

**NAME**

**zip\_source\_layered**, **zip\_source\_layered\_create** - create layered data source from function

**LIBRARY**

libzip (-lzip)

**SYNOPSIS**

```
#include <zip.h>
```

```
zip_source_t *
```

```
zip_source_layered(zip_t *archive, zip_source_t *source, zip_source_layered_callback fn,  
void *userdata);
```

```
zip_source_t *
```

```
zip_source_layered_create(zip_source_t *source, zip_source_layered_callback fn, void *userdata,  
zip_error_t *error);
```

**DESCRIPTION**

The functions **zip\_source\_layered**() and **zip\_source\_layered\_create**() create a layered zip source from the user-provided function *fn*, which must be of the following type:

```
typedef zip_int64_t (*p_source_layered_callback)(zip_source_t *source, void *userdata, void *data,  
zip_uint64_t length, zip_source_cmd_t cmd)
```

*archive* or *error* are used for reporting errors and can be NULL.

When called by the library, the first argument is the *source* of the lower layer, the second argument is the *userdata* argument supplied to the function. The next two arguments are a buffer *data* of size *length* when data is passed in or expected to be returned, or else NULL and 0. The last argument, *cmd*, specifies which action the function should perform.

See `zip_source_function(3)` for a description of the commands.

A layered source transforms the data or metadata of the source below in some way. Layered sources can't support writing and are not sufficient to cleanly add support for additional compression or encryption methods. This may be revised in a later release of libzip.

On success, the layered source takes ownership of *source*. The caller should not free it.

The interaction with the lower layer depends on the command:

### ZIP\_SOURCE\_ACCEPT\_EMPTY

If the layered source supports this command, the lower layer is not called automatically. Otherwise, the return value of the lower source is used.

### ZIP\_SOURCE\_CLOSE

The lower layer is closed after the callback returns.

### ZIP\_SOURCE\_ERROR

The lower layer is not called automatically. If you need to retrieve error information from the lower layer, use `zip_error_set_from_source(3)` or `zip_source_pass_to_lower_layer(3)`.

### ZIP\_SOURCE\_FREE

The lower layer is freed after the callback returns.

### ZIP\_SOURCE\_GET\_FILE\_ATTRIBUTES

The attributes of the lower layer are merged with the attributes returned by the callback: information set by the callback wins over the lower layer, with the following exceptions: the higher *version\_needed* is used, and *general\_purpose\_bit\_flags* are only overwritten if the corresponding bit is set in *general\_purpose\_bit\_mask*.

### ZIP\_SOURCE\_OPEN

The lower layer is opened before the callback is called.

### ZIP\_SOURCE\_READ

The lower layer is not called automatically.

### ZIP\_SOURCE\_SEEK

The lower layer is not called automatically.

### ZIP\_SOURCE\_STAT

*data* contains the stat information from the lower layer when the callback is called.

### ZIP\_SOURCE\_SUPPORTS

*data* contains the bitmap of commands supported by the lower layer when the callback is called. Since layered sources can't support writing, all commands related to writing are stripped from the returned support bitmap.

### ZIP\_SOURCE\_TELL

The lower layer is not called automatically.

**RETURN VALUES**

Upon successful completion, the created source is returned. Otherwise, NULL is returned and the error code in *archive* or *error* is set to indicate the error (unless it is NULL).

**ERRORS**

**zip\_source\_layered()** fails if:

[ZIP\_ER\_MEMORY]

Required memory could not be allocated.

**SEE ALSO**

libzip(3), zip\_file\_add(3), zip\_file\_attributes\_init(3), zip\_file\_replace(3), zip\_source(3), zip\_source\_function(3), zip\_source\_pass\_to\_lower\_layer(3)

**HISTORY**

**zip\_source\_layered()** and **zip\_source\_layered\_create()** were added in libzip 1.10.

**AUTHORS**

Dieter Baron <dillo@nih.at> and Thomas Klausner <tk@giga.or.at>