PHP bytecode Compiler

Introduction

Warning
This extension is *EXPERIMENTAL*. The behaviour of this extension?including the names of its functions and any other documentation surrounding this extension?may change without notice in a future release of PHP. This extension should be used at your own risk.

Bcompiler was written for several reasons:

- To encode entire script in a proprietary PHP application
- To encode some classes and/or functions in a proprietary PHP application
- To enable the production of php-gtk applications that could be used on client desktops, without the need for a php.exe.
- To do the feasibility study for a PHP to C converter

The first of these goals is achieved using the bcompiler_write_header(), bcompiler_write_file() and bcompiler_write_footer() functions. The bytecode files can be written as either uncompressed or plain. To use the generated bytecode, you can simply include it with include or require statements.

The second of these goals is achieved using the bcompiler_write_header(), bcompiler_write_class(), bcompiler_write_footer(), bcompiler_read(), and bcompiler_load() functions. The bytecode files can be written as either uncompressed or plain. The bcompiler_load() reads a bzip compressed bytecode file, which tends to be 1/3 of the size of the original file.

To create EXE type files, bcompiler has to be used with a modified sapi file or a version of PHP which has been compiled as a shared library. In this scenario, bcompiler reads the compressed bytecode from the end of the exe file.

bcompiler can improve performance by about 30% when used with uncompressed bytecodes only. But keep in mind that uncompressed bytecode can be up to 5 times larger than the original source code. Using bytecode compression can save your space, but decompression requires much more time than parsing a source. bcompiler also does not do any bytecode optimization, this could be added in the future...

In terms of code protection, it is safe to say that it would be impossible to recreate the exact source code that it was built from, and without the accompanying source code comments. It would effectively be useless to use the bcompiler bytecodes to recreate and modify a class. However it is possible to retrieve data from a bcompiled bytecode file - so don't put your private passwords or anything in it.

Installing/Configuring

Requirements

No external libraries are needed to build this extension.

Installation

short installation note:

- You need at least PHP 4.3.0 for the compression to work
- To install on PHP 4.3.0 and later at the Unix command prompt type *pear install bcompiler*
- To install on Windows, until the binary package distribution mechanism is finished please search the archives of the pear-general mailing list for pre-built packages. (or send an email to it if you could not find a reference)
- To install on older versions you need to make some slight changes to the build.
- untar the *bcompiler.tgz* archive into *php4/ext*.(Get it directly from PECL » http://pecl.php.net/get/bcompiler)
- If the new directory is now called something like bcompiler-0.x, then you should rename it to bcompiler (except you only want to build it as self-contained php-module).
- If you are using versions before PHP 4.3.0, the you will need to copy the *Makefile.in.old* to *Makefile.in* and *config.m4.old* to *config.m4*.
- run *phpize* in *ext/bcompiler*
- run*./buildconf* in *php4*
- run *configure* with *--enable-bcompiler* (and your other options)
- *make; make install*
- that's it.

Runtime Configuration

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

Resource Types

This extension has no resource types defined.

Predefined Constants

This extension has no constants defined.

bcompiler Functions

Contact Information

If you have comments, bugfixes, enhancements or want to help developing this beast, you can drop me a mail at » alan_k@php.net. Any help is very welcome.

bcompiler_load_exe

bcompiler_load_exe -- Reads and creates classes from a bcompiler exe file

Description

bool **bcompiler_load_exe** (string `$filename`)

Reads data from a bcompiler exe file and creates classes from the bytecodes.

Parameters

`filename`
The exe file path, as a string.

Return Values

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

Examples

Example #1 - bcompiler_load() example

```
<?php

bcompiler_load_exe("/tmp/example.exe");
print_r(get_defined_classes());

?>
```

Notes

Warning

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

See Also

- bcompiler_load()

bcompiler_load

bcompiler_load -- Reads and creates classes from a bz compressed file

Description

bool **bcompiler_load** (string `$filename`)

Reads data from a bzcompressed file and creates classes from the bytecodes.

Parameters

`filename`
The bzcompressed file path, as a string.

Return Values

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

Examples

Example #2 - bcompiler_load() example

```
<?php

bcompiler_load("/tmp/example");

print_r(get_defined_classes());

?>
```

Notes

Warning

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

Note
Please use include or require statements to parse bytecodes, it's more portable and convenient way than using this function. Please note that this function won't execute script body code contained in the bytecode file.

