

**NAME**

**archive\_write\_set\_format\_cpio**, **archive\_write\_set\_format\_pax**,  
**archive\_write\_set\_format\_pax\_restricted**, **archive\_write\_set\_format\_shar**,  
**archive\_write\_set\_format\_shar\_dump**, **archive\_write\_set\_format\_ustar** — func-  
tions for creating archives

**LIBRARY**

Streaming Archive Library (libarchive, -larchive)

**SYNOPSIS**

```
#include <archive.h>

int
archive_write_set_format_cpio(struct archive *);

int
archive_write_set_format_pax(struct archive *);

int
archive_write_set_format_pax_restricted(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 *);
```

**DESCRIPTION**

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

The library can write POSIX octet-oriented cpio format archives, POSIX-standard “pax interchange” format archives, traditional “shar” archives, enhanced “dump” shar archives that store a variety of file attributes and handle binary files, and POSIX-standard “ustar” archives. The pax interchange format is a backwards-compatible tar format that adds key/value attributes to each entry and supports arbitrary filenames, linknames, uids, sizes, etc. “Restricted pax interchange format” is the library default; this is the same as pax format, but suppresses the pax extended header for most normal files. In most cases, this will result in ordinary ustar archives.

**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)**,  
**mtree(5)**, **tar(5)**