archive_write_set_format_filter_by_ext_def(struct archive *,
    const char *filename, const char *def_ext);

int
archive_write_set_format_gnutar(struct archive *);

```

```

int
archive_write_set_format_iso9660(struct archive *);

int
archive_write_set_format_mtree(struct archive *);

int
archive_write_set_format_pax(struct archive *);

int
archive_write_set_format_pax_restricted(struct archive *);

int
archive_write_set_format_raw(struct archive *);

int
archive_write_set_format_shar(struct archive *);

int
archive_write_set_format_shar_dump(struct archive *);

int
archive_write_set_format_ustar(struct archive *);

int
archive_write_set_format_v7tar(struct archive *);

int
archive_write_set_format_warc(struct archive *);

int
archive_write_set_format_xar(struct archive *);

int
archive_write_set_format_zip(struct archive *);

```

DESCRIPTION

These functions set the format that will be used for the archive.

The library can write a variety of common archive formats.

archive_write_set_format()

Sets the format based on the format code (see `archive.h` for the full list of format codes). In particular, this can be used in conjunction with **archive_format()** to create a new archive with the same format as an existing archive.

archive_write_set_format_by_name()

Sets the corresponding format based on the common name.

archive_write_set_format_filter_by_ext(), archive_write_set_format_filter_by_ext_def()

Sets both filters and format based on the output filename. Supported extensions: `.7z`, `.zip`, `.jar`, `.cpio`, `.iso`, `.a`, `.ar`, `.tar`, `.tgz`, `.tar.gz`, `.tar.bz2`, `.tar.xz`

```

archive_write_set_format_7zip()           archive_write_set_format_ar_bsd(),
archive_write_set_format_ar_svr4(),      archive_write_set_format_cpio()
archive_write_set_format_cpio_newc()
archive_write_set_format_gnutar() archive_write_set_format_iso9660()
archive_write_set_format_mtree()
archive_write_set_format_mtree_classic()

```

```
archive_write_set_format_pax()  
archive_write_set_format_pax_restricted()  
archive_write_set_format_raw()      archive_write_set_format_shar()  
archive_write_set_format_shar_dump()  
archive_write_set_format_ustar()    archive_write_set_format_v7tar()  
archive_write_set_format_warc()     archive_write_set_format_xar()  
archive_write_set_format_zip()
```

Set the format as specified. More details on the formats supported by libarchive can be found in the `libarchive-formats(5)` manual page.

RETURN VALUES

These functions return **ARCHIVE_OK** on success, or **ARCHIVE_FATAL**.

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(3)`, `archive_write_set_options(3)`, `cpio(5)`, `libarchive-formats(5)`, `mtree(5)`, `tar(5)`