See Also

- bcompiler_load_exe()

bcompiler_parse_class

bcompiler_parse_class -- Reads the bytecodes of a class and calls back to a user function

Description

bool **bcompiler_parse_class** (string `$class`, string `$callback`)

Reads the bytecodes of a class and calls back to a user function.

Parameters

`class`
The class name, as a string.

`callback`

Return Values

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

Examples

Example #3 - bcompiler_parse_class() **example**

```
<?php

function readByteCodes($data) {
   print_r($data);
}

bcompiler_parse_class("DB","readByteCodes");

?>
```

Notes

Warning

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

Note
This function has been removed from bcompiler and is no longer available as of bcompiler 0.5.

bcompiler_read

bcompiler_read -- Reads and creates classes from a filehandle

Description

bool **bcompiler_read** (resource `$filehandle`)

Reads data from a open file handle and creates classes from the bytecodes.

Parameters

`filehandle`
A file handle as returned by fopen().

Return Values

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

Examples

Example #4 - bcompiler_read() example

```
<?php
$fh = fopen("/tmp/example","r");
bcompiler_read($fh);
fclose($fh);
print_r(get_defined_classes());

?>
```

Notes

Warning

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

Note
Please use include or require statements to parse bytecodes, it's more portable and convenient way than using this function. Please note that this function won't execute script body code contained in the bytecode file.

bcompiler_write_class

bcompiler_write_class -- Writes an defined class as bytecodes

Description

bool **bcompiler_write_class** (resource `$filehandle`, string `$className` [, string `$extends`])

Reads the bytecodes from PHP for an existing class, and writes them to the open file handle.

Parameters

`filehandle`
A file handle as returned by fopen().

`className`
The class name, as a string.

`extends`

Return Values

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

Examples

Example #5 - bcompiler_write_class() example

```
<?php
$fh = fopen("/tmp/example","w");
bcompiler_write_header($fh);
bcompiler_write_class($fh,"DB");
// you must write DB_common before DB_mysql, as DB_mysql extends DB_common.
bcompiler_write_class($fh,"DB_common");
bcompiler_write_class($fh,"DB_mysql");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Notes

Warning

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

Note

This function does not perform dependency checking, so make sure you write the classes in an order that will not result in an *undefined class* error occurring when you load it.

See Also

- bcompiler_write_header()
- bcompiler_write_footer()

bcompiler_write_constant

bcompiler_write_constant -- Writes a defined constant as bytecodes

Description

bool **bcompiler_write_constant** (resource `$filehandle`, string `$constantName`)

Reads the bytecodes from PHP for an existing constant, and writes them to the open file handle.

Parameters

`filehandle`
A file handle as returned by fopen().

`constantName`
The name of the defined constant, as a string.

Return Values

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

Examples

Example #6 - bcompiler_write_constant() example

```
<?php
define("MODULE_MAX", 30);

$fh = fopen("/tmp/example","w");
bcompiler_write_header($fh);
bcompiler_write_constant($fh,"MODULE_MAX");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Notes

Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and

surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

See Also

- bcompiler_write_header()
- bcompiler_write_footer()

bcompiler_write_exe_footer

bcompiler_write_exe_footer -- Writes the start pos, and sig to the end of a exe type file

Description

bool **bcompiler_write_exe_footer** (resource `$filehandle`, int `$startpos`)

An EXE (or self executable) file consists of 3 parts:

- The stub (executable code, e.g. a compiled C program) that loads PHP interpreter, bcompiler extension, stored Bytecodes and initiates a call for the specified function (e.g. main) or class method (e.g. main::main)
- The Bytecodes (uncompressed only for the moment)
- The bcompiler EXE footer

To obtain a suitable stub you can compile php_embed-based stub phpe.c located in the examples/embed directory on bcompiler's CVS.

Parameters

`filehandle`
A file handle as returned by fopen().

`startpos`
The file position at which the Bytecodes start, and can be obtained using ftell().

Return Values

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

Examples

Example #7 - bcompiler_write_footer() example

```
<?php

/* creating the output file (example.exe) */
$fh = fopen("example.exe", "w");

/* 1) writing a stub (phpe.exe) */
$size = filesize("phpe.exe");
$fr = fopen("phpe.exe", "r");
fwrite($fh, fread($fr, $size), $size);
$startpos = ftell($fh);
```

```
/* 2) writing bytecodes */
bcompiler_write_header($fh);
bcompiler_write_class($fh, "myclass");
bcompiler_write_function($fh, "main");
bcompiler_write_footer($fh);

/* 3) writing EXE footer */
bcompiler_write_exe_footer($fh, $startpos);

/* closing the output file */
fclose($fh);
?>
```

Notes

Warning

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

See Also

- bcompiler_write_header()
- bcompiler_write_class()
- bcompiler_write_footer()

bcompiler_write_file

bcompiler_write_file -- Writes a php source file as bytecodes

Description

bool **bcompiler_write_file** (resource `$filehandle`, string `$filename`)

This function complies specified source file into bytecodes, and writes them to the open file handle.

