

Zip

Introduction

This extension enables you to transparently read or write ZIP compressed archives and the files inside them.

Installing/Configuring

Requirements

PHP 4

The bundled PHP 4 version requires [» ZZIPlib](#), by Guido Draheim, version 0.10.6 or later

PHP 5.2.0 or later

This extension uses the functions of [» zlib](#) by Jean-loup Gailly and Mark Adler.

Installation

PHP 4

Note
Zip support before PHP 4.1.0 is experimental.

Warning
Because the PHP 4 zip extension is unmaintained we recommend that the PECL extension is used rather than the bundled one.

Linux systems

In order to use these functions you must compile PHP with zip support by using the `--with-zip[=DIR]` configure option, where [DIR] is the prefix of the [» ZZIPlib](#) library install.

Windows

Windows users need to enable `php_zip.dll` inside of `php.ini` in order to use these functions.

PHP 5.2.0 and later

Linux systems

In order to use these functions you must compile PHP with zip support by using the `--enable-zip` configure option.

Windows

Windows users need to enable `php_zip.dll` inside of `php.ini` in order to use these functions.

Installation via PECL

Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: [» http://pecl.php.net/package/zip](http://pecl.php.net/package/zip).

The DLL for this PECL extension may be downloaded from either the [» PHP Downloads](#) page or from [» http://pecl4win.php.net/](http://pecl4win.php.net/)

In PHP 4 this DLL resides in the `extensions/` directory within the PHP Windows binaries download.

Runtime Configuration

This extension has no configuration directives defined in `php.ini`.

Resource Types

There are two resource types used in the Zip module. The first one is the Zip directory for the Zip archive, the second Zip Entry for the archive entries.

Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

ZipArchive uses class constants. There is three types of constants, Flags (FL_) errors (ER_) or mode (no prefix).

ZIPARCHIVE::CREATE ([integer](#))

Create the archive if it does not exist.

ZIPARCHIVE::OVERWRITE ([integer](#))

Always start a new archive, this mode will overwrite the file if it already exists.

ZIPARCHIVE::EXCL ([integer](#))

Error if archive already exists.

ZIPARCHIVE::CHECKCONS ([integer](#))

Perform additional consistency checks on the archive, and error if they fail.

ZIPARCHIVE::FL_NOCASE ([integer](#))

Ignore case on name lookup

ZIPARCHIVE::FL_NODIR ([integer](#))

Ignore directory component

ZIPARCHIVE::FL_COMPRESSED ([integer](#))

Read compressed data

ZIPARCHIVE::FL_UNCHANGED ([integer](#))

Use original data, ignoring changes.

ZIPARCHIVE::CM_DEFAULT ([integer](#))

better of deflate or store.

ZIPARCHIVE::CM_STORE ([integer](#))

stored (uncompressed).

ZIPARCHIVE::CM_SHRINK ([integer](#))

shrunk

ZIPARCHIVE::CM_REDUCE_1 ([integer](#))

reduced with factor 1

ZIPARCHIVE::CM_REDUCE_2 ([integer](#))

reduced with factor 2

ZIPARCHIVE::CM_REDUCE_3 ([integer](#))

reduced with factor 3

ZIPARCHIVE::CM_REDUCE_4 ([integer](#))

reduced with factor 4

ZIPARCHIVE::CM_IMPLODE ([integer](#))

imploded

ZIPARCHIVE::CM_DEFLATE ([integer](#))

deflated

ZIPARCHIVE::CM_DEFLATE64 ([integer](#))

deflate64

ZIPARCHIVE::CM_PKWARE_IMPLODE ([integer](#))

PKWARE imploding

ZIPARCHIVE::CM_BZIP2 ([integer](#))

BZIP2 algorithm

ZIPARCHIVE::ER_OK ([integer](#))

No error.

ZIPARCHIVE::ER_MULTIDISK ([integer](#))

Multi-disk zip archives not supported.

ZIPARCHIVE::ER_RENAME ([integer](#))

Renaming temporary file failed.

