Version 3.1.0

Table of Contents

Zend SafeGuard Suite Introduction

Zend SafeGuard Suite is the first Electronic Licensing solution for the PHP marketplace, combined with the Encoding solution that pioneered PHP intellectual property protection.

Your unprotected intellectual property, in the form of plain-text PHP scripts and software with no license restrictions, can be copied, modified, and maintained by someone else. It is open to your competitor, to hackers, and to well-meaning developers at deployed customer sites.

The Zend SafeGuard Suite provides the tools that eliminate the risks of your intellectual property being viewed or tampered with.

Software piracy losses run in the billions, estimates place it a $59 billion in the last five years.[1] This translates into lost opportunities and diminished sales.

Using the SafeGuard Suite maximizes software profitability by:

- Eliminating unauthorized duplication or use of your applications.
- Ensuring that only licensed customers use your products and that they stay within the terms of your license.
- Offering flexible licensing terms that make your software products more attractive, increase sales and improve customer satisfaction.
- Increasing conversion rate from evaluation to licensed product.

[1] Survey by the Business Software Alliance (BSA) and the Software & Information Industry Association (SIIA).

The Zend SafeGuard Suite is comprised of two components providing a dual layer of protection for PHP applications:

Zend Encoder

The industry standard in PHP intellectual property protection, the Zend Encoder protects exclusive and commercial PHP applications by encoding PHP scripts. Developers enjoy the benefits of a leading open-source language while protecting PHP code when ready to distribute applications. By protecting applications, Independent Software Vendors (ISVs) expand distribution and revenue on maintenance and support.

The Zend Encoder creates encoded binary files that can be distributed instead of the human-readable PHP files, while performing exactly the same. By saving the code in a protected Zend Intermediate Code format, the Zend Encoder allows you to protect source code from copyright infringement. This enables you to create exclusive software solutions and commercial PHP applications while protecting their investment and intellectual property.

Zend License Manager

The Zend License Manager enables ISVs to manage the commercial distribution of their PHP applications by generating license keys and creating files that require a license key to operate. Software vendors can easily specify license models without changing their application's source code. Licensing PHP applications enables software vendors to dramatically penetrate their available market, enhance customer loyalty and account penetration, resulting in increased revenue and profitability.

The Zend License Manager extends to developers and ISVs the option to not only protect that code from copyright infringement, but also impose license restrictions on those files by marking such

encoded files as ones that require a license to operate. The Zend License Manager generates unique licenses based on defined criteria, manages those licenses and then ensures those files require the licenses to operate. Applications will not run unless the proper software license is found, allowing vendors to invoke license policies that will result in maximized sales.

System Requirements

- Supported PHP versions: 4.0.4 and above.
- Supported operating systems:

  -Windows NT, Windows 98, Windows 2000 or Windows XP

  -Linux (Intel) – glibc 2.1 or later

  -FreeBSD 3.4 or later (Command Line Encoder Only)

  -Solaris (Sparc) 2.6 or later (Command Line Encoder Only)
- Supported Web Servers for deployment: Apache 1.3.x, IIS 4 / 5, Zeus (using FastCGI) or any Web server that supports CGI
- The Graphical User Interface is supported under Windows and Linux only.

Version compatibility

PHP files encoded with the Zend Encoder 3.1 require Zend Optimizer 2.0.0 or greater in order to be run. Encoded files can be used with PHP 4.0.5 or greater.

PHP files encoded with the Zend Encoder 2.0 or Zend Encoder 1.2.0 require Zend Optimizer 1.2.0 or greater in order to be run. Encoded files can be used with PHP 4.0.4 or greater.

PHP files encoded using the Zend Encoder 1.1.0 require Zend Optimizer 1.0.0 or greater in order to run correctly and can be used on PHP 4.0.3 or greater.

Note:	Newer PHP language features and constructs should be run on PHP versions that support them. This is true for both encoded and non-encoded php files.

Quick Start

The following Quick Start procedures take you through the steps needed for encoding and generating licenses for files.

There are two procedures

- Quick Start A: Encode
- Quick Start B: License

Quick Start A: Encode

1. If the Zend Encoder tab is not the active tab, select the **Zend Encoder** tab, then click  , enter a valid file name in the dialog box, and click **OK**.
2. **To add directories:**
   In the Project Browser, right mouse click and select **Add Directory**, select a directory and click **OK.**

   To add files:
   In the Project Browser, right mouse click and select **Add Files**, select a directory and click **OK.**
3. Select an optimization level in the Optimizations area.
4. In the Target Directory area, click the browsing button, select the directory for output and click **OK.**

Caution: Files in the target directory will be overwritten.

5. If you want the encoded files to expire on a set date, in the **Expiration Options** area, select **Expire On**, then enter the day, month and year in their respective boxes.

6. If you intend to license the encoded files, in the Licensing Support area select the level of licensing support and enter the name of the product in the Product Name box.

The product name must exactly match the product name used to generate the license (case sensitive).

7. In the **Options** area, select the options you want:
   - If your scripts use PHP short tags, select **Short Tags**.
   - If your scripts use ASP tags, select **ASP Tags**.
   - If you would those file types not specified in the general settings copied into the output, select **Copy Non-PHP Files**
   - If you would like the encoded files to only work with files encoded and sharing the same signature, select **Work Only with Other Encoded Files**.
8. Click **Encode.**

Quick Start B: License

1. If the Zend License Generator tab is not the active tab, select the Zend License Generator and enter the product name in the Product Name box.

> The product name must exactly match the product name used for encoding the files (case sensitive).

2. Enter the name of the registered owner in the **Registered to** box.

3. If you want to lock the license to a designated computer,

- Run the zendid.exe program on the command line of the machine where the license will be used and get the Zend Host ID number. For multiple machines, you will have to repeat this step on each machine. For more details, see

- Getting the Zend Host ID on page 37
- In the Zend SafeGuard Suite, select Lock to Zend Host ID(s) and enter the Zend Host ID number. Multiple Zend Host ID numbers should be separated by spaces.

4. If you want the license to expire on a set date, select **Expire On** and then enter day, month and year in their respective boxes.

5. If you want the license to be limited to use by an IP or an IP series, select **Limit to the following IP(s)** and enter one of multiple IP address numbers separated by spaces.

Limit to the following IP(s) supports the use the asterisk for a wildcard placeholder in order to specify an IP series.

6. If you want to specify additional information within the license, select **Additional License Information** and type the information in the following format:

   FieldName **=** *FieldValue*

   where:

   FieldName is a name assigned to the item of information

   FieldValue is the string text (information) for the item

   Field names, which begin with the prefix **X-** will not be coded into the license signature and can be changed within the license without affecting the licenses validity.

