phpSHIELD

The PHP Encoder

Version 8.1

Introduction

An introduction to phpSHIELD

This phpSHIELD User Manual covers all of the features necessary for you to encode and protect your scripts.

If there is anything that you feel has been omitted from this user manual, then please let us know as we are passionate about providing excellent service.

Have fun using your new product...

phpSHIELD 8.1

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Table of Contents

Part I

1 Introduction

1.1 About phpSHIELD

We created phpSHIELD to be a low-cost, but powerful, alternative to some of the more expensive encoding solutions out there. Our encoder is as good as, if not better in some respects, than other encoders currently selling for thousands of dollars. We pride ourselves on the quality of our products and have created phpSHIELD as a low-cost product that any developer - whether a freelancer or a corporation - can afford to protect and encode their software. The protected code is fully native bytecode, with an encryption layer on top of it and this means that it is not possible to reverse engineer back to the original script, as this script has been removed.

phpSHIELD was created by a global team of PHP experts, who are based in the USA, UK and Russia. Thank you for choosing our product - we hope that you enjoy using it, and if you have any suggestions, please let us know

With thanks

The phpSHIELD team

1.2 How to buy

To purchase phpSHIELD, please visit the following:

https://phpshield.com/purchase/

1.3 Features

phpSHIELD Features List

Fully native byte code encoding
Native encoding to fully protect your application

PHP5 and PHP4 support
Both PHP4 and PHP5 support built in to the encoder and loaders including the latest PHP 5.3.x

Fast encryption
Powerful and fast encryption to protect your files

Easy to use
A simple and easy to use interface. Three steps and your files are protected

Windows encoding client and loaders
Simple and easy to use Windows client encoder and loaders

Loaders for other operating systems
Encrypted and protected code works under Windows, Linux, Mac OS X, FreeBSD, NetBSD, OpenBSD, OpenSolaris

Protect your intellectual property

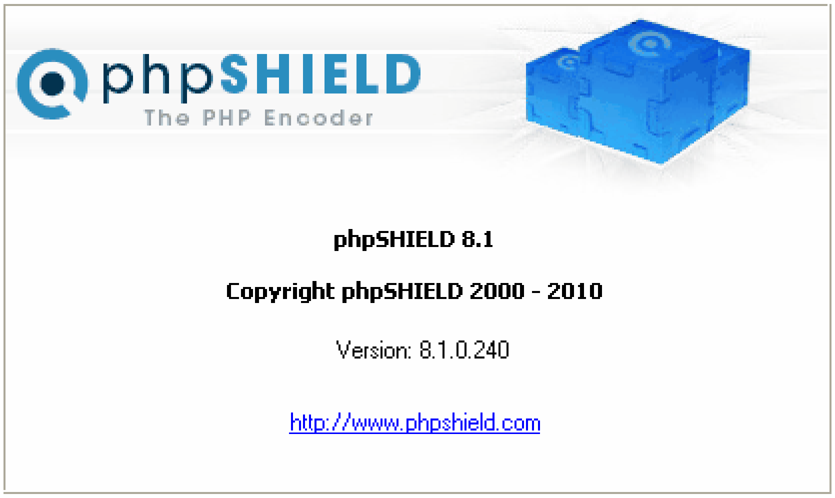
Protect your intellectual property. Protect your php source code.

Evaluation
We provide a Free 7 days evaluation of phpSHIELD

Part II

2 GUI manual

2.1 Overview



phpSHIELD 8.1

Copyright phpSHIELD 2000 - 2010

Version: 8.1.0.240

http://www.phpshield.com

phpSHIELD protects your PHP Source Code with a powerful, easy to use encoder, which creates a native bytecode version of the script and then encrypts it.

