

Zend Optimizer™

User Guide:

Zend Optimizer™



By Zend Technologies Inc.

Disclaimer

The information in this document is subject to change without notice and does not represent a commitment on the part of Zend Technologies Ltd. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the written permission of Zend Technologies Ltd.

All trademarks mentioned in this document, belong to their respective owners.

© 1999-2006 Zend Technologies Ltd. All rights reserved.

Zend Optimizer™ User Guide issued April 2006.

Product Version: Zend Optimizer 3.0

DN: OUG-T-290306-3.0-004

Table of Contents

Zend Optimizer	1
Introduction	1
Overview	1
System Requirements	2
Installing the Zend Optimizer	3
Uninstall.....	4
Installation Verification	4
Troubleshooting	5
Zend Optimizer Settings.....	8
Zend Optimizer PHP API Functions	9
Getting Additional Information	11
Zend Optimizer Technical FAQ.....	12

Zend Optimizer

Introduction

The Zend Optimizer™ boosts PHP performance by going over the intermediate code generated by the standard Zend run-time compiler and optimizing it for faster execution. In addition, it runs the files encoded by the Zend Guard, while enhancing the running speed of PHP applications.

The Zend Optimizer is available **free** of charge, from the Zend Store, <http://www.zend.com/store/products/zend-optimizer.php>

Overview

The Zend Optimizer is loaded as a PHP engine extension. This extension, automatically optimizes PHP scripts, and transparently detects and loads encoded files.

The Zend Optimizer User Guide is intended for supplying information about installing and operating Zend Optimizer in the following sections:

- System Requirements
- Installation
- Troubleshooting
- Additional Information

System Requirements

In order to run properly, the following requirements have to be met:

1. Zend Optimizer is installed on a supported operating system.
2. Zend Optimizer is installed on a supported version of PHP 4 or PHP 5.
3. A web server is installed on the machine on which the PHP is running.

A complete list of system requirements is constantly kept up-to-date with new versions, updates and changes in the Zend optimizer's System Requirements section on zend.com: <http://www.zend.com/store/products/optimizer-sysreq.php>

Installing the Zend Optimizer

This section describes the Zend Optimizer installation procedures for all-compatible Operating Systems.

Note:

After installing the Zend Optimizer, check whether the installation was successfully completed (see Checking for Successful Installation.)

The Zend Optimizer is a free package that can be downloaded from zend.com. To download and install the Zend Optimizer:

1. Log in or register with your Zend account.
The Zend Optimizer – Free Download page opens.
2. Select the Optimizer version that corresponds with the platform system you use and select “Download Now”.
The File Download dialog opens.
3. Choose the Save option and save the selected file to your local system.

From this point the installation instructions differ according to the operating system on which you would like to install the Zend Optimizer

To install the Zend Optimizer on Windows:

1. Double click on the package ZendOptimizer-x.x.x-windows-i386.exe
The Optimizer Install Shield opens.
2. Follow the instructions in the Installer.
These instructions define the basic system configurations necessary for installing the Zend Optimizer such as the Web Server in use and the location of the php.ini.

You can now jump to the section: Installation Verification, page 4 for instructions on how to confirm the Zend Optimizer was successfully installed.

To install the Zend Optimizer on Unix Variants:

1. Open a terminal window and change dir to the directory in which the Zend Optimizer package was saved.
To untar the package use the command:
`gunzip -c <package name> | tar xf -`
2. From the extracted package's directory, type the following command: `./install`.
The installer is launched
3. Follow the instructions in the installer.
These instructions define the basic system configurations necessary for installing the Zend Optimizer, such as the Web Server in use and the location of the php.ini.

4. If for some reason, the graphical installation will not start or is illegible, use the command: `./install-tty` which is useful on low-end terminals such as when installing on a remote server via SSH/Telnet.".
The text only installation starts.

Important note:

To install or uninstall Zend Optimizer, you must be logged in with administrator/root permissions.

Uninstall

Windows

In order to uninstall Optimizer:

1. Double click on the uninstaller icon, which is located at Control Panel -> Add/Remove Programs and select Zend Optimizer.

Unix Variants

In order to uninstall the Optimizer:

1. Replace the `php.ini` file with the backed up copy (`php.ini-zend_optimizer.bak`).
2. Delete the Optimizer installation directory.
3. Restart the Web Server.

Important note:

If you have other Zend products installed under the same directory, do not delete the installation directory. In this case, you should only change the `php.ini` file.

Installation Verification

This program makes use of the Zend scripting language engine.