7. Click **Generate License**.

8. Enter the license name and extension and click **OK**.

9. Install the license as described in License Installation section on page 31.

Workplace Overview

The SafeGuard Suite user interface has two tabs:

- Zend Encoder
- Zend License Generator

Zend Encoder Toolbar

The following are the Zend Toolbar Icons and brief description for each.

Toolbar Icon	Description
	**Save Project Button:** Saves the current project.
	**Create New Project:** Opens a blank project in the project browser.
	**Open Existing Project:** Opens a file browser for opening existing project files.
	**Stop Encoding:** Stops the execution of the encoding process.

Zend Encoder Settings

Name	Description
Optimizations	Allows selection of optimization levels to be applied in the encoding of PHP files.
Target Directory	Allows you specify the upper path, which the encoded files and included directory structure will be written. Contains a **Create Directory Button**, which opens an input box to enter and create a path.
Licensing Support	If Zend License Generator is installed, provides several selections for setting the level on license requirement. Adds product name and license support levels to the encoded signature.
Encoder Messages	This area displays encoding messages, such as error messages and encoding results.
Application Tabs	Allow navigation between the Zend Encoder and the Zend License Generator.
Expiration Options	Allows you to set an expiration date for encoded files, causing the files to expire and no longer work on the stated date.
Encoder Options	Allows you to select several settings to control the following: • Encoding of Short Tags • Encoding of ASP Tags • Work Only with Other Encoded Files • Copy Non-PHP Files
Encode Command Button	Performs the encoding as specified by the various Encoding settings.

Zend License Generator Settings

Name	Description
Product Name	The name of the product. Must match the Product Name given to the encoded files which the license will work with.
Registered to	The name of the licensee.
Lock to Zend Host IDs	Lock to a specified Zend Host ID(s). A Zend Host ID is a generated code which identifies a specific machine.
Expire On Date	The expiration date on which the license is no longer valid.
Limit to the following IPs	The IP information, which binds license to a specified IP or IP series.
Additional Information	User information, which can be included in the license file.
Generate License Command Button	Generates a license based on the License information entered.

Overview Zend Encoder

The Zend Encoder allows code developers to encode their PHP script at any time during the development process, but more importantly prior to distributing or publishing it.

The Zend Encoder allows you to encode your PHP scripts in several different ways. The following lists some of the options for encoding files:

- **Expedited at Run-time:**
  Eliminating compilation and optimization at run-time.
- **Unreadable Source Code:**
  Files are encoded and not in a human readable format
- **Require Valid License:**
  Files can be encoded to support or require licensing
- **Time Limited:**
  Files can be encoded to expire at a set date
- **Encoded-Only Mode**
  Files can be set only to cooperate with associated encoded files that bear the same encoded signature.

Expedited at Run-time

The Zend Encoder optimizes PHP code, which means faster execution and a reduction of the CPU load for the server. Because the files are encoded and optimized, there is no optimization or compilation process done at runtime. This reduces the number of processing steps at runtime, which can translate into faster runtime execution.

Secure

By saving the code in a closed Zend Intermediate Code format, the Encoder allows developers to protect source code from

copyright infringement, enabling companies to create exclusive software solutions and commercial PHP applications.

The Zend Encoder converts your plain-text PHP scripts into a platform-independent binary format. This provides protection so that no one can easily access or tamper with your original source file. In addition, this provides extra protection against reverse engineering.

License Requirements

This feature works in conjunction with the Zend License Generator and allows you to specify the level of licensing. This level of licensing is encoded into the file. There are three choices, which you can specify to the encoded files:

- **No Licensing**
  There is no interaction with a license file and no license is required to use the file.
- **Support Licensing**
  The file will not require a valid license unless the script was developed to require a valid license. Developers can use a Zend SafeGuard Suite API function to check for a valid license.
- **Require Licensing**
  The file will not work unless a valid license is available.

Encoding only determines these levels, specific licensing details, such as the scope of a license is determined in the license generation. For more details about license generation, see Overview of the Zend License Generator on page 28.

Time Limited

This feature allows you to specify a set date on which the encoded file will no longer be valid (expires at the arrival of the expiration date.) this means all distributed copies of the file will cease being valid on the expiration date. This option should not be confused with the expiration of a license, which controls an individual user. For more details on license expiration dates, see Setting License Restrictions on page 33.

Encoded-Only Mode

For added protection of your encoded files, you can specify Work Only with Other Encoded Files as an option. These files cannot be called by files that do not share the same encoded association, thus creating barriers, which make exposure of the inner workings much more difficult for hackers and reverse-engineers.

The encoded platform-independent-format can be deployed on all supported platforms which have the Zend Optimizer installed on their server. For details on supported platforms, see System Requirements on page 4.

Zend Encoder - Getting Started

The Zend Encoder has three main steps

- Creating a Project and Inserting Files and Folders
- Selecting the Encoding Settings
- Encoding Execution

The first step before you can begin is to create a project; lets take a look at what projects are and how to create them.

Creating a Project and Inserting Files and Folders

Projects are a collection of source inputs such as files and paths of files to be encoded. Project files have the file extension of **zep**.

When the Zend Development Studio is installed, files contained in a project offer quick-open functionality. By double clicking on a project file in the Zend SafeGuard Suite Project Browser, the file is automatically opened in the Zend Development Environment.

This functionality is handy if need for scripts changes arises when you are encoding.

To Create a Project

1. If the Zend Encoder tab is not the active tab, select the Zend Encoder tab, then click 
   The New Project Dialog opens.
2. Enter a valid file name and click OK.
   The new blank project opens in the Project Browser with the project name at the top of the browser frame.

To Open an Existing Project

1. If the Zend Encoder tab is not the active tab, select the **Zend Encoder** tab, then click 
   The Project Selector opens.
2. Choose a project file and click **OK.**
   The new blank project opens in the Project Browser.

To Save a Project

- If the Zend Encoder tab is not the active tab, select the Zend Encoder tab, then click 
  The active project is saved

Once you have the project defined, you can begin adding the source file and folders, which you want encoded.

Add Files and Folders to a Project

Adding files and folders to the current project, determines what files will be encoded. By default, you can only the following:

- Folder Structures
- Defined File Types

Folder Structures are treated as recursive. This means any file found in the immediate folder or below, which match the **Extension Definitions** will be processed for encoding.