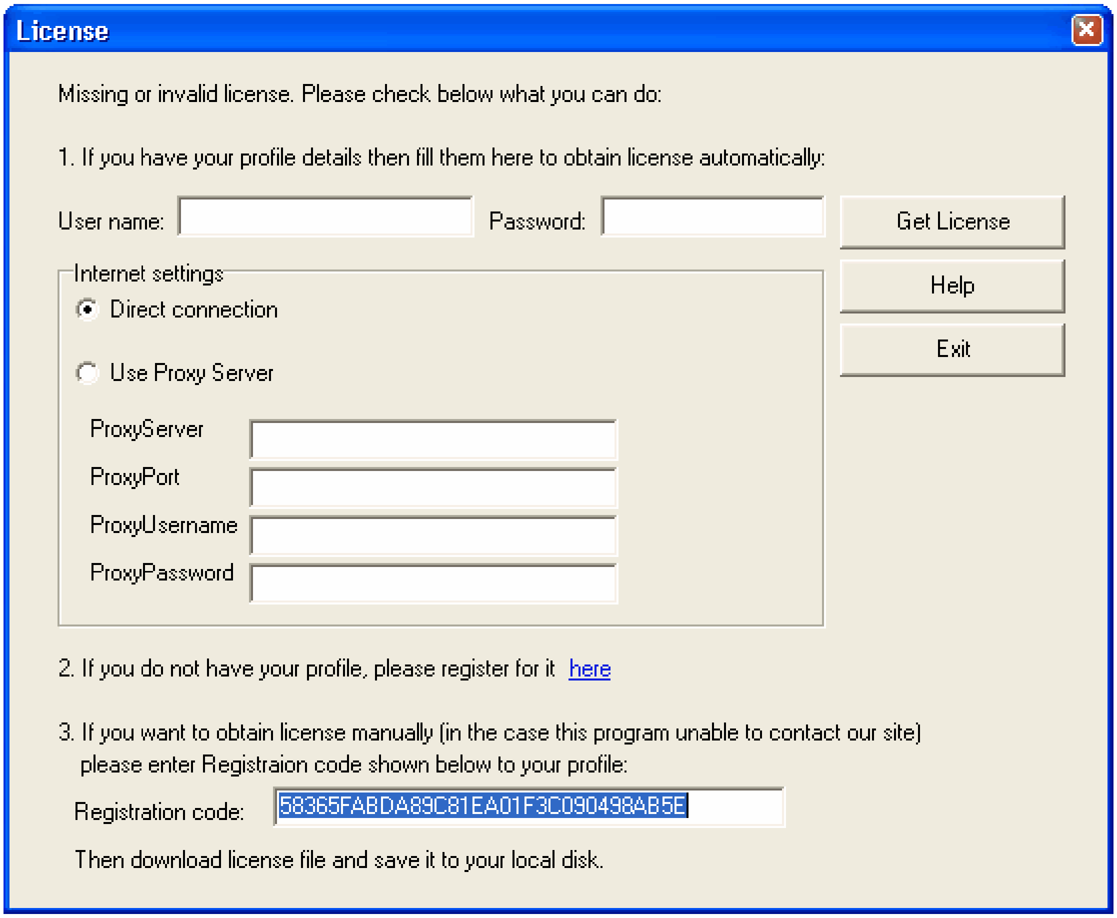
Some other PHP encoding systems do not use bytecode and it is very easy to reverse engineer their protected scripts back to the original source code. With bytecode encryption, it is never possible to do this as the original sourcecode has been completely removed from the protected file.

phpSHIELD has been developed to help software authors, freelance developers and anyone publishing scripts to protect their intellectual property with ease. Developers working on freelance projects do not have to give away all of their legacy code during a project - phpSHIELD will allow them to protect the elements at they do not want the client to have full access to. Software authors can release their scripts without the fear of losing revenue from piracy.

Encrypted and protected code works under Windows, Linux, Mac OS X, FreeBSD, NetBSD, OpenBSD, OpenSolaris.

2.2 First run - obtaining license

On your first of run of phpSHIELD you should see following screen:



Missing or invalid license. Please check below what you can do:

1. If you have your profile details then fill them here to obtain license automatically:

User name: Password:

Internet settings

● Direct connection

○ Use Proxy Server

ProxyServer

ProxyPort

ProxyUsername

ProxyPassword

2. If you do not have your profile, please register for it here

3. If you want to obtain license manually (in the case this program unable to contact our site) please enter Registraion code shown below to your profile:

Registration code: 58365FABDA89C81EA01F3C090498AB5E

Then download license file and save it to your local disk.

This screen means that you need to obtain a license first in order to run phpSHIELD.

Obtaining a license.

You need a direct connection to the internet or configure a Proxy server so that the phpSHIELD application can connect to the internet.

NOTE: Some firewalls may prohibit phpSHIELD from connecting to the internet, so you may have to enable internet access for phpSHIELD. On how to enable internet access for a custom application with your firewall please consult your firewall documentation.

