

# cFileHandler

The cFileHandler class encapsulates functionalities for dealing with files.

## Methods:

All provided functionalities are in a public static context.

methodName	Description	Example
chmod(\$filename, \$mode)	<p>This method can be used to change file permissions on a *nix system.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• \$filename: the full path name of a file (including the filename)</li><li>• \$mode: permission you want to set in octal mode</li></ul>	<pre>chmod( '/var/www/index.php', 777 );</pre> <p>The call of the method above grants read, write and execute rights.</p>
copy(\$filename, \$destination)	<p>This method copies a file to a given location. Existing files will be overwritten</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• \$filename: the full path name of a file (including the filename)</li><li>• \$destination: the full path name of a file (including the filename)</li></ul> <p><b>Returns:</b> bool</p>	<pre>copy( '/var/www/index.php', '/var/www/index2.php' );</pre> <p>The call of the method above copies the index.php file to the index2.php file.</p>
exists(\$filename)	<p>This method checks if a file exists at the given location</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• \$filename: the full path name of a file (including the filename)</li></ul> <p><b>Returns:</b> bool</p>	<pre>exists( '/var/www/index.php' );</pre> <p>The call of the method above returns true if the 'var/www/index.php' file exists, otherwise false.</p>
fileNameIsDot(\$fileName)	<p>This method checks if the given filename is '.' or '..'.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• \$fileName: the full path name of a file (including the filename)</li></ul> <p><b>Returns:</b> bool</p>	<pre>fileNameIsDot( '.' );</pre> <p>The call of the method above returns true.</p>
getExtension(\$basename)	<p>This method returns the file extension of the given file</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• \$filename: the full path name of a file (including the filename)</li></ul> <p><b>Returns:</b> string</p>	<pre>getExtension( '/var/www/index.php' );</pre> <p>The call of the method above returns the string 'php'.</p>

<p>info(\$filename)</p>	<p>This method returns information about the given file. The returning array includes the following information:</p> <ul style="list-style-type: none"> <li>• size (byte)</li> <li>• atime (unix timestamp: last access)</li> <li>• ctime (unix timestamp: creation)</li> <li>• mtime (unix timestamp: last modified)</li> <li>• perms (4 octal digit: file permissions)</li> <li>• extension (string: mime type of the file)</li> </ul> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> array</p>	
<p>isDirectoryEmpty(\$dir)</p>	<p>This method was moved to cDirHandler class with version 4.9.3 and is deprecated now.</p>	<p>see <a href="#">cDirHandler</a></p>
<p>move(\$filename, \$destination)</p>	<p>This method changes the location of the given file</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> <li>• \$destination: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>move('/var/www/index.php', '/var/www/test/index.php');</pre> <p>The call of the method above moves the index.php file from path '/var/www/' to '/var/www/test/'.</p>
<p>read(\$filename, \$length = 0, \$offset = 0, \$reverse = false)</p>	<p>This method reads bytes from the given file.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Optional parameters:</b></p> <ul style="list-style-type: none"> <li>• \$length (int: number of bytes to read)</li> <li>• \$offset (int: byte offset)</li> <li>• \$reverse (bool: start reading from the front or from the end of the file)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>read('/var/www/index.php');</pre> <p>The call of the method above returns the content of the index.php file.</p>
<p>readable(\$filename)</p>	<p>This method checks if the given file is readable.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>readable('/var/www/index.php');</pre> <p>The call of the method above moves returns true if the index.php file is readable, otherwise false.</p>
<p>readLine(\$filename, \$lines = 0, \$lineoffset = 0)</p>	<p>This method reads a line from the given file.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Optional parameters:</b></p> <ul style="list-style-type: none"> <li>• \$lines(int: number of lines to read)</li> <li>• \$lineoffset (int: lines offset)</li> </ul> <p><b>Returns:</b></p> <ul style="list-style-type: none"> <li>• string: read only one line</li> <li>• array of strings: read more than one line</li> <li>• bool: read nothing</li> </ul>	<pre>readableLine('/var/www/index.php');</pre> <p>The call of the method above returns the first line of the content from the index.php file.</p>
<p>recursiveCopy(\$filename, \$destination)</p>	<p>This method was moved to cDirHandler class with version 4.9.3 and is deprecated now.</p>	<p>see <a href="#">cDirHandler</a></p>

recursiveRmdir(\$dirname)	This method was moved to cDirHandler class with version 4.9.3 and is deprecated now.	see <a href="#">cDirHandler</a>
remove(\$filename)	<p>This method removes the given file from the filesystem.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>remove( '/var/www/index.php' );</pre> <p>The call of the method above removes the index.php file.</p>
rename(\$filename, \$new_filename)	<p>This method renames the given file from the filesystem.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> <li>• \$new_filename: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>rename( '/var/www/index.php', '/var/www/index.html' );</pre> <p>The call of the method above renames the index.php file to index.html.</p>
setDefaultDirPerms(\$pathname)	This method was moved to cDirHandler class with version 4.9.3 and is deprecated now.	see <a href="#">cDirHandler</a>
setDefaultFilePerms(\$filename)	<p>This method sets default permissions to the given file</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> bool</p>	
truncate(\$filename)	<p>This method clears the content of the given file</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>truncate( '/var/www/index.php' );</pre> <p>The call of the method above removes the content of the index.php file.</p>
validateFilename(\$filename, \$notifyAndExitOnFailure = true)	<p>This method validates the filename of the given file with CONENTIDO standard.</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> </ul> <p><b>Optional parameters:</b></p> <ul style="list-style-type: none"> <li>• \$notifyAndExitOnFailure(bool: show notification and exit the script)</li> </ul> <p><b>Returns:</b> bool</p>	
write(\$filename, \$content, \$append = false)	<p>This method writes data to the given file</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"> <li>• \$filename: the full path name of a file (including the filename)</li> <li>• \$content: content you want to write</li> </ul> <p><b>Optional parameters:</b></p> <ul style="list-style-type: none"> <li>• \$append(bool: flag to append the content to a existing file)</li> </ul> <p><b>Returns:</b> bool</p>	<pre>write( '/var/www/index.php', 'hello world!' );</pre> <p>The call of the method above writes the string 'hello world!' in the index.php file.</p>

<code>writeable(\$filename)</code>	<p>This method checks if the given file is writeable</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• <code>\$filename</code>: the full path name of a file (including the filename)</li></ul> <p><b>Returns:</b> bool</p>	<pre>writeable('/var/www/index.php');</pre> <p>The call of the method above returns true if the index.php file is writeable, otherwise false.</p>
<code>writeLine(\$filename, \$content, \$append = false)</code>	<p>This method writes a line to the given file</p> <p><b>Expected parameters:</b></p> <ul style="list-style-type: none"><li>• <code>\$filename</code>: the full path name of a file (including the filename)</li><li>• <code>\$content</code>: content you want to write</li></ul> <p><b>Optional parameters:</b></p> <ul style="list-style-type: none"><li>• <code>\$append</code>(bool: flag to append the content to a existing file)</li></ul> <p><b>Returns:</b> bool</p>	<pre>writeLine('/var/www/index.php', 'hello world!');</pre> <p>The call of the method above returns writes the line 'hello world!' to the index.php file.</p>