Additionally the contents of these structures should be considered if you plan to select **Copy Non-PHP Files** as an encoding setting. When **Copy Non-PHP Files** is selected, the folder remaining contents (those extensions not specified in the General Settings) are be copied to your output directory.

Defined File Types are files that match the extensions list as set in the Zend General Settings area of the Zend SafeGuard Settings. By default, these are limited to the following: inc, php, ihtml, php3, and php4. For more information on changing file extensions, see Changing the Extension to Encode on page 46.

To Add a Folder to a Project

1. In the Project Browser, right mouse click and select **Add Directory**.
   The folder browser opens.
2. Browser and select the directory to add, and click **OK**.

To Add Selected Files to a Project

1. In the Project Browser, right mouse click and select **Add Files**.
   The file browser opens.
2. Browse and select the directory to add then click **OK**. For multiple files hold the CTRL down and select the files.

Once the files and folders are added to a project, you can begin to select the encoding settings.

Selecting the Encoding Settings

When you set up your project you Encoding-setting need to be set according to the several factors that deal not only with the source files content, but also what the output and encoded results should be. The following are the available settings:

- Optimizations
- Expiration Date
- Encoding of Short Tags
- Encoding of ASP Tags
- Copy Non-PHP Files
- Work Only with Other Encoded Files

- License Support
- Target Directory

Optimizations

Optimization provides additional efficiency in the execution of scripts.

Optimizations are done to provide faster execution and a reduction of the CPU load for the server.

The following settings are supported:

- Full
- Minimal
- None

Some scripts may not support optimization (most notably, scripts that interface with COM or Java objects). If you experience problems with the encoded scripts, try reducing the optimization level.

Expiration Date

Encoded files can be set to expire on a set date. This allows you to put time restraints on encoded files.

When the actual date is greater than or equal to the expiration date encoded into file, then the file will no longer work.

Use this functionality for limiting a file's life span, such as a sample version that will expire on the date when the final product is released.

This feature is not related to the License Manager and should not be confused with the expiration date of licenses.

Enabling Short Tags Encoding

This setting allows you to encode scripts written using short PHP tags. Short tags are <?.

Unless this option is enabled, the Zend Encoder will not recognize and encode short tags and PHP marked by short tags will be treated as regular HTML.

Enable ASP Tags Encoding

This setting allows you to encode scripts containing ASP tags. The Zend Encoder will not encode ASP tags unless this option is enabled. If not selected, ASP code within ASP tags, will be treated as regular HTML.

Copy Non-PHP Files

During encoding, the output of the PHP encoded files is sent to the target directory by default. However, there are often files, which were not encoded, such as images that need to be copied to the output directories.

This option allows you to copy those project files, which do not fall under the General Settings file types and are resident in the project folder. These files will be copied into their respective output directories saving the effort of manual copying.

Work Only with Other Encoded Files

This option generates files that work exclusively with associated encoded files that bear the same signature of origin. In other words, encoded files generated by the same encoder.

You can encode your files to only cooperate with encoded files bearing your signature This prevents users from including the encoded file from their own code, and possibly using PHP's introspection functions to reveal information about the structure of the file.

This creates an added barrier, which make exposure of the inner workings much more difficult potential hackers and reverse-engineers.

License Support

This option allows you to encode files to work with the License files. There are three setting to choose from:

- **No Licensing Support**
- **Require Valid License**
- **Support Licensing**

```
Licensing Options are only available if the Zend
License Generator is installed.
```

Require Valid License

This allows you to generate encoded files that cannot be run unless a valid license exists. The license requirement is automatically enforced at all times. Supports use of Zend Safeguard Suite API function only when a valid license exists.

Either there is a valid license found and the file can be used or no valid license is found and the file will not work.

Note that license placement is vital for validation to be performed. If a license is not in the correct location, a not-licensed status will be assumed. License validity and locations are checked and loaded at the time PHP starts up.

No Licensing Support

Files are encoded without any form of licensing support. Additionally, these files do not support the use of the license API.

Support Licensing allows you to set up license functionality that is only enforced using your scripts and a Zend Safeguard Suite API function check for a valid license.

This allows you to generate encoded files that can operate within a limited scope, such as a limited trial version that requires a valid license in order to be fully functional.

Disable PHP-compatible header

This option allows you to create old-style encoded files, that don't have begin with a PHP-compatible header. Files that don't have a PHP-compatible header, will not be recognized by PHP, unless the Zend Optimizer is properly installed. This will save approximately 1.5KB for every encoded file, but does not affect performance in any way. It's recommended that you don't disable PHP-compatible headers, unless you need to save every last bit of disk space.

Append header information

This option allows you to add information and/or code to the header of every encoded file. The information, embedded in the form of PHP comments, will be viewable by your end-users, and can be used to display copyright and version information. If you embed any PHP code in the header as well, it will be executed only in the event that the Zend Optimizer is not present. Adding PHP code can be useful in case you would like to display custom

error messages when the Zend Optimizer is not properly installed. For instance, if you want to tell people to reinstall the product in case the Zend Optimizer is not properly installed, you can use code similar to this:

```
print “ExampleApp is not properly
      installed.  Please consult the
      <a
      href=\http://www.exampleapp.com/
      \”> User Manual</a>, and
      reinstall it.\n”;
exit(1);
```

The combined size of the comments and PHP code blocks is limited to approximately 62KB.

Ignore Patterns

This space-delimited list contains patterns that will be completely ignored when encoding a directory. Entries that match these patterns will not be encoded, nor will they be copied as-is to the target directory. By default, the list contains the CVS directory and .cvsignore files.

Output

Output includes the encoded files, any sub folders containing files, and if **Copy Non-PHP Files** is selected, copies the rest of the files located in the project folders.

With the recursive functionality of project folders and sub folders, it is important to consider the output of the path structures, such as sub folders, when defining a Project.

Individual files added to the project will be output with no path information.

For Directories defined in a project, output will only contain unique directory structures. Directory structures common to all project directories will be not be output.

Example 1

The directory FinalRelease was the only directory added to a project.

The following are the full path of the files below the FinalRelease directory:

- C:\FinalRelease**Module1\Dialogs**NameSelector.php
- C:\FinalRelease**Module1\Screens**Welcome.inc

If the target path were C:\Products\ABCSoftware\FinalBeta the resulting output would be:

- C:\Products\ABCSoftware**Dialogs**NameSelector.php
- C:\Products\ABCSoftware**Screens**Welcome.inc

Notice that the Module1 directory was not included even though it was a subfolder of the FinalRelease directory.