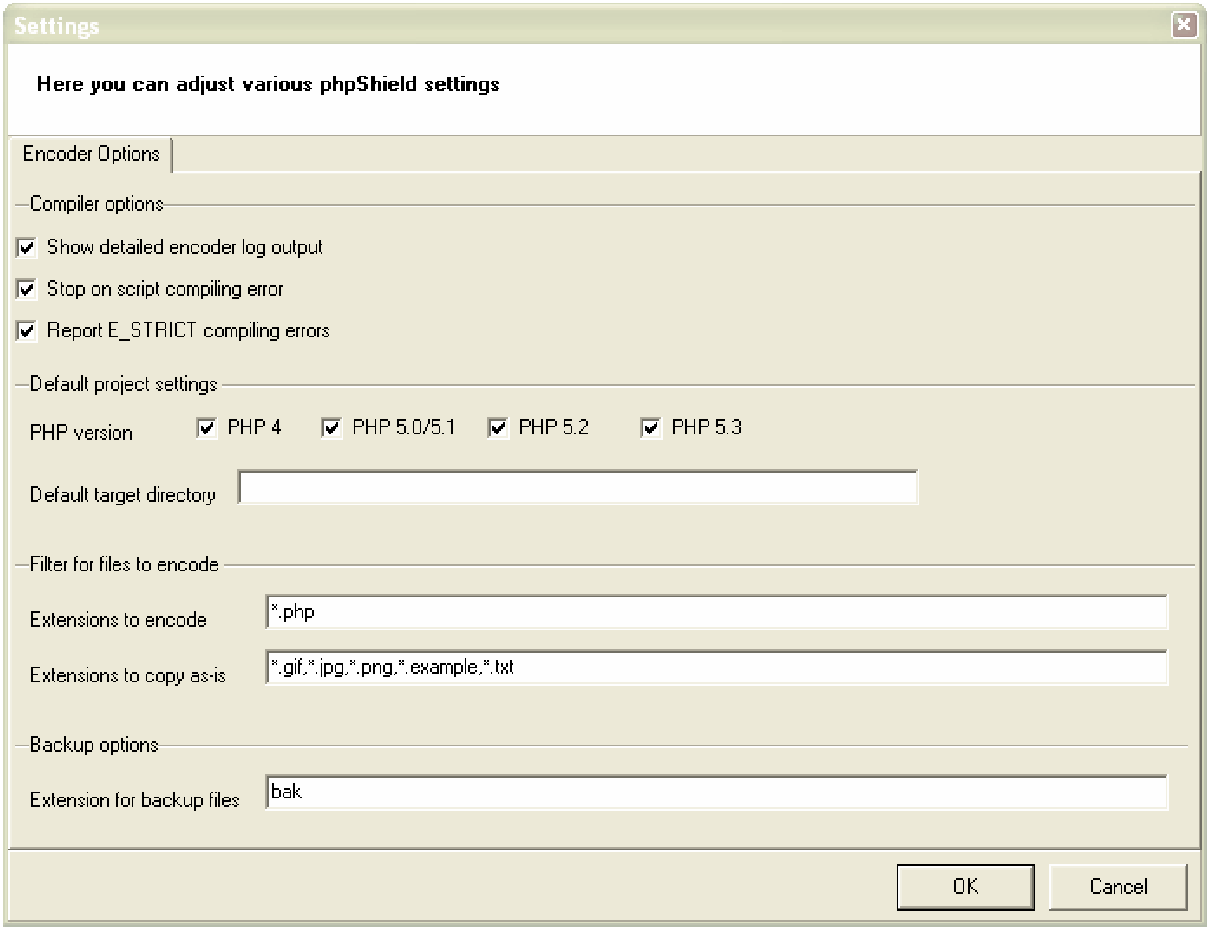
When you purchase the full version of phpSHIELD, or request demo version of it, you will receive an email with details on how to access your profile on our site. This email contains a user name and a profile password. Just type them into the 'User name' and 'Password' fields and click on the 'Get License' button. After the license has been downloaded you will see a message box saying: "Application will now close. Please start the application again to activate your copy". Press 'Ok' and then start the program again. If everything has installed correct, the application will start normally. If anything has gone wrong with the installtion, you will see the 'License error' screen again. Make sure you enter your user name and password correctly, check your internet settings and try again.

Configuring internet connection. This is useful if you normally need to access the internet via a proxy server.
Direct connection - This is selected by default and is used when you have a direct internet connection.

Use Proxy Server - Select this option if you are behind a Proxy Server. Enter the Proxy Server address and the Port. Also the Proxy Username and Password should be entered if it is applicable. Please consult your network administrator about information about your Proxy Server.