ZIPARCHIVE::ER_CLOSE ([integer](#))

Closing zip archive failed

ZIPARCHIVE::ER_SEEK ([integer](#))

Seek error

ZIPARCHIVE::ER_READ ([integer](#))

Read error

ZIPARCHIVE::ER_WRITE ([integer](#))

Write error

ZIPARCHIVE::ER_CRC ([integer](#))

CRC error

ZIPARCHIVE::ER_ZIPCLOSED ([integer](#))

Containing zip archive was closed

ZIPARCHIVE::ER_NOENT ([integer](#))

No such file.

ZIPARCHIVE::ER_EXISTS ([integer](#))

File already exists

ZIPARCHIVE::ER_OPEN ([integer](#))

Can't open file

ZIPARCHIVE::ER_TMPOPEN ([integer](#))

Failure to create temporary file.

ZIPARCHIVE::ER_ZLIB ([integer](#))

Zlib error

ZIPARCHIVE::ER_MEMORY ([integer](#))

Memory allocation failure

ZIPARCHIVE::ER_CHANGED ([string](#))

Entry has been changed

ZIPARCHIVE::ER_COMPNOTSUPP ([integer](#))

Compression method not supported.

ZIPARCHIVE::ER_EOF ([integer](#))

Premature EOF

ZIPARCHIVE::ER_INVALID ([integer](#))

Invalid argument

ZIPARCHIVE::ER_NOZIP ([integer](#))

Not a zip archive

ZIPARCHIVE::ER_INTERNAL ([integer](#))

Internal error

ZIPARCHIVE::ER_INCONS ([integer](#))

Zip archive inconsistent

ZIPARCHIVE::ER_REMOVE ([integer](#))

Can't remove file

ZIPARCHIVE::ER_DELETED ([integer](#))

Entry has been deleted

Examples

Example #1 - Create a Zip archive

```
<?php

$zip = new ZipArchive();
$filename = "./test112.zip";

if ($zip->open($filename, ZIPARCHIVE::CREATE)!==TRUE) {
    exit("cannot open <$filename>\n");
}

$zip->addFromString("testfilephp.txt" . time(), "#1 This is a test string
added as testfilephp.txt.\n");
$zip->addFromString("testfilephp2.txt" . time(), "#2 This is a test string
added as testfilephp2.txt.\n");
$zip->addFile($thisdir . "/too.php", "/testfromfile.php");
echo "numfiles: " . $zip->numFiles . "\n";
echo "status:" . $zip->status . "\n";
$zip->close();
?>
```

Example #2 - Dump the archive details and listing

```
<?php
$za = new ZipArchive();

$za->open('test_with_comment.zip');
print_r($za);
var_dump($za);
echo "numFiles: " . $za->numFiles . "\n";
echo "status: " . $za->status . "\n";
echo "statusSys: " . $za->statusSys . "\n";
echo "filename: " . $za->filename . "\n";
echo "comment: " . $za->comment . "\n";

for ($i=0; $i<$za->numFiles;$i++) {
    echo "index: $i\n";
    print_r($za->statIndex($i));
}
echo "numFile:" . $za->numFiles . "\n";
?>
```

Example #3 - Zip stream wrapper, read an OpenOffice meta info

```
<?php
$reader = new XMLReader();

$reader->open('zip://' . dirname(__FILE__) . '/test.odt#meta.xml');
$oedt_meta = array();
```



```

while ($reader->read()) {
    if ($reader->nodeType == XMLREADER::ELEMENT) {
        $selm = $reader->name;
    } else {
        if ($reader->nodeType == XMLREADER::END_ELEMENT && $reader->name ==
'office:meta') {
            break;
        }
        if (!trim($reader->value)) {
            continue;
        }
        $odt_meta[$selm] = $reader->value;
    }
}
print_r($odt_meta);
?>

```

This example uses the old API (PHP 4), it opens a ZIP file archive, reads each file in the archive and prints out its contents. The *test2.zip* archive used in this example is one of the test archives in the ZZIPlib source distribution.