Example 2

The scenario is the same as example 1 except FinalRelease now contains a new directory Module2. The following are the full path of the files below the FinalRelease directory.

- C:\FinalRelease**Module2\Dialogs**MyForm.php
- C:\FinalRelease**Module1\Dialogs**NameSelector.php
- C:\FinalRelease**Module1\Screens**Welcome.inc

If the target path were C:\Products\ABCSoftware the resulting output would be:

- C:\Products\ABCSoftware**Module2\Dialogs** MyForm.php
- C:\Products\ABCSoftware**Module1\Dialogs** NameSelector.php
- C:\Products\ABCSoftware**\Module1\Screens** Welcome.inc

Notice that Module1 folder is included.

If the project had an additional directory added to it, all the structures below the added directories are output.

You can either select a file structure or use the Create New feature to enter your desired folder structure.

Select a Target Path

1. On the Zend Encoder tab in the Target Directory area, click the browsing button
   The Directory browser opens.
2. Select the directory for output and click **OK.**

Caution: Files in the target path will be overwritten.

Create a New Target Path

1. On the Zend Encoder tab in the Target Directory area, click the **Create New** button
   The Create New dialog opens.
2. Enter the full path to create and click **OK**.
   Target Directory is set to the created path.

Encoding Execution

Once you have selected all the Encoding Settings, you can execute the Encoding of the Project files.

Results, errors and messages are reported to the Encoder Messages area during encoding. When encoding errors occur a brief explanation will appear and a prompt will display.

If a php generates an error during encoding, the encoding-execution of project files stops. You must either correct the error or remove the file containing the error from the project. In order to encode all the files or a project they must be free of errors.

If you have Zend Studio installed, you can double click on the error message and the selected file will open in the Zend Development Environment to the line of the script that caused the error. For information on owning the Zend Studio refer to the Zend Website: http://www.zend.com/products.php

You can stop encoding execution by clicking the  button on the Zend Encoder Toolbar.

Overview of the Zend License Generator

The Zend License Generator allows you to issue permission in the form of a license file. This license allows you to control the use of your encoded files as well as preventing piracy, unauthorized or invalid use.

You can define several types of restriction upon of a license. The basic result is that the use of license is either valid or it is not. In addition, the requirement of a license can be enforce at all times (**Require Valid License**) or only when you code checks for a license (**Support Licensing.**)

The Zend License Generator allows you to control the following:

- **Lock to Zend Host ID**
  The Zend Host ID must match the Zend Host ID for which the license is issued.
- **License Expiration Date**
  License is only valid for a period of time prior to the expiration date.
- **Limit IP Access**
  License is only valid if the IP address numbers match the IP or IP series specified for the license.

In addition to these, a license file can contain lines of user-defined information, which if changed, invalidate the license.

- **Additional Information**
  Any additional information in the file is protected by the digital signature. Tampering with these extra lines will invalidate the license. This can be used to store additional information in the file, which can later be used by the application (e.g., Shareware=yes).

These lines appear as text in the license file and can be accessed by the same means as accessing a text file. In addition, if a license is valid, the Zend Safeguard Suite API function `zend_loader_file_licensed` returns an array that contains these values.

You can also include Additional Information lines, which will not affect the license. In order not to affect, the license for Additional-information-lines must begin with **X-** prefix (e.g., X-Comment = Install in /etc).

All lines that begin with X- will be ignored when the license signature is generated, however these lines will be returned by the `zend_loader_file_licensed` when the license file is determined valid.

Caution: Any information changed within the license file that does not begin with X- will invalidate the license. This prevents unauthorized modifications to the license file.

License Support Options

When you encode your php files to work in conjunction with a license, you can choose Require Valid License or Support Licensing as settings to control the level of licensing support.

Require Valid License

This option encodes you php files to only work if a valid license file exists. Because licenses can only have two states valid or invalid, the file will either work or not work. This means that if a valid license is not found, PHP will refuse to load the file. A license file will only work when a valid license is installed and the

php server is restarted. For information on license installation, see License Installation on page 31.

Example: A Require Valid License application file is copied to an IP other than the IP address specified in the license. The result is that the application would not open.

Support Licensing

This option works together with the License API. Files encoded with this option can be opened and run the same as other encoded files, however with the use of the License API you can program at what point you will check for a valid license.

Example: A Support Licensing application file is copied to an IP other than the IP address specified in the license. The encoded files do not automatically check for a license until the script calls for a license check. The application loads and all the menus appear because the programmer has not yet called for a license check.

The user does some work and sees the great functionality of this application and then he tries to save his work. At this point, the programmer uses the Zend SafeGuard Suite API function to check for a valid license.

The function returns a FALSE value. The programmer implemented a script to handle unlicensed users (FALSE value). The script prompts the user to get a license or quit. The user chooses to buy a license through a web link, and everybody is happy.

License Installation

In order for licenses to be validated and loaded, the location of the license must be entered in the php.ini file prior to the startup of the php server.

Licenses are determined valid or not valid. Situations where no valid license is found are treated the same as an invalid license. Therefore, proper installation of a license is necessary.

Information regarding the location of license file should be written in the php.ini file. In this way the license-file can verified and if valid, loaded into the license registry.

Note: The php web-server must be restarted for changes in the php.ini and new valid licenses to be loaded into the license registry.

You can define specific files or license file directories. If you specify a license directory, the result is that all files with the file extension **zl**, are checked for validity and if valid loaded into the license registry. This is done once per server session at the startup of the php server.

Files encoded to check for valid licenses will check the license registry for a license matching its product specification.

The directive in the php.ini file for license paths is zend_optimizer.license_path. The syntax is as follows:

zend_optimizer.license_path=*"LicensePath*[1]*:LicensePath*[2]*: .... Path*[n]

Where:

LicensePath is the path to the file or directory where the correct license file can be found. For UNIX, multiple paths are entered

separated by colons (colon delimited.) For Windows, multiple paths are entered separated by semicolons (semicolon delimited.)

Example 1

The following line shows two license files specified (UNIX).

```
zend_optimizer.license_path=/usr ocal/Zend/licenses/Lic1.zl:
/usr/local/Zend/licenses/Lic2.zl
```

Example 2

The following line shows a license file and a license folder specified (Windows).

```
zend_optimizer.license_path=/usr ocal/Zend/licenses/Lic1.zl;
/usr/local/Zend/user_licenses
```

To Install a License File

1. Open the php.ini file in a text editor.
2. Find the line with the directive **zend_optimizer.license_path**, if a line for this directive does not exist add a line to the php.ini and type **zend_optimizer.license_path=**
3. If the path where the license file resides is not found on the **zend_optimizer.license_path** directive-line (after the = sign), then add it to the end of the line. Remember to separate the paths with a colon (for UNIX) or a semicolon (for Windows).
4. Restart the Web-server for the changes take effect.