2.3 Settings

Encoder options



Settings

Here you can adjust various phpShield settings

Encoder Options

Compiler options

☑ Show detailed encoder log output

☑ Stop on script compiling error

☑ Report E_STRICT compiling errors

Default project settings

PHP version ☑ PHP 4 ☑ PHP 5.0/5.1 ☑ PHP 5.2 ☑ PHP 5.3

Default target directory

Filter for files to encode

Extensions to encode: *.php

Extensions to copy as-is: *.gif,*.jpg,*.png,*.example,*.txt

Backup options

Extension for backup files: bak

Compiler options

Show detailed encoder log output - enable this to show detailed encoder log during encoding process.
Stop on script compiling error - if this is enabled then encoder will stop encoding when it finds any error in the script
Report E_STRICT compiling errors - if this is enabled then errors of this type will be reported

Default project settings

This allows you to set up default project settings so that when you start the New Project then these settings will be the default for your projects.

Filter for files to encode

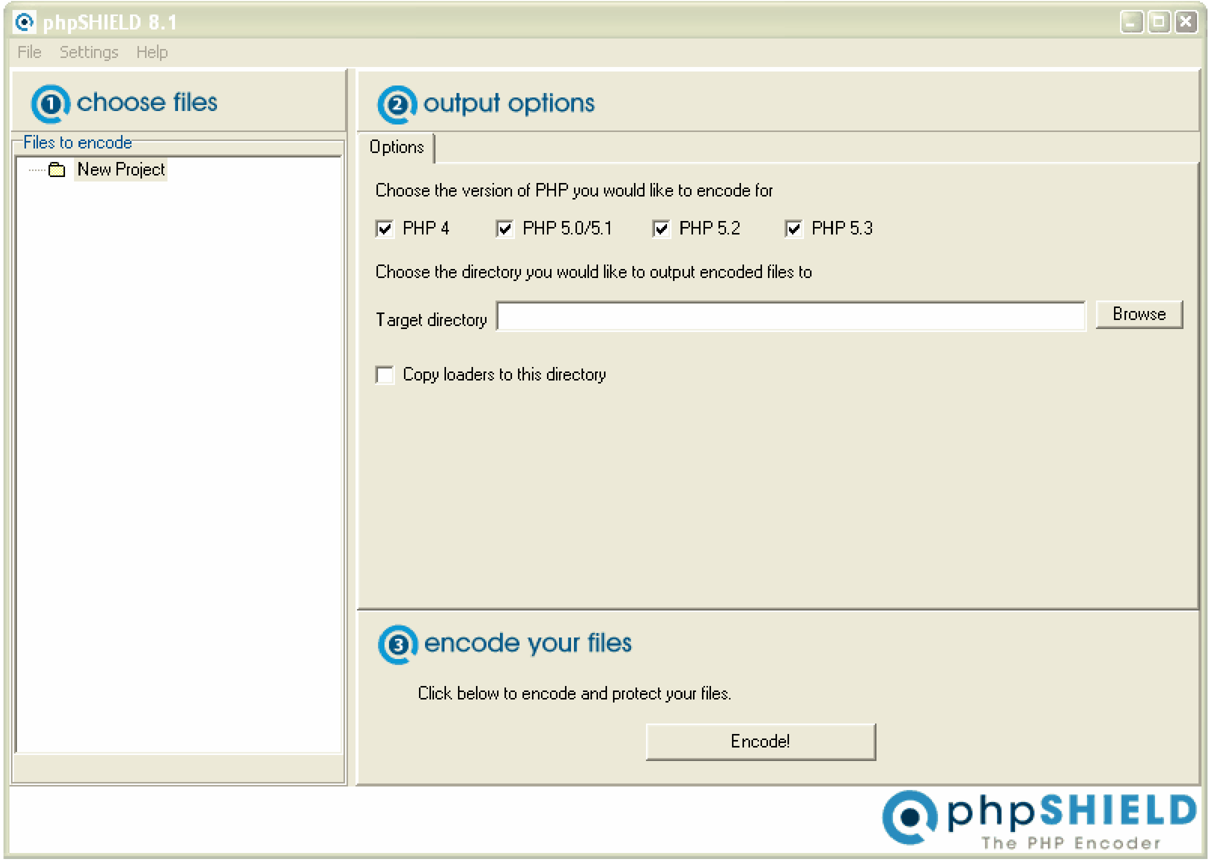
Extensions to encode - here you can list an extension that you wish to encode when you add files/ directories to your project. You should separate extensions by commas. For example: *.php,*.php4,*.inc

Extensions to copy as-is - here you can list extensions that will be marked to copy as-is (ie without any encoding) when you add files/directories to your project. You should separate extensions by commas. For example: *.html,*.tpl,*.txt

Backup options

This option is used when you want to encode files in the same place (with overwrite of source scripts). For example: bak

2.4 Encoding



Choose the version of PHP you would like to encode for

☑ PHP 4 ☑ PHP 5.0/5.1 ☑ PHP 5.2 ☑ PHP 5.3

Choose the directory you would like to output encoded files to

☐ Copy loaders to this directory

Click below to encode and protect your files.