Example #4 - Zip Usage Example

```

<?php

$zip = zip_open("/tmp/test2.zip");

if ($zip) {
    while ($zip_entry = zip_read($zip)) {
        echo "Name:           " . zip_entry_name($zip_entry) . "\n";
        echo "Actual Filesize:  " . zip_entry_filesize($zip_entry) . "\n";
        echo "Compressed Size:   " . zip_entry_compressedsize($zip_entry) .
"\n";
        echo "Compression Method: " . zip_entry_compressionmethod($zip_entry)
. "\n";

        if (zip_entry_open($zip, $zip_entry, "r")) {
            echo "File Contents:\n";
            $buf = zip_entry_read($zip_entry,
zip_entry_filesize($zip_entry));
            echo "$buf\n";

            zip_entry_close($zip_entry);
        }
        echo "\n";
    }

    zip_close($zip);
}
?>

```

Zip Functions

zip_close

zip_close -- Close a ZIP file archive

Description

void zip_close (resource *\$zip*)

Closes the given ZIP file archive.

Parameters

zip
A ZIP file previously opened with [zip_open\(\)](#).

Return Values

No value is returned.

See Also

- [zip_open\(\)](#)
- [zip_read\(\)](#)

zip_entry_close

zip_entry_close -- Close a directory entry

Description

bool **zip_entry_close** (resource \$zip_entry)

Closes the specified directory entry.

Parameters

zip_entry

A directory entry previously opened [zip_entry_open\(\)](#).

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- [zip_entry_open\(\)](#)
- [zip_entry_read\(\)](#)

zip_entry_compressedsize

zip_entry_compressedsize -- Retrieve the compressed size of a directory entry

Description

```
int zip_entry_compressedsize ( resource $zip_entry )
```

Returns the compressed size of the specified directory entry.

Parameters

zip_entry

A directory entry returned by [zip_read\(\)](#).

Return Values

The compressed size.

See Also

- [zip_open\(\)](#)
- [zip_read\(\)](#)

zip_entry_compressionmethod

zip_entry_compressionmethod -- Retrieve the compression method of a directory entry

Description

string **zip_entry_compressionmethod** (resource \$zip_entry)

Returns the compression method of the directory entry specified by *zip_entry*.

Parameters

zip_entry

A directory entry returned by [zip_read\(\)](#).

Return Values

The compression method.

See Also

- [zip_open\(\)](#)
- [zip_read\(\)](#)

zip_entry_filesize

zip_entry_filesize -- Retrieve the actual file size of a directory entry

Description

```
int zip_entry_filesize ( resource $zip_entry )
```

Returns the actual size of the specified directory entry.

Parameters

zip_entry

A directory entry returned by [zip_read\(\)](#).

Return Values

The size of the directory entry.

See Also

- [zip_open\(\)](#)
- [zip_read\(\)](#)

zip_entry_name

zip_entry_name -- Retrieve the name of a directory entry

Description

string **zip_entry_name** (resource \$zip_entry)

Returns the name of the specified directory entry.

Parameters

zip_entry

A directory entry returned by [zip_read\(\)](#).

Return Values

The name of the directory entry.

See Also

- [zip_open\(\)](#)
- [zip_read\(\)](#)

zip_entry_open

zip_entry_open -- Open a directory entry for reading

Description

bool **zip_entry_open** (resource \$zip, resource \$zip_entry [, string \$mode])

Opens a directory entry in a zip file for reading.

Parameters

zip

A valid resource handle returned by [zip_open\(\)](#).

zip_entry

A directory entry returned by [zip_read\(\)](#).

mode

Any of the modes specified in the documentation of [fopen\(\)](#).

Note
Currently, <i>mode</i> is ignored and is always "rb". This is due to the fact that zip support in PHP is read only access.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Note
Unlike fopen() and other similar functions, the return value of zip_entry_open() only indicates the result of the operation and is not needed for reading or closing the directory entry.

See Also

- [zip_entry_close\(\)](#)
- [zip_entry_read\(\)](#)

zip_entry_read

zip_entry_read -- Read from an open directory entry

Description

string **zip_entry_read** (resource \$zip_entry [, int \$length])

Reads from an open directory entry.