The License File

The license file contains digitally signed data, Zend License Generator settings and any user defined information entered at the time of generation. The following is an example of the content of a license file:

```
Product-Name = My Product
Registered-To = ABC Compa y
Hardware-Locked = No
Expires = 02-Jun-2002
X-Home = Region 7
Shareware = True
Verification-Code = wCQbt7C DnhX/PHimrNez4IYOs+LA
```

Setting License Restrictions

License restrictions control the status of a license file. License files are check for validity at the start of the PHP server. All valid license information is then stored in the memory of the License Registry. Invalid licenses are also registered, to allow the application to check what kind of error occurred.

The license enforcement is handled as follows:

- For **Require Valid License** files, the Zend Optimizer checks and then allows or prohibits its running of the encoded file.
- For **Support Licensing** files, you must program the call for a license lookup using the Zend SafeGuard Suite API, and implement your policy based on whether a valid license is available or not.

- The following is a description of the license fields:

Name	Type of information	Description
Product-Name	This is the product name, which must match (case sensitive) to the product name assigned to the encoded files.	Used to match licenses and encoded files. When a license-check is performed, all licenses found in the license directory are reviewed by license signature that includes the product name. Only a license that matches by product can determine license rights and information.
Registered-To	String that contains the Registered Owners name.	Coded into the license signature. It is included in the array returned by the Zend SafeGuard Suite API on valid licenses.
Expires	A proper date string of the format: DD-MMM-YYYY	Sets a date limit on the validity of a license. License is only valid on dates that are less than the stated expiration date. If the date is equal or greater than the stated expiration date the license file is invalid.

Name	Type of information	Description
IP-Range	Standard format of IP addresses that includes four numbers separated by periods. Wildcard asterisk characters can be used in place of numbers (placeholder) to allow a range of IP addresses. Example 10.172.132.*	Ties the license to the stated IP number or range or IP numbers. License becomes invalid if located on any other IP other than one that matches the specification.

Name	Type of information	Description
Additional Information	User-defined information is entered in an initializing format. *FieldName* = *FieldValue* Field names entered with a prefixed of X- are not coded into the license signature. All field values are treated as text. Fields not beginning with X- are coded into the license signature. Example: City = New Brunswick X-Navigation = 4	Provides licensing and information capabilities for customized licensing requirements. Information contained on lines not prefixed with X- are coded into the license signature. If these lines do not match the license signature, the license is not valid. Comparison is case sensitive. Field names prefixed with X- can be used to hold data, which can be updated and changed without affecting the validity of the license. All the information in a valid the license file can be retrieved using the licensing API.

Getting the Zend Host ID

The Zend Host ID is used for generating licenses locked to a specific machine (hardware).

In order to obtain the Zend Host ID you must copy and run the zendid.exe program on the machine, which the license is to be issued.

The zendid.exe program can be found with the files of the Zend License Generator.

To Get the Zend Host ID (Windows)

1. Copy the zendid.exe application to the users computer.
2. From the Windows Start menu, select Run.
   In the Run dialog opens.
3. In the Run dialog type:
   - For Windows NT, 2000, or XP operating systems, type **cmd** and click **OK**
   - For Windows 95, 98, or ME operating systems, type **command** and click **OK**

   The DOS Command Shell opens.
4. In the shell application type **zendid.exe** and press **ENTER**.
   The Zend Host ID prints to the screen as shown in the example below.

```
C:\WINDOWS\Desktop>zend d.exe
H:MFM43-Q9CXC-B9EDX-GW  'SU
C:\WINDOWS\Desktop>
```

5. Record the Zend Host ID code.
   In the example H:MFM43-Q9CXC-B9EDX-GWYSU

6. In the shell application type **Exit** and press **ENTER**.
   The shell application closes.

▸ To Get the Zend Host ID (UNIX)

1. Copy the zendid.exe application to the users computer.

2. Open a terminal and run the command **zendid.exe**
   The Zend Host ID prints to the screen as shown in the example below.

3. Record the Zend Host ID code and close the terminal session.

Zend SafeGuard Suite API

With the Zend SafeGuard Suite API, you can complete the following tasks.

- Check if the Zend Optimizer is enabled to handle encoded files.
- Check if a valid license exists and gather information from a valid license.
- Get the running files path at runtime

Name	**`zend_loader_enabled`**
Synopsis	Checks the Zend Optimizer's configuration to verify that it is configured to load encoded files.

Note: The Zend Optimizer setting can be configured in the php.ini file. Enable Optimizer line syntax: **zend_optimizer.enable_loader = on|off**

Syntax	**`zend_loader_enabled()`**
Results	Returns Boolean TRUE The optimizer is configured to load encoded files. FALSE The optimizer is not configured to load encoded files.

Name	**zend_loader_file_licensed**
Synopsis	Compares the signature of the running file against the signatures of the License files loaded by the php.ini file into the License Registry. If a valid license file exists, the values of the license file are read into an array. If a valid license does not exist or is not indicated in the php.ini, it will not be entered in the license registry at the startup of the php server. If a valid license, matching in product and signature cannot be found in the license directory, an array is not created. For information on the proper installation of a license file as well as the php.ini directive, see License Installation on page 31
Syntax	**$lic_info = zend_loader_file_licensed()**

Results

Returns an array or FALSE

array

If an array is returned, a valid license exists in the location indicated in the php.ini file and for the product. The array has an element for each line of the license file. This includes the license generation settings and any additional user information added to the license.

Example of a line in the license file:

Product-Name = My Product

The array index names are the line content from the left side of the equations (***Product-Name***) within the license file text.

The array values are the text content to the right or the equation (***My Product***).

FALSE

Indicates no valid license found. This can mean that no license exists in the license directory or the license file exists and has become invalid or corrupted.

Name	**zend_loader_current_file**
Synopsis	Obtains the full path of the currently running file at runtime. In other words the path of the file calling this API function. Does not to evaluate the running files path during encoding, but evaluates only at runtime.
Syntax	**zend_loader_current_file()**
Results	Returns a string containing the full path of the currently running file.

