

g h Goto homepage

g s Goto search
(current page)

/ Focus search box

[zip_close »](#)

[« ZipArchive::unchangeName](#)

- [PHP Manual](#)
- [Function Reference](#)
- [Compression and Archive Extensions](#)
- [Zip](#)

Change language:

[Edit Report a Bug](#)

Zip Functions ¶

Table of Contents ¶

- [zip_close](#) — Close a ZIP file archive
- [zip_entry_close](#) — Close a directory entry
- [zip_entry_compressedsize](#) — Retrieve the compressed size of a directory entry
- [zip_entry_compressionmethod](#) — Retrieve the compression method of a directory entry
- [zip_entry_filesize](#) — Retrieve the actual file size of a directory entry
- [zip_entry_name](#) — Retrieve the name of a directory entry
- [zip_entry_open](#) — Open a directory entry for reading
- [zip_entry_read](#) — Read from an open directory entry
- [zip_open](#) — Open a ZIP file archive
- [zip_read](#) — Read next entry in a ZIP file archive

[+ add a note](#)

User Contributed Notes 20 notes

[up](#)
[down](#)

11

[nielsvandenberge at hotmail dot com ¶](#)

11 years ago

This is the function I use to unzip a file.

It includes the following options:

- * Unzip in any directory you like
- * Unzip in the directory of the zip file
- * Unzip in a directory with the zipfile's name in the directory of the zip file. (i.e.: C:\test.zip will be unzipped in C:\test\)
- * Overwrite existing files or not
- * It creates non existing directories with the function Create_dirs(\$path)

You should use absolute paths with slashes (/) instead of backslashes (\).

I tested it with PHP 5.2.0 with php_zip.dll extension loaded

```
<?php
/**
 * Unzip the source_file in the destination dir
 *
 * @param string    The path to the ZIP-file.
 * @param string    The path where the zipfile should be unpacked, if false the directory of the
 * zip-file is used
 * @param boolean   Indicates if the files will be unpacked in a directory with the name of the
 * zip-file (true) or not (false) (only if the destination directory is set to false!)
 * @param boolean   Overwrite existing files (true) or not (false)
 *
 * @return boolean  Successful or not
 */
function unzip($src_file, $dest_dir=false, $create_zip_name_dir=true, $overwrite=true)
{
    if ($zip = zip_open($src_file))
    {
        if ($zip)
        {
            $splitter = ($create_zip_name_dir === true) ? "." : "/";
            if ($dest_dir === false) $dest_dir = substr($src_file, 0, strrpos($src_file, $splitter)).$splitter;

            // Create the directories to the destination dir if they don't already exist
            create_dirs($dest_dir);

            // For every file in the zip-packet
            while ($zip_entry = zip_read($zip))
            {
                // Now we're going to create the directories in the destination directories

                // If the file is not in the root dir
                $pos_last_slash = strrpos(zip_entry_name($zip_entry), "/");
                if ($pos_last_slash !== false)
                {
                    // Create the directory where the zip-entry should be saved (with a "/" at the end)
                    create_dirs($dest_dir.substr(zip_entry_name($zip_entry), 0, $pos_last_slash+1));
                }

                // Open the entry
                if (zip_entry_open($zip,$zip_entry,"r"))
                {
                    // The name of the file to save on the disk
                    $file_name = $dest_dir.zip_entry_name($zip_entry);

                    // Check if the files should be overwritten or not
                    if ($overwrite === true || $overwrite === false && !is_file($file_name))
                    {
                        // Get the content of the zip entry
                        $fstream = zip_entry_read($zip_entry, zip_entry_filesize($zip_entry));
```

```

        file_put_contents($file_name, $fstream );
        // Set the rights
        chmod($file_name, 0777);
        echo "save: ".$file_name."<br />";
    }

    // Close the entry
    zip_entry_close($zip_entry);
}
}
// Close the zip-file
zip_close($zip);
}
}
else
{
    return false;
}

return true;
}

/**
 * This function creates recursive directories if it doesn't already exist
 *
 * @param String  The path that should be created
 *
 * @return void
 */
function create_dirs($path)
{
    if (!is_dir($path))
    {
        $directory_path = "";
        $directories = explode("/", $path);
        array_pop($directories);

        foreach($directories as $directory)
        {
            $directory_path .= $directory."/";
            if (!is_dir($directory_path))
            {
                mkdir($directory_path);
                chmod($directory_path, 0777);
            }
        }
    }
}

// Extract C:/zipfiletest/zip-file.zip to C:/zipfiletest/zip-file/ and overwrites existing files
unzip("C:/zipfiletest/zip-file.zip", false, true, true);

// Extract C:/zipfiletest/zip-file.zip to C:/another_map/zipfiletest/ and doesn't overwrite existing
files. NOTE: It doesn't create a map with the zip-file-name!