Parameters

zip_entry

A directory entry returned by [zip_read\(\)](#).

length

The number of bytes to return. If not specified, this function will attempt to read 1024 bytes.

Note
This should be the uncompressed length you wish to read.

Return Values

Returns the data read, or **FALSE** if the end of the file is reached.

See Also

- [zip_entry_open\(\)](#)
- [zip_entry_close\(\)](#)
- [zip_entry_filesize\(\)](#)

zip_open

zip_open -- Open a ZIP file archive

Description

mixed **zip_open** (string \$filename)

Opens a new zip archive for reading.

Parameters

filename

The file name of the ZIP archive to open.

Return Values

Returns a resource handle for later use with [zip_read\(\)](#) and [zip_close\(\)](#) or returns the number of error if *filename* does not exist or in case of other error.

See Also

- [zip_read\(\)](#)
- [zip_close\(\)](#)

zip_read

zip_read -- Read next entry in a ZIP file archive

Description

mixed **zip_read** (resource \$zip)

Reads the next entry in a zip file archive.

Parameters

zip

A ZIP file previously opened with [zip_open\(\)](#).

Return Values

Returns a directory entry resource for later use with the *zip_entry_...* functions or **FALSE** if there's no more entries to read or number of error in case of other error.

See Also

- [zip_open\(\)](#)
- [zip_close\(\)](#)
- [zip_entry_open\(\)](#)
- [zip_entry_read\(\)](#)

ZipArchive::addEmptyDir

ZipArchive::addEmptyDir -- Add a new directory

Description

bool **ZipArchive::addEmptyDir** (string \$dirname)

Adds an empty directory in the archive.

Parameters

dirname

The directory to add.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #5 - Create a new directory in an archive

```
<?php
$zip = new ZipArchive;
if ($zip->open('test.zip') === TRUE) {
    if($zip->addEmptyDir('newDirectory')) {
        echo 'Created a new root directory';
    } else {
        echo 'Could not create the directory';
    }
    $zip->close();
} else {
    echo 'failed';
}
?>
```

ZipArchive::addFile

ZipArchive::addFile -- Adds a file to a ZIP archive from the given path

Description

bool **ZipArchive::addFile** (string \$filename [, string \$localname])

Adds a file to a ZIP archive from a given path

Parameters

filename

The path to the file to add.

localname

local name inside ZIP archive.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

This example opens a ZIP file archive *test.zip* and add the file */path/to/index.txt*. as *newname.txt*.

Example #6 - Open and extract

```
<?php
$zip = new ZipArchive;
if ($zip->open('test.zip') === TRUE) {
    $zip->addFile('/path/to/index.txt', 'newname.txt');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::addFromString

ZipArchive::addFromString -- Add a file to a ZIP archive using its contents

Description

bool **ZipArchive::addFromString** (string \$localname, string \$contents)

Add a file to a ZIP archive using its contents.

Parameters

localname

The name of the entry to create.

contents

The contents to use to create the entry. It is used in a binary safe mode.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #7 - Add an entry to a new archive

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip', ZipArchive::CREATE);
if ($res === TRUE) {
    $zip->addFromString('test.txt', 'file content goes here');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

Example #8 - Add file to a directory inside an archive

```
<?php
$zip = new ZipArchive;
if ($zip->open('test.zip') === TRUE) {
    $zip->addFromString('dir/test.txt', 'file content goes here');
    $zip->close();
    echo 'ok';
}
```

```
} else {  
    echo 'failed';  
}  
?>
```


ZipArchive::close

ZipArchive::close -- Close the active archive (opened or newly created)

Description

bool **ZipArchive::close** (void)

Close opened or created archive and save changes. This method is automatically called at the end of the script.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

ZipArchive::deleteIndex

ZipArchive::deleteIndex -- delete an entry in the archive using its index

Description

bool **ZipArchive::deleteIndex** (int \$index)

Delete an entry in the archive using its index.

Parameters

index
Index of the entry to delete.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #9 - Delete file from archive using its index

```
<?php
$zip = new ZipArchive;
if ($zip->open('test.zip') === TRUE) {
    $zip->deleteIndex(2);
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::deleteName

ZipArchive::deleteName -- delete an entry in the archive using its name

Description

bool **ZipArchive::deleteName** (string \$name)

Delete an entry in the archive using its name.