Use the PHP function: `phpinfo()` to verify a successful installation, as shown in the following script:

```
<?php
    phpinfo();
?>
```

The part of the output where the Zend Optimizer is listed looks similar to this:

```
Zend Engine vX.Y.Z, copyright (c) 1998-2006, Zend Technologies
Ltd.,
with the Zend Optimizer vX.Y.Z, copyright (c) 1998-2006 Zend
Technologies Ltd.
```

Troubleshooting

This section discusses the possible incompatibilities that may arise and how they can be prevented/resolved in order to properly activate and use the Zend Optimizer.

Limiting Number of Concurrent Licenses and “Cookies”

When the “Require Cookies” option is **NOT** selected, any page can be loaded, once, from *any number* of browsers concurrently even when the number of concurrent users is limited (even to a single user).

*This happens because – the first time that the page is accessed – no cookie is present. The second time that the page is accessed, a cookie **is** present (even though not required). The server then limits access.*

In addition, if the “Require Cookies” option is NOT selected, AND the browser is set to not receive cookies, any page can be loaded, any number of times, from any number of browsers.

This happens because no cookie is present at any time. The server has no way to know that the page has been accessed.

PHP Version Incompatibility

The Zend Optimizer must be fully compatible with the PHP version installed. An incompatibility will cause the Zend Optimizer to fail and register an error message. (See message in the Web server log file). Make sure the file you downloaded corresponds with the installed PHP version. If necessary, uninstall the Zend Optimizer and download/install the version suitable for the PHP version in use.

Note:

Zend Optimizer 2.6 supports all recent PHP 4 and PHP 5 versions, not including non-released versions and old PHP versions (like 4.0.x).

Web Server Incompatibility

The Zend Optimizer must be fully compatible with the installed Web Server. An incompatibility will cause the Zend Optimizer to fail and write an error message to the log. Make sure the file you selected the proper web server in the installation process. If necessary, uninstall the Zend Optimizer and download/install the proper version.

Note:

The Zend Optimizer only runs with supported Web Servers.

Configuration Options

The Zend Optimizer will not run if the following PHP settings are enabled:

- **Debug Mode** - Check that your PHP was compiled in non-debug mode - This can be checked by looking at the top part of the `phpinfo()` function output and checking "Debug Build" row (ZEND_DEBUG in older PHP versions) - it should say "no".

- **Thread Safety** - Check that your PHP was compiled in ZTS mode for Windows and non-ZTS mode for Unix. This can be checked by looking at the top part of the `phpinfo()` function and checking "Thread Safety" row - it should say "disabled" for Unix and "enabled" for Windows.
- **Enable Versioning** - Check that your PHP was compiled without the Enable Versioning setting. This can be checked by looking at the top part of the `phpinfo()` function and checking to see if a row called "--enable-versioning" exists in the Configure Command section - if it does exist the PHP should be compiled without this option.

Other PHP Extension Incompatibilities

Another reason for the Zend Optimizer not running could be due to extension incompatibility. This problem arises when the `php.ini` includes extensions from different sources that interfere or clash with the Zend Optimizer. As a general rule, any extension that is related to debugging or optimizing that has been added to the `php.ini` by a third party supplier, has the potential to disrupt the Zend Optimizer's performance and should be disabled/removed.

Zend Guard Incompatibility

Files encoded with the Zend Guard must be compiled with compatible version of Zend Guard. When in doubt always use the latest version of Zend Optimizer. If the versions of Zend Guard and Zend Optimizer are incompatible, the Optimizer will fail to run encoded files and will register an error message (Zend Optimizer not installed or General Optimizer error).

Note:

If you are using Zend Guard 4.0 and above, use the latest Zend Optimizer version.

Zend Optimizer Settings

As mentioned earlier the Zend Optimizer goes over the intermediate code generated by the standard Zend run-time compiler and optimizes it for faster execution. In order to customize the Optimizers performance, the following directives and extensions have been provided.

Configure Optimizer Settings

Directive	Description
zend_optimizer.enable_loader	If you do not plan to use the Zend Optimizer to load encoded files, you can slightly improve the Optimizer's performance by adding the zend_optimizer.enable_loader = 0. This disables the transparent auto-loading mechanism that is built into the Zend Optimizer.
zend_optimizer.disable_licensing	If you do not need to use any licensing features, you can disable the Zend Optimizer license request. Setting this option lowers Optimizer memory usage and slightly enhances performance
zend_optimizer.licence_path	A license file is required to load encoded PHP scripts that were encoded with a "license" option. If you turn off this option, the encoded scripts on your server that require a license may not load. This directive is a list of directories or files, separated by path separator (; on windows, : on Unix) that gives the Optimizer a place to look for encoded file licenses. When directories are added to this path, all files ending in .zl will be loaded as licenses. If the file that was added to this path, was loaded as a license. Licenses are loaded on server start time (unless loaded by PHP API), so if you add a license or change a path, you should restart the server.
zend_optimizer.obfuscation_level_support	The Obfuscation level supported by Optimizer.

Located in the zend.ini, these directives can be easily configured through Zend Platform and Zend Core (all flavors).

