

**xattr**

# Introduction

The xattr extension allows for the manipulation of extended attributes on a filesystem.

# Installing/Configuring

## Requirements

To use xattr, you will need libattr installed. It is available at  
» <http://oss.sgi.com/projects/xfs/>.

### Note

These functions only work on filesystems that support extended attributes, and have them enabled at mount time. Some common filesystems that support extended attributes are ext2, ext3, reiserfs, jfs, and xfs.

## Installation

xattr is currently available through PECL » <http://pecl.php.net/package/xattr>.

If » [PEAR](#) is available on your \*nix-like system you can use the pear installer to install the xattr extension, by the following command: `pear -v install xattr`.

You can always download the tar.gz package and install xattr by hand:

### Example #1 - xattr install by hand

```
gunzip xattr-xxx.tgz
tar -xvf xattr-xxx.tar
cd xattr-xxx
phpize
./configure && make && make install
```

## Runtime Configuration

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

## Resource Types

This extension has no resource types defined.

# 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.

**XATTR\_ROOT** ( [integer](#) )

Set attribute in root (trusted) namespace. Requires root privileges.

**XATTR\_DONTFOLLOW** ( [integer](#) )

Do not follow the symbolic link but operate on symbolic link itself.

**XATTR\_CREATE** ( [integer](#) )

Function will fail if extended attribute already exists.

**XATTR\_REPLACE** ( [integer](#) )

Function will fail if extended attribute doesn't exist.

# xattr Functions

# xattr\_get

xattr\_get -- Get an extended attribute

## Description

string **xattr\_get** ( string *\$filename*, string *\$name* [, int *\$flags* ] )

This function gets the value of an extended attribute of a file.

Extended attributes have two different namespaces: user and root. The user namespace is available to all users, while the root namespace is available only to users with root privileges. xattr operates on the user namespace by default, but this can be changed with the *flags* parameter.

## Parameters

*filename*

The file from which we get the attribute.

*name*

The name of the attribute.

*flags*

### Supported xattr flags

<b>XATTR_DONTFOLLOW</b>	Do not follow the symbolic link but operate on symbolic link itself.
<b>XATTR_ROOT</b>	Set attribute in root (trusted) namespace. Requires root privileges.

## Return Values

Returns a string containing the value or **FALSE** if the attribute doesn't exist.

## Examples

## Example #2 - Checks if system administrator has signed the file

```
<?php
$file = '/usr/local/sbin/some_binary';
$signature = xattr_get($file, 'Root signature', XATTR_ROOT);

/* ... check if $signature is valid ... */

?>
```

## See Also

- [xattr\\_list\(\)](#)
- [xattr\\_set\(\)](#)
- [xattr\\_remove\(\)](#)

# xattr\_list

xattr\_list -- Get a list of extended attributes

## Description

array **xattr\_list** ( string *\$filename* [, int *\$flags* ] )

This functions gets a list of names of extended attributes of a file.

Extended attributes have two different namespaces: user and root. The user namespace is available to all users, while the root namespace is available only to users with root privileges. xattr operates on the user namespace by default, but this can be changed with the *flags* parameter.

## Parameters

*filename*  
The path of the file.

*flags*

### Supported xattr flags

<b>XATTR_DONTFOLLOW</b>	Do not follow the symbolic link but operate on symbolic link itself.
<b>XATTR_ROOT</b>	Set attribute in root (trusted) namespace. Requires root privileges.

## Return Values

This function returns an array with names of extended attributes.

## Examples

Example #3 - Prints names of all extended attributes of file
<pre>&lt;?php \$file = 'some_file'; \$root_attributes = xattr_list(\$file, XATTR_ROOT); \$user_attributes = xattr_list(\$file);</pre>



```
echo "Root attributes: \n";
foreach ($root_attributes as $attr_name) {
    printf("%s\n", $attr_name);
}

echo "\n User attributes: \n";
foreach ($attributes as $attr_name) {
    printf("%s\n", $attr_name);
}

?>
```

## See Also

- [xattr\\_get\(\)](#)

# xattr\_remove

xattr\_remove -- Remove an extended attribute

## Description

bool **xattr\_remove** ( string \$filename, string \$name [, int \$flags ] )

This function removes an extended attribute of a file.