```

```
unzip("C:/zipfiletest/zip-file.zip", "C:/another_map/zipfiletest/", true, false);
```

```
?>
```

[up](#)

[down](#)

7

[yarms at mail dot ru ¶](#)

9 years ago

very short version of function for unzipping files with folders structure:

```
<?php
```

```
function unzip($file){

    $zip=zip_open(realpath(".")."/".$file);
    if(!$zip) {return("Unable to proccess file '{$file}'");}

    $e='';

    while($zip_entry=zip_read($zip)) {
        $zdir=dirname(zip_entry_name($zip_entry));
        $zname=zip_entry_name($zip_entry);

        if(!zip_entry_open($zip,$zip_entry,"r")) {$e.="Unable to proccess file '{$zname}'";continue;}
        if(!is_dir($zdir)) mkdirr($zdir,0777);

        #print "{$zdir} | {$zname} \n";

        $zip_fs=zip_entry_filesize($zip_entry);
        if(empty($zip_fs)) continue;

        $zz=zip_entry_read($zip_entry,$zip_fs);

        $z=fopen($zname,"w");
        fwrite($z,$zz);
        fclose($z);
        zip_entry_close($zip_entry);
    }
    zip_close($zip);

    return($e);
}
```

```
function mkdirr($pn,$mode=null) {

    if(is_dir($pn)||empty($pn)) return true;
    $pn=str_replace(array('/', ' '),DIRECTORY_SEPARATOR,$pn);

    if(is_file($pn)) {trigger_error('mkdirr() File exists', E_USER_WARNING);return false;}

    $next_pathname=substr($pn,0,strrpos($pn,DIRECTORY_SEPARATOR));
    if(mkdirr($next_pathname,$mode)) {if(!file_exists($pn)) {return mkdir($pn,$mode);}} }
    return false;
}
```

```
unzip("test.zip");
```

```
?>
```

have a nice day :)

[up](#)

[down](#)

3

[nheimann at gmx dot net](#)

11 years ago

With this extension you can Add dirs with files with the ZipArchive Object

```
<?php
/**
 * FlxZipArchive, Extends ZipArchiv.
 * Add Dirs with Files and Subdirs.
 *
 * <code>
 * $archive = new FlxZipArchive;
 * // .....
 * $archive->addDir( 'test/blub', 'blub' );
 * </code>
 */
class FlxZipArchive extends ZipArchive {
    /**
     * Add a Dir with Files and Subdirs to the archive
     *
     * @param string $location Real Location
     * @param string $name Name in Archive
     * @author Nicolas Heimann
     * @access private
     */

    public function addDir($location, $name) {
        $this->addEmptyDir($name);

        $this->addDirDo($location, $name);
    } // EO addDir;

    /**
     * Add Files & Dirs to archive.
     *
     * @param string $location Real Location
     * @param string $name Name in Archive
     * @author Nicolas Heimann
     * @access private
     */

    private function addDirDo($location, $name) {
        $name .= '/';
        $location .= '/';

        // Read all Files in Dir
```