Note:

If you are using the Zend Optimizer along with Zend Guard to encode files, please refer to the Zend Guard User Guide for more information.

Zend Optimizer PHP API Functions

The following table lists the PHP API functions that can be used with the Zend Optimizer.

Function	Syntax	Description
zend_loader_enabled	bool zend_loader_enabled()	Shown if loader is enabled.
zend_loader_file_encoded	bool zend_loader_file_encoded()	Returns true if the current file is a Zend-encoded file.
zend_loader_file_licensed	array zend_loader_file_licensed()	Returns license (array with fields) if the current file has a valid license and is encoded, otherwise it returns false.
zend_loader_current_file	string zend_loader_current_file()	Returns the name of the file currently being executed.
zend_loader_install_license	bool zend_loader_install_license(string license_file[, bool override])	Dynamically loads a license for applications encoded with ZendGuard. The Override controls if it will override old licenses for the same product.
obfuscate_function_name	string obfuscate_function_name(string function_name)	Obfuscate and return the given function name with the internal obfuscation function.
current_obfuscation_level	int current_obfuscation_level()	Returns the current run-time obfuscation level.
Note:		
To insure backward compatability with scripts obfuscated with the Zend Guard at level "Basic", their run-time obfuscation level will remain 0 !		
zend_runtime_obfuscate	bool zend_runtime_obfuscate()	Start runtime-obfuscation support that allows limited mixing of obfuscated and un-obfuscated code.
zend_get_id	array zend_get_id([bool all_ids = false])	Returns array of the host ids. If all_ids is true, then all IDs are returned, otherwise only IDs considered "primary" are returned.

Getting Additional Information

There are several sources of information that Zend Optimizer users can in order to find answers to questions that may be raised.

Answers to technical and operating system related questions can be found in the Zend Knowledge Base. Here you will find a repository of articles that address know issues and how to deal with them:

<http://www.zend.com/support/knowledgebase.php>

Otherwise, you can access one of the Zend forums:

<http://www.zend.com/support/>

For issues other than technical support, email us at: support@zend.com

Zend Optimizer Technical FAQ

What is the Zend Optimizer? What does it do?

The Zend Optimizer goes over the intermediate code generated by the standard Zend run-time compiler and optimizes it for faster execution. In addition, the Zend Optimizer enables PHP to load and execute files encoded by the ZendGuard™.

How do I get the Zend Optimizer?

The Zend Optimizer is available for download free of charge from the [Zend Store](http://www.zend.com/store/), at <http://www.zend.com/store/>.

I have installed the Zend Optimizer. How do I start it?

The Zend Optimizer is a passive module that runs within the framework of PHP. It optimizes scripts automatically, and transparently detects and loads encoded files, therefore there is no need for you to start it.

How many optimizations does the Zend Optimizer perform altogether?

The Zend Optimizer is capable of performing around 20 different optimizations, some of which are quite complex. One of the simpler optimizations that the Zend Optimizer does is to change post-incrementing to pre-incrementing, where possible, since pre-incrementing is the faster operation of the two.

Is there any way to further improve the performance of the Optimizer?

If you don't plan to use the Zend Optimizer in order to load encoded files, you can slightly improve the performance of the Optimizer by adding the `zend_optimizer.enable_loader = 0` directive to `php.ini`. This disables the transparent auto-loading mechanism that is built into the Zend Optimizer.

Can encoded and non-encoded (source) PHP files be used together?

Encoded and non-encoded PHP files can be used together transparently.

Will using encoded files (instead of source files) change run-time speed or file size?

There might be some speed gain because the compilation stage is saved on every run of the script, however, speed improvements, if any, are dependent on the nature of the script. The size of an encoded file might be somewhat smaller or larger than the source file, but this too is dependent on the nature of the script. The factors that tend to improve run-time speed are not necessarily the same as those that tend to decrease file size.

When I start my server, the Zend Optimizer gives error messages and fails: Why?

Your version of PHP may be incompatible with the Zend Optimizer that you installed. Make sure that you download a version of the Zend Optimizer that is compatible with your version of PHP.

When I try to run an encoded file, error messages are displayed: Why?

The most common cause of error messages and failure is incompatibility either with the PHP version, or with the Zend Intermediate Code file (that is, with the version of the Zend Guard that encoded the file).

Another possible cause for failure could be encoded file corruption caused by transferring the file between systems. This happens because even though encoded

files are system-independent, certain tools for transferring files from Windows to Unix and vice-versa change the contents of the files so that they appear to be text files. To prevent this from happening, always use "binary transfer mode" when transferring encoded files with file transfer tools or ensure integrity of the files by using a package such as: zip or tar.gz.

For additional information, see the "Avoiding Incompatibilities" section in the Zend Guard User Guide.