Encoding process is simple like 1-2-3.

1.Choose files.

Use a mouse right click on the treeview to get a popup menu. There you can select what you would like to do (add file(s), add folder, rename, remove, remove all, create directory, mark file/dir to encode or to not encode).
Also you can drag and drop objects inside this treeview. Operations on multiple selected objects are allowed.

2. Output options.

Once you have added files into your project you need to select your output options.

- ***PHP versions.***

  Different versions of bytecodes will be packed into one protected script file. With phpSHIELD 8 you do not need to have separately encoded files for PHP 4 and PHP 5. You can specify target PHP versions instead.

  It's important that your source code should be compatible with all specified versions of PHP (or all versions including PHP 4 if the above option is not used). If your code is not compatible with any of the PHP versions then the encoder will report the error code [2] with the message "PHP syntax or other compiler error" for this file. We suggest that you run the problematic source php script with all of the required versions of PHP. The PHP engine will print a detailed error message if there are any problems. We have plans to improve the PHP error displaying in the encoder, but please use the above method for now

- Next, you need to specify a target directory where the encoded files will be placed.

- Also you may wish to place phpSHIELD loaders into this directory to deploy them with your encoded project.

3. Encode your files.

When you are ready just click the 'Encode!' button. The Encoding window appears saying that encoding has begun. Please wait until it say that encoding has finished. Then you can click the 'Close' button.

Now you can find the encoded files in the target directory that you specified at the 2nd step.

You may wish to save your project to use it later. Goto File->Save project to do it.

Part III

3 Protected script loaders

Protected script loaders are dynamically loaded PHP extensions which load the protected script, decrypt it and then run the bytecode. The Source code is never restored at any time, even in memory. There are different versions of the loaders available for different operating systems and PHP installations. The appropriate version of the loader will be automatically loaded by the protected script, if this is supported by your web server and PHP configuration.

phpSHIELD protected script loaders may also be installed manually in to a php.ini file - This is useful, for example, if automatic loading is not supported or if faster performance is required.

Protected Script Loaders will be updated periodically and the latest loaders are always freely available from:

http://www.phpshield.com/loaders/

3.1 Loader filename structure

The following provides an overview of the loader naming conventions:

ixed.X.Y.Zts.os

X.Y - major PHP version number (4.3 for 4.3.x, 5.0 for 5.0.x, 5.1 for 5.1.x, 5.2 for 5.2.x)
Z - minor PHP version number (2 for 5.0.2) This is optional and most loaders DOES NOT have the minor version number in the file name.

This part may be missed in the loader name which means that this loader is for all higher PHP versions:
ixed.4.3.lin - for all PHP 4.3.x versions
ixed.5.0.0.lin - for PHP 5.0.0 only
ixed.5.0.1.lin - for PHP 5.0.1 only
ixed.5.0.2.lin - for PHP 5.0.2 only
ixed.5.0.lin - for all PHP 5.0.3+ versions and higher
ixed.5.1.lin - for all PHP 5.1 versions and higher
ixed.5.2.lin - for all PHP 5.2 versions and higher
ixed.5.3.lin - for all PHP 5.3 versions and higher

ts. - this loader is suited for manual installation on systems with Thread Safety enabled PHP

os - three char code of operating system type. Currently supported:
.win - Windows
.lin - Linux
.fre - FreeBSD
.net - NetBSD
.ope - OpenBSD
.sun - SunOS
.dar - MacOSX (darwin)

For some operating systems there are different versions of loaders for 32-bit and 64-bit mode. File names of such loaders are the same as it is impossible to determine 32/64-bit mode on a PHP level. Although 32-bit and 64-bit loaders are packed in different zip (tar.gz, tar.bz2) files so you can easily determine them. You need to use the correct 32-bit or 64-bit version of the loader on your system according to the platform and the mode PHP executable or shared object is built. You **may safely try**

32-bit version and then 64-bit one if you are unsure. Usually you will get the following error message in the case of wrong 32/64-bit loader is installed: "Unable to load dynamic library 'ixed....' cannot open shared object file" or "Unable to load dynamic library 'ixed...' wrong ELF class: ELFCLASS32(64)". If you have access to a command line shell you may check the PHP using command line "file" tool, e.g. "file /path/to/php".