Parameters

`filehandle`
A file handle as returned by fopen().

`filename`
The source file path, as a string.

Return Values

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

Examples

Example #8 - bcompiler_write_file() **example**

```
<?php
$fh = fopen("example.phb", "w");
bcompiler_write_header($fh);
bcompiler_write_file($fh, "example.php");
bcompiler_write_footer($fh);
fclose($fh);
/* the following should be equivalent:
include "example.php";
  and
include "example.phb";
*/
?>
```

Notes

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

See Also

- bcompiler_write_header()
- bcompiler_write_footer()

bcompiler_write_footer

bcompiler_write_footer -- Writes the single character \x00 to indicate End of compiled data

Description

bool **bcompiler_write_footer** (resource `$filehandle`)

Writes the single character \x00 to indicate End of compiled data.

Parameters

`filehandle`
A file handle as returned by fopen().

Return Values

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

Examples

Example #9 - bcompiler_write_footer() **example**

```
<?php
$fh = fopen("/tmp/example","w");
bcompiler_write_header($fh);
bcompiler_write_class($fh,"DB");
bcompiler_write_class($fh,"DB_common");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Notes

Warning

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

See Also

- bcompiler_write_header()

bcompiler_write_function

bcompiler_write_function -- Writes an defined function as bytecodes

Description

bool **bcompiler_write_function** (resource `$filehandle`, string `$functionName`)

Reads the bytecodes from PHP for an existing function, and writes them to the open file handle. Order is not important, (eg. if function b uses function a, and you compile it like the example below, it will work perfectly OK).

Parameters

`filehandle`
A file handle as returned by fopen().

`functionName`
The function name, as a string.

Return Values

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

Examples

Example #10 - bcompiler_write_function() example

```
<?php
$fh = fopen("/tmp/example","w");
bcompiler_write_header($fh);
bcompiler_write_function($fh,"my_function_a");
bcompiler_write_function($fh,"my_function_b");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Notes

Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and

surrounding documentation may change without notice in a future release of PHP. This function should be used at your own risk.

See Also

- bcompiler_write_header()
- bcompiler_write_footer()

bcompiler_write_functions_from_file

bcompiler_write_functions_from_file -- Writes all functions defined in a file as bytecodes

Description

bool **bcompiler_write_functions_from_file** (resource `$filehandle`, string `$fileName`)

Searches for all functions declared in the given file, and writes their correspondent bytecodes to the open file handle.

Parameters

`filehandle`
A file handle as returned by fopen().

`fileName`
The file to be compiled. You must always include or require the file you intend to compile.

Return Values

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

Examples

Example #11 - bcompiler_write_functions_from_file() example

```
<?php
require('module.php');

$fh = fopen("/tmp/example","w");
bcompiler_write_header($fh);
bcompiler_write_functions_from_file($fh,'module.php');
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Notes

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

See Also

- bcompiler_write_header()
- bcompiler_write_footer()

bcompiler_write_header

bcompiler_write_header -- Writes the bcompiler header

Description

bool **bcompiler_write_header** (resource `$filehandle` [, string `$write_ver`])

Writes the header part of a bcompiler file.

Parameters

`filehandle`
A file handle as returned by fopen().

`write_ver`
Can be used to write bytecode in a previously used format, so that you can use it with older versions of bcompiler.

Return Values

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

Examples

Example #12 - bcompiler_write_header() example

```
<?php
$fh = fopen("/tmp/example","w");
bcompiler_write_header($fh);
bcompiler_write_class($fh,"DB");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Notes

Warning

This function is *EXPERIMENTAL*. The behaviour of this function, its name, and surrounding documentation may change without notice in a future release of PHP. This

function should be used at your own risk.

See Also

- bcompiler_write_footer()

bcompiler_write_included_filename

bcompiler_write_included_filename -- Writes an included file as bytecodes

Description

bool **bcompiler_write_included_filename** (resource `$filehandle`, string `$filename`)

Warning
This function is currently not documented; only its argument list is available.

Return Values

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

Notes

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