

NAME

zip_open, **zip_open_from_source** - open zip archive

LIBRARY

libzip (-lzip)

SYNOPSIS

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

```
zip_t *
```

```
zip_open(const char *path, int flags, int *errorp);
```

```
zip_t *
```

```
zip_open_from_source(zip_source_t *zs, int flags, zip_error_t *ze);
```

DESCRIPTION

The **zip_open()** function opens the zip archive specified by *path* and returns a pointer to a *struct zip*, used to manipulate the archive. The *flags* are specified by *or*'ing the following values, or 0 for none of them.

| | |
|---------------|---|
| ZIP_CHECKCONS | Perform additional stricter consistency checks on the archive, and error if they fail. |
| ZIP_CREATE | Create the archive if it does not exist. |
| ZIP_EXCL | Error if archive already exists. |
| ZIP_TRUNCATE | If archive exists, ignore its current contents. In other words, handle it the same way as an empty archive. |
| ZIP_RDONLY | Open archive in read-only mode. |

If an error occurs and *errorp* is non-NULL, it will be set to the corresponding error code.

The **zip_open_from_source()** function opens a zip archive encapsulated by the *zip_source* *zs* using the provided *flags*. In case of error, the *zip_error* *ze* is filled in.

RETURN VALUES

Upon successful completion **zip_open()** and **zip_open_from_source()** return a *struct zip* pointer. Otherwise, NULL is returned and **zip_open()** sets **errorp* to indicate the error, while **zip_open_from(source)** sets *ze* to indicate the error.

EXAMPLES

Here's an example of how you could report errors during **zip_open**:

```
zip_t *za;
int err;

if ((za = zip_open(name, 0, &err)) == NULL) {
    zip_error_t error;
    zip_error_init_with_code(&error, err);
    fprintf(stderr, "%s: cannot open zip archive '%s': %s\n",
            progname, name, zip_error_strerror(&error));
    zip_error_fini(&error);
    return -1;
}
```

ERRORS

The archive specified by *path* is opened unless:

[ZIP_ER_EXISTS] The file specified by *path* exists and ZIP_EXCL is set.

[ZIP_ER_INCONS] Inconsistencies were found in the file specified by *path*. This error is often caused by specifying ZIP_CHECKCONS but can also happen without it.

[ZIP_ER_INVALID] The *path* argument is NULL.

[ZIP_ER_MEMORY] Required memory could not be allocated.

[ZIP_ER_NOENT] The file specified by *path* does not exist and ZIP_CREATE is not set.

[ZIP_ER_NOZIP] The file specified by *path* is not a zip archive.

[ZIP_ER_OPEN] The file specified by *path* could not be opened.

[ZIP_ER_READ] A read error occurred; see *errno* for details.

[ZIP_ER_SEEK] The file specified by *path* does not allow seeks.

For newly created archives, **zip_open()** does not try to create the file; this is done when calling **zip_close(3)** and any errors, like missing write permissions, will be reported then.

SEE ALSO

libzip(3), zip_close(3), zip_error_strerror(3), zip_fdopen(3)

HISTORY

zip_open() and **zip_open_from_source()** were added in libzip 1.0.

AUTHORS

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