Changing Zend SafeGuard Settings

The Zend SafeGuard Suite allows you to change the following

- Change User-Interface Language
- Zend Encoder Path
- Zend Signer Path
- Zend Development Environment Path
- Extension to Encode
- Default Encoding Settings

Change User-Interface Language

You can change the language of the Zend SafeGuard Suite user interface to the following language settings:

- English
- German
- Japanese (requires special version)

▸ To Change the Language of the User Interface

1. From the main menu, select **Tools ▸ Settings**.
   The SafeGuard Suite Settings Dialog opens.
2. From the **User Interface Language** menu, select the language to apply and click **OK**.
   The user interface updates to the current language selection.

Zend Encoder Path

This is the location path to the Zend Encoder executable file (zendenc.exe). This setting is originally set in the installation process.

Change this if the Zend Encoder executable file has moved or its path changed. If this path is incorrect, the Zend encoder will not work.

▸ To Change the Path of the Zend Encoder

1. From the main menu, select **Tools ▶ Settings**.
   The SafeGuard Suite Settings Dialog opens.
2. Click the **Zend Encoder** browse button and select the path for **Zend Encoder** executable (zendenc.exe).

Zend Signer Path

This is the location path to the Zend Signer executable file (zendenc_sign.exe). This setting is originally set in the installation process.

Change this if the Zend Signer executable file has moved or its path changed. If this path is incorrect, the Zend encoder will not work.

▸ To Change the Path of the Zend Signer

1. From the main menu, select **Tools ▶ Settings**.
   The SafeGuard Suite Settings Dialog opens.
2. Click the **Zend Signer** browse button and select the path for the **Zend Signer** executable (zendenc_sign.exe).

Zend Development Environment Path

If you have the Zend Development Environment (Previously the Zend IDE) installed entering this path will allow you to use the go-to functionality of the Project browser and Encoding Messages, to open a script within the Zend Development Environment.

This is the location path to the Zend Development Environment executable file (ZDE.exe).

The Zend Development Environment is part of the Zend Studio, for more information about obtaining a copy of the Zend Studio visit the Products page of Zend.com.

To Add or Change the Path of the Zend Development Environment

1. From the main menu, select **Tools** ▶ **Settings**.
   The SafeGuard Suite Settings Dialog opens.
2. Click the **Zend Development Environment Location** browse button and select the path for the Zend Development Environment executable (ZDE.exe).

Changing the Extension to Encode

These file extension determine which files from project folders will be encoded as well as effecting which file types can be included in the Project Definitions.

This option allows you to encode PHP files with the file extensions you specify.

▸ To Change the Defined File Types

1. From the main menu, select **Tools ▸ Settings**.
   The SafeGuard Suite Settings Dialog opens.
2. In **Extensions to Encode** box of the Zend SafeGuard Settings dialog box enter the file extension to add. Each extension should be separated by a SPACE.
3. Click **OK**.
4. If you have directories previously added to your project, right mouse click in the project browser and select **Rescan Directories**.
   Directories are updated to reflect the defined file types.

Default Encoding Settings

You can set the defaults to suite your encoding preferences. This is especially handy if you encode in a somewhat consistent manner. These default settings appear when opening a new project.

These are only the defaults; the actual values of encoding settings at the time of encoding determine how the files are encoded.

The customizable default settings include:

- **Optimizations**
  Full, Minimum, and None
- **Short Tags**
- **ASP Tags**
- **Copy Non-PHP Files**
- **Work Only with Other Encoded Files**

▸ To Change the Default Encoding Settings

1. From the main menu, select **Tools ▶ Settings**.
   The SafeGuard Suite Settings Dialog opens.
2. In the Default Projects Settings area, select the new settings defaults and click **OK**.
3. If you have directories previously added to your project, right mouse click in the project browser and select **Rescan Directories**.
   Directories are updated to reflect the defined file types.

Command Line

Encoding and License Generation can be performed on the command-line using the following commands inside a command shell.

- **zendenc**
  Command-line version of the Zend Encoder for encoding files.
- **zendenc_sign**
  Command-line version of the License Generator for creating a signature license file from a license definition file.

zendenc - Command

Command	**zendenc**
Synopsis	**Command-line function for encoding files.**
Syntax 1	**zendenc** *[--option1 option_parameter1 --option2 option_parameter2 --optionn option_parameter^n]* **SourceInputPath** [*EncodedOutputPath*]

Arguments	SourceInputPath	The path and/or file name of the file source directory or file. This must be a valid full or relative path. This is a mandatory argument for all encoding operations. The command line syntax requires the **SourceInputPath** parameter precede the **EncodedOutputPath** parameter.
	EncodedOutputPath	The path and/or file name of the target file name where the encoded file is written. This must be a valid full or relative path. This argument is not required for encoding operations using the **--delete-source** and **--rename-source** options. The command line syntax requires the **EncodedOutputPath** parameter (when used) is entered following the **SourceInputPath** parameter.

--option[x] [option_parameter[x]]	Various options can be entered to control the functionality of **zendenc**. Options may have parameters that immediately follow the option. Every option must be preceded by a double dash **--** prefix.

Option Syntax	Option Description
--asp-tags {**on**|**off**}	Turns ASP tag (“<%”) recognition either on or off. **on** or **off** and must be specified as an argument when using this option. The default, when option is not used in the command-line, is **off**.
--optimizations *opt_mask*	Optimization mask. opt_mask is an integer representing a bit-mask. The default value enables all of the optimization passes. Each optimization pass of the Zend Optimizer can be turned on or off based on the mask entered.

Option Syntax	Option Description
--encoded-only	**Generate for Encoded Cooperation Only**. This option generates files that work exclusively with associated encoded files. Associated encoded files are those generated by the same company. These files cannot be called by files that do not share the same encoded company association
--recursive	Source files are selected recursively in the ***SourceInputPath*** and subdirectories. By default, the following extensions are encoded when this option is used: **inc php ihtml php3 php4**
--include-ext *ext*	Used in conjunction with the **--recursive** option to add additional file extensions to the default extensions (inc php ihtml php3 php4.) This option can be entered more than once to include multiple files. ***ext*** is the file extension of the files, which will be included in the encoding source files.

Option Syntax	Option Description
--exclude-ext *ext*	Used in conjunction with the **--recursive** option to remove file extensions from the default extensions (inc php ihtml php3 php4.) This option can be entered more than once to exclude multiple files. ***ext*** is the file extension of the files, which will be excluded from the encoding source files.
--ignore *FileSpecification*	Used in conjunction with recursive mode to filter out files matching these specifications. This option can be entered more than once for ignoring multiple files. ***FileSpecification*** is the string to compare filter out matching files. **Example:** Data*.php would exclude all php files beginning with “Data”.
--ignore-errors	Used when encoding multiple files. If an encoding error occurs while encoding a file, the Zend Encoder continues processing the other files.
--force-encode	Allow encoding of previously encoded files (NOT recommended!)