Parameters

name

Name of the entry to delete.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #10 - Delete file from archive using its name

```
<?php
$zip = new ZipArchive;
if ($zip->open('test1.zip') === TRUE) {
    $zip->deleteName('testfromfile.php');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::extractTo

ZipArchive::extractTo -- Extract the archive contents

Description

mixed ZipArchive::extractTo (string \$destination [, **mixed** \$entries])

Extract the complete archive or the given files to the specified destination.

Parameters

destination

Location where to extract the files.

entries

The entries to extract. It accepts either a single entry name or an array of names.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

This example opens a ZIP file archive, reads each file in the archive and prints out its contents. The *test2.zip* archive used in this example is one of the test archives in the ZZIPLib source distribution.

Example #11 - Extract all entries

```
<?php
$zip = new ZipArchive;
if ($zip->open('test.zip') === TRUE) {
    $zip->extractTo('/my/destination/dir/');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

Example #12 - Extract only two entries

```
<?php
$zip = new ZipArchive;
```

```
$res = $zip->open('test_im.zip');  
if ($res === TRUE) {  
    $zip->extractTo('/my/destination/dir/', array('pear_item.gif',  
'testfromfile.php'));  
    $zip->close();  
    echo 'ok';  
} else {  
    echo 'failed';  
}  
?>
```

ZipArchive::getArchiveComment

ZipArchive::getArchiveComment -- Returns the Zip archive comment

Description

string **ZipArchive::getArchiveComment** (void)

Returns the Zip archive comment.

Return Values

Returns the Zip archive comment or **FALSE** on failure.

Examples

Example #13 - Dump an archive comment

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test_with_comment.zip')
if ($res === TRUE) {
    var_dump($zip->getArchiveComment());
    /* Or using the archive property */
    var_dump($zip->comment);
} else {
    echo 'failed, code:' . $res;
}
?>
```

ZipArchive::getCommentIndex

ZipArchive::getCommentIndex -- Returns the comment of an entry using the entry index

Description

string **ZipArchive::getCommentIndex** (int *\$index* [, int *\$flags*])

Returns the comment of an entry using the entry index.

Parameters

index

Index of the entry

flags

If flags is set to **ZIPARCHIVE::FL_UNCHANGED**, the original unchanged comment is returned.

Return Values

Returns the comment on success or **FALSE** on failure.

Examples

Example #14 - Dump an entry comment

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test1.zip')
if ($res === TRUE) {
    var_dump($zip->getCommentIndex(1));
} else {
    echo 'failed, code:' . $res;
}
?>
```

ZipArchive::getCommentName

ZipArchive::getCommentName -- Returns the comment of an entry using the entry name

Description

string **ZipArchive::getCommentName** (string \$name [, int \$flags])

Returns the comment of an entry using the entry name.

Parameters

name

Name of the entry

flags

If flags is set to **ZIPARCHIVE::FL_UNCHANGED**, the original unchanged comment is returned.

Return Values

Returns the comment on success or **FALSE** on failure.

Examples

Example #15 - Dump an entry comment

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test1.zip')
if ($res === TRUE) {
    var_dump($zip->getCommentName('test/entry1.txt'));
} else {
    echo 'failed, code:' . $res;
}
?>
```


ZipArchive::getFromIndex

ZipArchive::getFromIndex -- Returns the entry contents using its index.

Description

mixed ZipArchive::getFromIndex (int *\$index* [, int *\$flags*])

Returns the entry contents using its index.

Parameters

index

Index of the entry

flags

The flags to use to open the archive. the following values may be ORed to it.

- **ZIPARCHIVE::FL_UNCHANGED**
- **ZIPARCHIVE::FL_COMPRESSED**

Return Values

Returns the contents of the entry on success or **FALSE** on failure.

Examples

Example #16 - Get the file contents

