

**Cracklib**

**Crack**

# Introduction

These functions allow you to use the CrackLib library to test the 'strength' of a password. The 'strength' of a password is tested by that checks length, use of upper and lower case and checked against the specified CrackLib dictionary. CrackLib will also give helpful diagnostic messages that will help 'strengthen' the password.

<b>Note</b>
This extension has been moved to the <a href="#">» PECL</a> repository and is no longer bundled with PHP as of PHP 5.0.0.

# Installing/Configuring

## Requirements

More information regarding CrackLib along with the library can be found at » <http://sourceforge.net/projects/cracklib>.

## Installation

This » [PECL](#) extension is not bundled with PHP. 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/crack>.

In PHP 4 this PECL extensions source can be found in the *ext/* directory within the PHP source or at the PECL link above. In order to use these functions you must compile PHP with Crack support by using the `--with-crack[=DIR]` configuration option.

Windows users will enable *php\_crack.dll* inside of *php.ini* in order to use these functions. In PHP 4 this DLL resides in the *extensions/* directory within the PHP Windows binaries download. The DLL for this PECL extension may be downloaded from either the » [PHP Downloads](#) page or from » <http://pecl4win.php.net/>

## Runtime Configuration

The behaviour of these functions is affected by settings in *php.ini*.

### Crack configuration options

Name	Default	Changeable	Changelog
crack.default_dictionary	NULL	PHP_INI_PERDIR	PHP_INI_SYSTEM in crack <= 0.2. Available since PHP 4.0.5. Removed in PHP 5.0.0.

For further details and definitions of the `PHP_INI_*` constants, see the [php.ini directives](#).

## Resource Types

The CrackLib extension defines a dictionary resource identifier returned by [crack\\_opendict\(\)](#).

# Predefined Constants

This extension has no constants defined.

# Examples

This example shows how to open a CrackLib dictionary, test a given password, retrieve any diagnostic messages, and close the dictionary.

## Example #1 - CrackLib example

```
<?php
// Open CrackLib Dictionary
$dictionary = crack_opendict('/usr/local/lib/pw_dict')
    or die('Unable to open CrackLib dictionary');

// Perform password check
$check = crack_check($dictionary, 'gx9A2s0x');

// Retrieve messages
$diag = crack_getlastmessage();
echo $diag; // 'strong password'

// Close dictionary
crack_closedict($dictionary);
?>
```

## Note

If [crack\\_check\(\)](#) returns **TRUE**, [crack\\_getlastmessage\(\)](#) will return 'strong password'.

# Crack Functions

# crack\_check

crack\_check -- Performs an obscure check with the given password

## Description

bool **crack\_check** ( resource \$dictionary, string \$password )

bool **crack\_check** ( string \$password )

Performs an obscure check with the given password on the specified dictionary.

Warning
This function is <i>EXPERIMENTAL</i> . The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Parameters

*dictionary*

The crack lib dictionary. If not specified, the last opened dictionary is used.

*password*

The tested password.

## Return Values

Returns **TRUE** if *password* is strong, or **FALSE** otherwise.



# crack\_closedict

crack\_closedict -- Closes an open CrackLib dictionary

## Description

bool **crack\_closedict** ( [ resource \$dictionary ] )

[crack\\_closedict\(\)](#) closes the specified *dictionary* identifier.

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Parameters

*dictionary*

The dictionary to close. If not specified, the current directory is closed.

## Return Values

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

## See Also

- [crack\\_owendict\(\)](#)

# crack\_getlastmessage

crack\_getlastmessage -- Returns the message from the last obscure check

## Description

string **crack\_getlastmessage** ( void )

[crack\\_getlastmessage\(\)](#) returns the message from the last obscure check.

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

## Return Values

The message from the last obscure check or **FALSE** if there was no obscure checks made so far.

The returned message is one of:

- *it's WAY too short*
- *it is too short*
- *it does not contain enough DIFFERENT characters*
- *it is all whitespace*
- *it is too simplistic/systematic*
- *it looks like a National Insurance number.*
- *it is based on a dictionary word*
- *it is based on a (reversed) dictionary word*
- *strong password*

## See Also

- [crack\\_check\(\)](#)

# crack\_opendict

crack\_opendict -- Opens a new CrackLib dictionary

## Description

resource **crack\_opendict** ( string \$dictionary )

[crack\\_opendict\(\)](#) opens the specified CrackLib *dictionary* for use with [crack\\_check\(\)](#).

### Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

### Note

Only one dictionary may be open at a time.

## Parameters

*dictionary*

The path to the Cracklib dictionary.

## Return Values

Returns a dictionary resource identifier on success, or **FALSE** on failure.

## See Also

- [crack\\_check\(\)](#)
- [crack\\_closedict\(\)](#)