

**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)`:

`hardlink` Destination of the hardlink.

`link` Update only. For a symlink, update the destination. Otherwise, make the entry a hardlink and alter the destination for that.

`pathname` Path in the archive

`sourcepath` Path on the disk for use by `archive_read_disk(3)`.

`symlink` Destination of the symbolic link.

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

`char *` Multibyte strings in the current locale.

`wchar_t *` Wide character strings in the current locale. The accessor functions are named **XXX\_w()**.

**UTF-8** Unicode strings encoded as UTF-8. These 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)`