

**NAME**

**archive\_entry\_hardlink**, **archive\_entry\_hardlink\_w**,  
**archive\_entry\_set\_hardlink**, **archive\_entry\_copy\_hardlink**,  
**archive\_entry\_copy\_hardlink\_w**, **archive\_entry\_update\_hardlink\_utf8**,  
**archive\_entry\_set\_link**, **archive\_entry\_copy\_link**, **archive\_entry\_copy\_link\_w**,  
**archive\_entry\_update\_link\_utf8**, **archive\_entry\_pathname**,  
**archive\_entry\_pathname\_w**, **archive\_entry\_set\_pathname**,  
**archive\_entry\_copy\_pathname**, **archive\_entry\_copy\_pathname\_w**,  
**archive\_entry\_update\_pathname\_utf8**, **archive\_entry\_sourcepath**,  
**archive\_entry\_copy\_sourcepath**, **archive\_entry\_symlink**,  
**archive\_entry\_symlink\_w**, **archive\_entry\_set\_symlink**,  
**archive\_entry\_copy\_symlink**, **archive\_entry\_copy\_symlink\_w**,  
**archive\_entry\_update\_symlink\_utf8** — functions for manipulating path names in archive entry descriptions

**LIBRARY**

Streaming Archive Library (libarchive, -larchive)

**SYNOPSIS**

```

#include <archive_entry.h>

const char *
archive_entry_hardlink(struct archive_entry *a);

const wchar_t *
archive_entry_hardlink_w(struct archive_entry *a);

void
archive_entry_set_hardlink(struct archive_entry *a, const char *path);

void
archive_entry_copy_hardlink(struct archive_entry *a, const char *path);

void
archive_entry_copy_hardlink_w(struct archive_entry *a, const wchar_t,
    *path");

int
archive_entry_update_hardlink_utf8(struct archive_entry *a,
    const char *path);

void
archive_entry_set_link(struct archive_entry *a, const char *path);

void
archive_entry_copy_link(struct archive_entry *a, const char *path);

void
archive_entry_copy_link_w(struct archive_entry *a, const wchar_t *path);

int
archive_entry_update_link_utf8(struct archive_entry *a, const char *path);

const char *
archive_entry_pathname(struct archive_entry *a);
  
```

```

const wchar_t *
archive_entry_pathname_w(struct archive_entry *a);
void
archive_entry_set_pathname(struct archive_entry *a, const char *path);
void
archive_entry_copy_pathname(struct archive_entry *a, const char *path);
void
archive_entry_copy_pathname_w(struct archive_entry *a,
    const wchar_t *path);
int
archive_entry_update_pathname_utf8(struct archive_entry *a,
    const char *path);
const char *
archive_entry_sourcepath(struct archive_entry *a);
void
archive_entry_copy_sourcepath(struct archive_entry *a, const char *path);
const char *
archive_entry_symlink(struct archive_entry *a);
const wchar_t *
archive_entry_symlink_w(struct archive_entry *a);
void
archive_entry_set_symlink(struct archive_entry *a, const char *path);
void
archive_entry_copy_symlink(struct archive_entry *a, const char *path);
void
archive_entry_copy_symlink_w(struct archive_entry *a, const wchar_t *path);
int
archive_entry_update_symlink_utf8(struct archive_entry *a,
    const char *path);

```

## DESCRIPTION

Path names supported by `archive_entry(3)`:

<code>hardlink</code>	Destination of the hardlink.
<code>link</code>	Update only. For a symlink, update the destination. Otherwise, make the entry a hardlink and alter the destination for that.
<code>pathname</code>	Path in the archive
<code>sourcepath</code>	Path on the disk for use by <code>archive_read_disk(3)</code> .
<code>symlink</code>	Destination of the symbolic link.

Path names can be provided in one of three different ways:

<code>char *</code>	Multibyte strings in the current locale.
<code>wchar_t *</code>	Wide character strings in the current locale. The accessor functions are named <b>XXX_w()</b> .
UTF-8	Unicode strings encoded as UTF-8. This are convenience functions to update both the multibyte and wide character strings at the same time.

The sourcepath is a pure filesystem concept and never stored in an archive directly.

For that reason, it is only available as multibyte string. The link path is a convenience function for conditionally setting hardlink or symlink destination. It doesn't have a corresponding get accessor function.

**archive\_entry\_set\_XXX()** is an alias for **archive\_entry\_copy\_XXX()**.

**SEE ALSO**

archive\_entry(3) libarchive(3),