```
<?php
$zip = new ZipArchive;
if ($zip->open('test.zip') === TRUE) {
    echo $zip->getFromIndex(2);
    $zip->close();
} else {
    echo 'failed';
}
?>
```

ZipArchive::getFromName

ZipArchive::getFromName -- Returns the entry contents using its name.

Description

mixed ZipArchive::getFromName (string *\$name* [, int *\$flags*])

Returns the entry contents using its name.

Parameters

name

Name of the entry

flags

The flags to use to open the archive. the following values may be ORed to it.

- **ZIPARCHIVE::FL_UNCHANGED**
- **ZIPARCHIVE::FL_COMPRESSED**

Return Values

Returns the contents of the entry on success or **FALSE** on failure.

Examples

Example #17 - Get the file contents

```
<?php
$zip = new ZipArchive;
if ($zip->open('test1.zip') === TRUE) {
    echo $zip->getFromName('testfromfile.php');
    $zip->close();
} else {
    echo 'failed';
}
?>
```

Example #18 - Convert an image from a zip entry

```
<?php
$z = new ZipArchive();
```

```
if ($z->open(dirname(__FILE__) . '/test_im.zip')) {  
    $im_string = $z->getFromName("pear_item.gif");  
    $im = imagecreatefromstring($im_string);  
    imagepng($im, 'b.png');  
}  
?>
```

ZipArchive::getNameIndex

ZipArchive::getNameIndex -- Returns the name of an entry using its index

Description

string **ZipArchive::getNameIndex** (int *\$index*)

Returns the name of an entry using its index.

Parameters

index
Index of the entry.

Return Values

Returns the name on success or **FALSE** on failure.

ZipArchive::getStream

ZipArchive::getStream -- Get a file handler to the entry defined by its name (read only).

Description

resource **ZipArchive::getStream** (string \$name)

Get a file handler to the entry defined by its name. For now it only supports read operations.

Parameters

name

The name of the entry to use.

Return Values

Returns a file pointer (resource) on success or **FALSE** on failure.

Examples

Example #19 - Get the entry contents with fread and store it.

```
<?php
$content = '';
$z = new ZipArchive();
if ($z->open('test.zip')) {
    $fp = $z->getStream('test');
    if(!$fp) exit("failed\n");

    while (!feof($fp)) {
        $contents .= fread($fp, 2);
    }

    fclose($fp);
    file_put_contents('t',$contents);
    echo "done.\n";
}
?>
```

Example #20 - Same as the previous example but with fopen and the zip stream wrapper

```
<?php
$fp = fopen('zip://' . dirname(__FILE__) . '/test.zip#test', 'r');
```

```
if (!$fp) {
    exit("cannot open\n");
}
while (!feof($fp)) {
    $contents .= fread($fp, 2);
    echo "$contents\n";
}
fclose($fp);
echo "done.\n";
?>
```

Example #21 - Stream wrapper and image, can be used with the xml function as well

```
<?php
$im = imagecreatefromgif('zip://' . dirname(__FILE__) .
'/test_im.zip#pear_item.gif');
imagepng($im, 'a.png');
?>
```

ZipArchive::locateName

ZipArchive::locateName -- Returns the index of the entry in the archive

Description

mixed ZipArchive::locateName (string \$name [, int \$flags])

Locates an entry using its name.

Parameters

name

The name of the entry to look up

flags

The function returns the index of the file named fname in archive. The flags are specified by ORing the following values, or 0 for none of them.

- **ZIPARCHIVE::FL_NOCASE**
- **ZIPARCHIVE::FL_NODIR**

Return Values

Returns the index of the entry on success or **FALSE** on failure.

Examples

Example #22 - Create an archive and then use it with locateName

```
<?php
$file = 'testlocate.zip';

$zip = new ZipArchive;
if ($zip->open($file, ZIPARCHIVE::CREATE) !== TRUE) {
    exit('failed');
}