Option Syntax	Option Description
--expires *yyyy-mm-dd*	Encode the designated files with an expiration date. Files expire upon the arrival of the expiration date. Date is in **yyyy-mm-dd** format.
--short-tags {**on**|**off**}	Turns short PHP tag (“<?”) recognition either on or off. **on** or **off** must be specified as an argument when using this option. The default, when option is not used in the command-line, is **on**.
--no-header	Disables the PHP-compatible header that is added to the top of every encoded file by default. Encoded files generated with this option will *not* display a meaningful error, when loaded by PHP that doesn’t have the Zend Optimizer properly installed. Using this option saves approximately 1.5KB for every encoded file; Using it is only recommended if disk space constraints are important.
--prolog-filename *filename*	The contents of the specified file will be added to the top of every encoded file. Overrides --no-header.

Option Syntax	Option Description
--license-product *ProductName*	Encodes files to only work with a valid license for the ***ProductName*** specified (encoded into a signature). ***ProductName*** must exactly match the Product Name entered when generating a license. This is the same as the **Require Licensing** setting.
--sign-product *ProductName*	Encode files with the product name signature. Same as the **Support Licensing** feature, which works with the Zend SafeGuard Suite API function to identify if a valid license exists. (Scripts check for this signature.) ***ProductName*** must exactly match the Product Name entered when generating a license
--delete-source	Permanently deletes (see warning below) the original source files specified in the ***SourceInputPath***, and saves the encoded files in its place. This option has no option parameter. When this option is use, do not use the ***EncodedOutputPath*** parameter.

Option Syntax	Option Description

Warning: To avoid permanent loss of non-encoded scripts, make a backup. Deleted files cannot be restored or recovered and will be permanently deleted with this option.

If you are unsure about deleting the source files, use the `--rename-source` option instead.

`--rename-source` *`ext`*	Renames the original source file by adding the parameter ***`ext`*** as a file extension. Creates an encoded file of the same name as the original ***`SourceInputPath`*** file. When this option is use, do not use the ***`EncodedOutputPath`*** parameter.
`--quiet`	Reports names and errors only. Does not report the progress of the encoding or messages other than errors. This option does not have an option parameter.
`--silence`	Reports errors only. Does not report names, progress, or messages other than errors. This option does not have an option parameter.
`--help`	Displays help about encoder options. This option is entered as follows: **`zendenc --help`**

zendenc - Command-Line Examples

The following are examples, which use the command-line zendenc and its options. Each begins with the command line and is followed by and explanation of its results.

Examples	Command Line and Results
Example 1	/usr/local/Zend/zendenc ––quiet /WorkFolder/mysource.php /EncodedFolder/myencoded.php
	Result: The file /WorkFolder/mysource.php is used as the source of the PHP to encoded. During encoding only errors are reported. The output is an encoded file written to /EncodedFolder/myencoded.php The source file remains unchanged.
Example 2	/usr/local/Zend/zendenc --rename-source old /WorkFolder/ReplaceMe.php
	Result: The /WorkFolder/ReplaceMe.php file is renamed by adding ".old" as a file extension. The new name of the source file is ReplaceMe.php.old. Output is an encoded file written to the same name as the original source file. The result is file /WorkFolder/ReplaceMe.php is an encoded file and the file /WorkFolder/ReplaceMe.php.old is the un-encoded source.
Example 3	/usr/local/Zend/zendenc --delete-source /WorkFolder/mysource.php
	Result: The file /WorkFolder/mysource.php is used as the source then deleted and is replaced by an encoded file of the same name. The result is that the file

	/WorkFolder/mysource.php is an encoded file and the original un-encoded content has been deleted.
Example 4	/usr/local/Zend/zendenc --asp-tags on mysource.php myencoded.php
	Result: The file /WorkFolder/mysource.php is used as the source of the PHP and ASP to encode. ASP tags are encoded along with the php. The resulting output is an encoded file written to /EncodedFolder/myencoded.php The source file remains unchanged.
Example 5	/usr/local/Zend/zendenc --short-tags on --expires 2005-01-01 --recursive --ignore-errors --license-product SuperWebGame --include-ext htm --exclude-ext ihtml –-ignore ”a*” /WorkFolder1/ /WorkFolder2/
	Result: Encodes files in /WorkFolder1/ and below with the file extensions: inc php htm php3 php4 but not ihtml or files beginning with the letter “a”. Encoding will be processed for short tags and each encoded file will both expire on Jan. 01 2005 and will require a valid license for the product name SuperWebGame to work. The output files and any sub structures will be written to /WorkFolder2/. Lastly, the encoding will continue to process all files and will not stop the encoding process because of encoding errors.

Creating a Signature License (command-line)

Signature license files work in conjunction with encoded files that have been encoded with **Require Valid License** or **Licensing Support**. In addition, they must be encoded from the same encoder and license generator.

This can be an encoded file created by the License Generator in the SafeGuard user interface, or an encoded file generated by **zendenc** with the **--license-product** or **--sign-product** options.

There are two ways to create a signed license file:

1. Create a License Definition File, and run **zendenc_sign <lic_def_file> <signed_file>**

or

2. Run zendenc_sign <signed_file>. The signing utility will then prompt you to type in the license information.

License Definition File

The following table lists the license definition parameters, a short description, rank, and required status, as well as examples. All of the following are case sensitive. Every line has to be written in the format:

Entry = Value

(with a single space before the equal sign and another one after the equal sign). The order of definitions will be a permanent part of the signature. This means the lines of the license file must maintain the same sequence. If the order of the definitions in the license file changes from what the signature contains, the license becomes invalid. Definition Files can be created in text editors, such as *Microsoft Notepad* or *vi*.