Extended attributes have two different namespaces: user and root. The user namespace is available to all users, while the root namespace is available only to users with root privileges. xattr operates on the user namespace by default, but this can be changed with the *flags* parameter.

## Parameters

*filename*  
The file from which we remove the attribute.

*name*  
The name of the attribute to remove.

*flags*

### Supported xattr flags

<b>XATTR_DONTFOLLOW</b>	Do not follow the symbolic link but operate on symbolic link itself.
<b>XATTR_ROOT</b>	Set attribute in root (trusted) namespace. Requires root privileges.

## Return Values

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

## Examples

#### Example #4 - Removes all extended attributes of a file

```
<?php
$file = 'some_file';
$attributes = xattr_list($file);

foreach ($attributes as $attr_name) {
    xattr_remove($file, $attr_name);
}
?>
```

## See Also

- [xattr\\_list\(\)](#)
- [xattr\\_set\(\)](#)
- [xattr\\_get\(\)](#)

# xattr\_set

xattr\_set -- Set an extended attribute

## Description

bool **xattr\_set** ( string \$filename, string \$name, string \$value [, int \$flags ] )

This function sets the value of an extended attribute of a file.

Extended attributes have two different namespaces: user and root. The user namespace is available to all users, while the root namespace is available only to users with root privileges. xattr operates on the user namespace by default, but this can be changed with the *flags* parameter.

## Parameters

*filename*  
The file in which we set the attribute.

*name*  
The name of the extended attribute. This attribute will be created if it doesn't exist or replaced otherwise. You can change this behaviour by using the *flags* parameter.

*value*  
The value of the attribute.

*flags*

### Supported xattr flags

<b>XATTR_CREATE</b>	Function will fail if extended attribute already exists.
<b>XATTR_REPLACE</b>	Function will fail if extended attribute doesn't exist.
<b>XATTR_DONTFOLLOW</b>	Do not follow the symbolic link but operate on symbolic link itself.
<b>XATTR_ROOT</b>	Set attribute in root (trusted) namespace. Requires root privileges.

## Return Values

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

## Examples

### Example #5 - Sets extended attributes on .wav file

```
<?php
$file = 'my_favourite_song.wav';
xattr_set($file, 'Artist', 'Someone');
xattr_set($file, 'My ranking', 'Good');
xattr_set($file, 'Listen count', '34');

/* ... other code ... */

printf("You've played this song %d times", xattr_get($file, 'Listen count'));
?>
```

## See Also

- [xattr\\_get\(\)](#)
- [xattr\\_remove\(\)](#)

# xattr\_supported

xattr\_supported -- Check if filesystem supports extended attributes

## Description

bool **xattr\_supported** ( string \$filename [, int \$flags ] )

This functions checks if the filesystem holding the given file supports extended attributes. Read access to the file is required.

## Parameters

*filename*

The path of the tested file.

*flags*

### Supported xattr flags

<b>XATTR_DONTFOLLOW</b>	Do not follow the symbolic link but operate on symbolic link itself.
-------------------------	--

## Return Values

This function returns **TRUE** if filesystem supports extended attributes, **FALSE** if it doesn't and **NULL** if it can't be determined (for example wrong path or lack of permissions to file).

## Examples

### Example #6 - [xattr\\_supported\(\)](#) example

The following code checks if we can use extended attributes.

```
<?php
$file = 'some_file';

if (xattr_supported($file)) {
    /* ... make use of some xattr_* functions ... */
}

?>
```

## See Also

- [xattr\\_get\(\)](#)
- [xattr\\_list\(\)](#)