$zip->addFromString('entry1.txt', 'entry #1');
$zip->addFromString('entry2.txt', 'entry #2');
$zip->addFromString('dir/entry2d.txt', 'entry #2');

if (!$zip->status == ZIPARCHIVE::ER_OK) {
    echo "failed to write zip\n";
}
$zip->close();
```

```
if ($zip->open($file) !== TRUE) {
    exit('failed');
}

echo $zip->locateName('entry1.txt') . "\n";
echo $zip->locateName('eNtry2.txt') . "\n";
echo $zip->locateName('eNtry2.txt', ZIPARCHIVE::FL_NOCASE) . "\n";
echo $zip->locateName('enTRy2d.txt',
ZIPARCHIVE::FL_NOCASE|ZIPARCHIVE::FL_NODIR) . "\n";
$zip->close();

?>
```


ZipArchive::open

ZipArchive::open -- Open a ZIP file archive

Description

mixed ZipArchive::open (string *\$filename* [, int *\$flags*])

Opens a new zip archive for reading, writing or modifying.

Parameters

filename

The file name of the ZIP archive to open.

flags

The mode to use to open the archive.

- **ZIPARCHIVE::OVERWRITE**
- **ZIPARCHIVE::CREATE**
- **ZIPARCHIVE::EXCL**
- **ZIPARCHIVE::CHECKCONS**

Return Values

Error codes

Returns **TRUE** on success or the error code.

- **ZIPARCHIVE::ER_EXISTS**
- **ZIPARCHIVE::ER_INCONS**
- **ZIPARCHIVE::ER_INVAL**
- **ZIPARCHIVE::ER_MEMORY**
- **ZIPARCHIVE::ER_NOENT**
- **ZIPARCHIVE::ER_NOZIP**
- **ZIPARCHIVE::ER_OPEN**
- **ZIPARCHIVE::ER_READ**
- **ZIPARCHIVE::ER_SEEK**

Examples

This example opens a ZIP file archive, reads each file in the archive and prints out its contents. The *test2.zip* archive used in this example is one of the test archives in the ZZIPLib source distribution.

Example #23 - Open and extract

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip')
if ($res === TRUE) {
    echo 'ok';
    $zip->extractTo('test');
    $zip->close();
} else {
    echo 'failed, code:' . $res;
}
?>
```

Example #24 - Create an archive

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip', ZipArchive::CREATE);
if ($res === TRUE) {
    $zip->addFromString('test.txt', 'file content goes here');
    $zip->addFile('data.txt', 'entryname.txt');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::renameIndex

ZipArchive::renameIndex -- Renames an entry defined by its index

Description

bool **ZipArchive::renameIndex** (int \$index, string \$newname)

Renames an entry defined by its index.

Parameters

index

Index of the entry to rename.

newname

New name.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #25 - Rename one entry

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip')
if ($res === TRUE) {
    $zip->renameIndex(2, 'newname.txt');
    $zip->close();
} else {
    echo 'failed, code:' . $res;
}
?>
```

ZipArchive::renameName

ZipArchive::renameName -- Renames an entry defined by its name

Description

bool **ZipArchive::renameName** (string \$name, string \$newname)

Renames an entry defined by its index.

Parameters

name

Name of the entry to rename.

newname

New name.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #26 - Rename one entry

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip');
if ($res === TRUE) {
    $zip->renameName('currentname.txt','newname.txt');
    $zip->close();
} else {
    echo 'failed, code:' . $res;
}
?>
```

ZipArchive::setArchiveComment

ZipArchive::setArchiveComment -- Set the comment of a ZIP archive

Description

mixed ZipArchive::setArchiveComment (string *\$comment*)

Set the comment of a ZIP archive.

Parameters

comment

The contents of the comment.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #27 - Create an archive and set a comment

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip', ZipArchive::CREATE);
if ($res === TRUE) {
    $zip->addFromString('test.txt', 'file content goes here');
    $z->setArchiveComment('new archive comment');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::setCommentIndex

ZipArchive::setCommentIndex -- Set the comment of an entry defined by its index

Description

mixed ZipArchive::setCommentIndex (int *\$index*, string *\$comment*)

Set the comment of an entry defined by its index.