3.2 Automatic loading

Many operating systems and installations of PHP will load the Protected scripts without any modification. PHP will be able to find and load the appropriate loader if the following conditions are met:
1) Operating system and PHP mode:
Linux, FreeBSD, NetBSD, OpenBSD, MacOSX, SunOS, other UNIX - PHP installed as CGI or CLI
Linux, FreeBSD, NetBSD, OpenBSD, MacOSX, SunOS, other UNIX - PHP installed as a webserver's module (with thread safety off)
Windows - PHP installed as CGI or CLI
2) Thread Safety is disabled. You may check phpinfo() output for this.
3) dl() is enabled. You should have enable_dl=On in your php.ini.
4) The PHP extensions directory (extension_dir) needs to exist. Please check that the extension_dir= option in php.ini points to the real directory. Some hosting companies have incorrect installations of PHP and this can
cause problems.
5) The latest ixed loaders are installed in an ixed/ subdirectory within your scripts directory or any parent directory.
6) PHP version is below 5.2.5.

Please note: if your server and PHP configuration conform to all conditions above for automatic loading except only a PHP version, then it is enough to copy an appropriate loader into the PHP extension directory (extension_dir). The loader will be used automatically from the extension_dir directory - no need for changes in the php.ini configuration file.

3.3 Manual installation

It is possible to manually install the loader and this is required in the following conditions:
1) Operating system and PHP mode:
Linux, FreeBSD, NetBSD, OpenBSD, MacOSX, SunOS, other UNIX - PHP installed as a webserver's module (with thread safety on)
Windows - PHP installed as Apache module (thread safety is always on)
2) If Thread Safety is enabled. You may check phpinfo() output for this. PHP installed as a webserver's module under Windows will always have Thread Safety on.
3) If dl() is disabled. You have an "enable_dl=Off" setting in the php.ini configuration file.

Manual installation requires permissions to access the extension_dir directory and the php.ini configuration file. Manual installation may be used even if automatic loading is available. With appropriate and manually installed loaders you give the maximum performance for your protected scripts. This is because the script will not need to search for a loader each time it runs.

To install the loader manually you need to do the following:
1) Choose an appropriate loader for your operating system and version of PHP. Please refer to the "Loader filename structure" section below to know which loader is required for your operating system and version of PHP.
2) Find the loader file in the loaders/ subdirectory within phpSHIELD main installation directory and copy it to the PHP extension directory (extension_dir - check the phpinfo() output).
3) Find the location of the php.ini configuration file (check the phpinfo() output) and add "extension=ixed.

X.X.YYY" directive at the end of the file. (X.X is the major version of PHP and YYY is the name of operating system)
4) Restart the webserver in order to apply changes done in the php.ini configuration file and reload PHP.
5) Optionally you may check the phpinfo() output now to find out that phpSHIELD loader was successfully installed - search for "phpshield" within the output.

Part IV

4 Common mistakes

This section includes common mistakes that people may make, either in encoding and protecting their files, or in uploading or running these files on the web server. They are not in any particular order, but we would suggest that you look at this section before you contact phpSHIELD regarding any support matter.

4.1 Encoded scripts modification

Encoded scripts are protected against modification. Please **DO NOT MODIFY** any single byte in the encoded scripts or you will get an error executing them.

4.2 Extension directory (php.ini setting)

If you want loader to be loaded dynamically using dl() function you have to make sure that *extension_dir* setting in your php.ini is valid. It should point to a directory that does exists on the server. If it doesn't exists then PHP cannot load any extension at all (including phpshield loader).

For windows users only: *extension_dir* option in php.ini should point to the directory located on the same drive with your document root and scripts directory.

Part V

5 What is new in phpSHIELD 8.1

This is major release and it contains many fundamental changes and additions.

- FULL PHP5.3 support
- Now this is possible to encode the script so that it will run on PHP4.x/PHP5.x servers without the need to encode for them separately
- New loaders for other operating systems. Encoded phpSHIELD scripts now work under Windows, Linux, Mac OS X, FreeBSD, NetBSD, OpenBSD, OpenSolaris, including 64-bit platforms.
- Encoder is fully Windows Vista and Windows 7 compatible.
- Clicking on phpSHIELD project file now opens encoder GUI with this project loaded automatically

Index