Field Name	Description
Product-Name	The name assigned to Product. This must be the same name used when encoding the PHP files. REQUIRED **Example:** Product-Name = Drink
Registered-To	The Name of the Registered owner of the license. REQUIRED **Example:** Registered-To = Bob

Field Name	Description
Expires	Expiration date of the license. Used if the license is issued with a date restriction. Format: DD-MMM-YYYY OPTIONAL **Example:** Expires = 09-Oct-2005

Field Name	Description
IP-Range	Limits the use of the license to IP addresses that fall within specification. Supports wildcards for any of the IP place holders as well as the two types of net masks (filters). netmask pair An IP a.b.c.d, and a netmask w.x.y.z. (i.e., 10.1.0.0/255.255.0.0), where the binary of mask is applied to filter IP addresses. ip/nnn (similar to a CIDR specification) This mask consists of nnn high-order 1 bits. (i.e., 10.1.0.0/16 is the same as 10.1.0.0/255.255.0.0). Instead of spelling out the bits of the subnet mask, this mask notation is simply listed as the number of 1s bits that start the mask. Rather than writing the address and subnet mask as 192.60.128.0/255.255.252.0 the network address would be written simply as: 192.60.128.0/22 which indicates starting address of the network, and number of 1s bits (22) in the network portion of the address. The mask in binary is (11111111.11111111.11111100.00000000). OPTIONAL **Example (Wildcard):** IP-Range = 10.1.*.* **Example (Net Mask):** IP-Range = 10.1.0.0/255.255.0.0 **Example (Net Mask):** IP-Range = 10.1.0.0/16

Field Name	Description
Host-ID	Coded string (Zend Host ID) used to lock the license to a specific hardware. The Zend Host ID obtained from the machine where the encoded files and license are to be installed. The Zend Host ID code can be obtained by using the zendid.exe utility. For more details, see Getting the Zend Host ID on page 37. REQUIRED if Hardware-Locked is set equal to YES. Meaningless if Hardware-Locked is set equal to NO. **Example:** Host-ID = H:MFM43-Q9CXC-B9EDX-GWYSU
Hardware-Locked	[YES | NO] option that indicates if the license will be locked to a specific machine using the Zend Host ID code(s). If set to YES the Host-ID is required. REQUIRED **Example:** Hardware-Locked = YES
UserDefinedField	OPTIONAL **Example:** Tea = Mint Flavor

Field Name	Description
X-UserDefinedField	User defined field prefixed with X-. Any additional information, which the user would like to add to the license file but not the signature. This information allows flexibility of input both prior to generating the signature and after. Any change to this information in the signature license-file, will have no impact on the validity of the license. OPTIONAL **Example:** X-Coffee = Black with no cream or sugar

Example License Definition Files

The following are samples of License Definitions Files.

Example 1

```
Product-Name = Drink
Registered-To = Bob
IP-Range = 10.1.*.*
Hardware-Locked = Yes
Host-ID = H:MFM43-Q9CXC-B9EDX-G /YSU
Host-ID = M:3QNKS-WPDD5-B3WU6-I FVZH
Expires = 09-Oct-2005
Tea = Mint Flavor
X-Coffee = Black with no cream or sug ar
```

Example 2

```
Product-Name = My Product
Registered-To = ABC Company
Hardware-Locked = No
```

```
Host-ID = Not-Locked
Expires = Never
X-Home0 = Region 1 Priority
X-Home1 = Region 8 Dismissed
X-Home2 = Region 12 will update nex  run
X-NextAppointment = See X-Home0
X-LastAppointment = See X-Home1
X-PromptRemoder = Hire a Janitor
ViolatedMyLicenseRules = False
```

Example 3

```
Product-Name = HIJK_Intranet
Registered-To = HIJK Company
IP-Range = 10.1.*.*
Hardware-Locked = No
Host-ID = Not-Locked
Expires = 02-Jul-2002
X-UsersOnline = GinaK,RO,RowlingsC, anice,Temp1,SysAdmin1
X-UnauthorizedCode = G5gH7^&*KkJ  00 by User ICNeilW at
15:30 June1
IllegalAccessDetected = False
HackerDetected = False
VirusDectected = False
```

zendenc_sign - Command

Command	**zendenc_sign**	
Synopsis	Command-line function for creating a signature license file from a license definition file.	
Syntax 1	**zendenc_sign** *LicenseDefinitionFile* *LicenseFileName*	
Arguments	*LicenseDefinitionFile*	File containing the source names and information in an equation format. *FieldName = FieldValue* For more information on how to create a license definition file, see License Definition File in the preceding section
	LicenseFileName	The name and extension to be given to the generated signature license file

Example Signature License File

The following is an example of a License file:

```
Registered-To = ABC Company
Hardware-Locked = No
Host-ID = Not-Locked
Expires = Never
X-Home0 = Region 1 Priority
X-Home1 = Region 8 Dismissed
X-Home2 = Region 12 will update nex  run
X-NextAppointment = See X-Home0
X-LastAppointment = See X-Home1
X-PromptRemoder = Hire a Janitor
ViolatedMyLicenseRules = False
Verification-Code =
wCcJ6mJMYH7vggYkS3m+3/dUL332al Ry0xtYnc55CC
TCtcxWXvuU5lOG4w==
```

Frequently Asked Questions

The following are common question and answers regarding encoding and encoded files. Additional FAQ and Support can be obtained at the Zend Website (www.zend.com).

Can encoded and non-encoded PHP files be used together?

Yes. Encoded and non-encoded PHP files, in most instances, can be used together transparently. The only exception is files, which were encoded using the **Work Only with Other Encoded Files** option. These files will only work with encoded files that were encoded by the same company.

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 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 Encoder that encoded the file). For additional information, see the "Avoiding Incompatibilities" section in the Zend Optimizer User Guide.

Can Zend Intermediate Code files be de-coded back into the PHP source file?

Encoded files are not 100% irreversible, because it may be possible to discover some of their functionality through various reverse-engineering techniques (tracing, raw code examination, guessing, etc.).

However, encoding with the Zend Encoder, `zendenc`, creates a file for which it would be impractical and inconvenient for anyone to access the proprietary code. Additionally, such an attempt will clearly infringe the software license. Moreover, the Zend Encoder first optimizes the PHP file (using the Zend Optimizer) before encoding it, thus introducing an additional level of complexity.

Do my clients need to install anything to run Zend-encoded files?

Yes, they need to install the Zend Optimizer as part of their PHP setup. The Zend Optimizer is available as a freeware download at the Zend Store, http://www.zend.com/store/freeware.php

What if my script is dependent on library file?

Files that are `include()'d` or `require()'d` in the script must be present at run time and will not be part of the encoded file. Both the encoded file and the library files should be shipped together. Such files can be either encoded or non-encoded – both file types can be used together transparently.

`Note: If Work Only with Other Encoded Files is used the library files must also be encoded by the same company.`

Keeping certain files as plain text can be useful if you wish to let users customize parts of the application.

My scripts use include_path or auto_prepend. Can I encode them without modification?

Yes. Since each file is encoded separately, all `include()'s` take place at run time and so do not interfere with the Zend Encoder.