Parameters

index

Index of the entry.

comment

The contents of the comment.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #28 - Open an archive and set a comment for an entry

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip');
if ($res === TRUE) {
    $z->setCommentIndex(2, 'new entry comment');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::setCommentName

ZipArchive::setCommentName -- Set the comment of an entry defined by its name

Description

mixed ZipArchive::setCommentName (string \$name, string \$comment)

Set the comment of an entry defined by its name.

Parameters

name

Name of the entry.

comment

The contents of the comment.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #29 - Open an archive and set a comment for an entry

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip');
if ($res === TRUE) {
    $z->setCommentName('entry1.txt', 'new entry comment');
    $zip->close();
    echo 'ok';
} else {
    echo 'failed';
}
?>
```

ZipArchive::statIndex

ZipArchive::statIndex -- Get the details of an entry defined by its index.

Description

mixed ZipArchive::statIndex (int \$index [, int \$flags])

The function obtains information about the entry defined by its index.

Parameters

index

Index of the entry

flags

ZIPARCHIVE::FL_UNCHANGED may be ORed to it to request information about the original file in the archive, ignoring any changes made.

Return Values

Returns an array containing the entry details or **FALSE** on failure.

Examples

Example #30 - Dump the stat info of an entry

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip')
if ($res === TRUE) {
    print_r($zip->statIndex(3));
    $zip->close();
} else {
    echo 'failed, code:' . $res;
}
?>
```

The above example will output something similar to:

```
Array
(
    [name] => foobar/baz
    [index] => 3
    [crc] => 499465816
    [size] => 27
    [mtime] => 1123164748
    [comp_size] => 24
)
```



```
[comp_method] => 8  
)
```

ZipArchive::statName

ZipArchive::statName -- Get the details of an entry defined by its name.

Description

mixed ZipArchive::statName (**name** \$name [, int \$flags])

The function obtains information about the entry defined by its name.

Parameters

name

Name of the entry

flags

The flags argument specifies how the name lookup should be done. Also, **ZIPARCHIVE::FL_UNCHANGED** may be ORed to it to request information about the original file in the archive, ignoring any changes made.

- **ZIPARCHIVE::FL_NOCASE**
- **ZIPARCHIVE::FL_NODIR**
- **ZIPARCHIVE::FL_UNCHANGED**

Return Values

Returns an array containing the entry details or **FALSE** on failure.

Examples

Example #31 - Dump the stat info of an entry

```
<?php
$zip = new ZipArchive;
$res = $zip->open('test.zip')
if ($res === TRUE) {
    print_r($zip->statName('foobar/baz'));
    $zip->close();
} else {
    echo 'failed, code:' . $res;
}
?>
```

The above example will output something similar to:

```
Array
(  
  [name] => foobar/baz  
  [index] => 3  
  [crc] => 499465816  
  [size] => 27  
  [mtime] => 1123164748  
  [comp_size] => 24  
  [comp_method] => 8  
)
```

ZipArchive::unchangeAll

ZipArchive::unchangeAll -- Undo all changes done in the archive.

Description

mixed ZipArchive::unchangeAll (void)

Undo all changes done in the archive.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

ZipArchive::unchangeArchive

ZipArchive::unchangeArchive -- Revert all global changes done in the archive.

Description

mixed ZipArchive::unchangeArchive (void)

Revert all global changes to the archive archive. For now, this only reverts archive comment changes.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

ZipArchive::unchangeIndex

ZipArchive::unchangeIndex -- Revert all changes done to an entry at the given index.

Description

mixed ZipArchive::unchangeIndex (int \$index)

Revert all changes done to an entry at the given index.

Parameters

index
Index of the entry.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

ZipArchive::unchangeName

ZipArchive::unchangeName -- Revert all changes done to an entry with the given name.

Description

mixed ZipArchive::unchangeName (string *\$name*)

Revert all changes done to an entry.

Parameters

name

Name of the entry.

Return Values

Returns **TRUE** on success or **FALSE** on failure.