rar

rar://

RAR

ENTITY

The wrapper takes the url encoded path to the RAR archive (relative or absolute), an optional asterisk (`*`), an optional number sign (`#`) and an optional url encoded entry name, as stored in the archive. Specifying an entry name requires the number sign; a leading forward slash in the entry name is optional.

This wrapper can open both files and directories. When opening directories, the asterisk sign forces the directory entries names to be returned unencoded. If it's not specified, they will be returned url encoded – the reason for this is to allow the wrapper to be correctly used with built-in functionality like the `RecursiveDirectoryIterator` in the presence of file names that seem like url encoded data.

If the pound sign and the entry name part are not included, the root of the archive will be displayed. This differs from regular directories in that the resulting stream will not contain information such as the modification time, as the root directory is not stored in an individual entry in the archive. The usage of the wrapper with `RecursiveDirectoryIterator` requires the number sign to be included in the URL when accessing the root, so that the URLs of the children may be constructed correctly.

> In order to use the `rar://` wrapper, the rar extension available from ENTITY must be installed.

`rar://` Available since PECL rar 3.0.0

ENTITY

- `rar://ENTITYurl encoded archive nameENTITY[*][#[ENTITYurl encoded entry nameENTITY]]`

ENTITY

Attribute	Supported
Restricted by allow_url_fopen	No
Restricted by allow_url_include	No
Allows Reading	Yes
Allows Writing	No
Allows Appending	No
Allows Simultaneous Reading and Writing	No
Supports `stat`	Yes
Supports `unlink`	No
Supports `rename`	No
Supports `mkdir`	No
Supports `rmdir`	No

Wrapper Summary {role="stream_wrapper"}

Name	Usage	Default
`open_password`	The password used to encrypt the headers of the archive, if any. WinRAR will encrypt all the files with the same password as the headers password when the later is present, so for archives with encrypted headers, `file_password` will be ignored.	
`file_password`	The password used to encrypt a file, if any. If the headers are also encrypted, this option will be ignored in favor of `open_password`. The reason there are two options is to cover the possibility of supporting archives with different headers and file passwords, should those archives arise. Note that if the archive does not have its headers encrypted, `open_password` will be ignored and this option must be used instead.	
`volume_callback`	A callback to determine the path of missing volumes. See RarArchive::open for more information.	

Context options {role="stream_wrapper"}

ENTITY

Traversing a RAR archive

```
<?php

class MyRecDirIt extends RecursiveDirectoryIterator {
    function current() {
        return rawurldecode($this->getSubPathName()) .
            (is_dir(parent::current())?" [DIR]":"");
    }
}

$f = "rar://" . rawurlencode(dirname(__FILE__)) .
    DIRECTORY_SEPARATOR . 'dirs_and_extra_headers.rar#';

$it = new RecursiveTreeIterator(new MyRecDirIt($f));

foreach ($it as $s) {
    echo $s, "\n";
}
?>
```

ENTITY

```
|-allow_everyone_ni [DIR]
|-file1.txt
|-file2_□.txt
|-with_streams.txt
\-□ [DIR]
  |-□\%2Fempty%2E [DIR]
  | \-□\%2Fempty%2E\file7.txt
  |-□\empty [DIR]
  |-□\file3.txt
  |-□\file4_□.txt
  \-□\□_2 [DIR]
    |-□\□_2\file5.txt
    \-□\□_2\file6_□.txt
```

Opening an encrypted file (header encryption)

```
<?php
$stream = fopen("rar://" .
    rawurlencode(dirname(__FILE__)) . DIRECTORY_SEPARATOR .
    'encrypted_headers.rar' . '#encfile1.txt', "r", false,
    stream_context_create(
        array(
            'rar' =>
                array(
                    'open_password' => 'samplepassword'
                )
            )
        )
    );
var_dump(stream_get_contents($stream));
/* creation and last access date is opt-in in WinRAR, hence most
 * files don't have them */
var_dump(fstat($stream));
?>
```

ENTITY

```
string(26) "Encrypted file 1 contents."
Array
(
    [0] => 0
    [1] => 0
    [2] => 33206
    [3] => 1
    [4] => 0
    [5] => 0
    [6] => 0
    [7] => 26
    [8] => 0
    [9] => 1259550052
    [10] => 0
    [11] => -1
```

```
    [12] => -1
    [dev] => 0
    [ino] => 0
    [mode] => 33206
    [nlink] => 1
    [uid] => 0
    [gid] => 0
    [rdev] => 0
    [size] => 26
    [atime] => 0
    [mtime] => 1259550052
    [ctime] => 0
    [blksize] => -1
    [blocks] => -1
)
```

□□□ #2
□ □□□ □□□ 2026-10-31 17:48:47 UTC
□ □□□ □□□ 2026-10-31 17:48:48 UTC