

# Parsekit

# Introduction

These functions allow runtime analysis of opcodes compiled from PHP scripts.

# Installing/Configuring

## Requirements

No external libraries are needed to build this extension.

## Installation

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

Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: [» http://pecl.php.net/package/parsekit](#).

The DLL for this PECL extension may be downloaded from either the [» PHP Downloads](#) page or from [» http://pecl4win.php.net/](#)

## Runtime Configuration

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

## Resource Types

This extension has no resource types defined.

# Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

**PARSEKIT\_QUIET** ( [int](#) )

Return full detail, but without unnecessary NULL entries.

**PARSEKIT\_SIMPLE** ( [int](#) )

Return shorthand opcode notation.

**PARSEKIT\_EXTENDED\_VALUE** ( [int](#) )

Opcodes Flag

**PARSEKIT\_RESULT\_CONST** ( [int](#) )

Opcodes Flag

**PARSEKIT\_RESULT\_EA\_TYPE** ( [int](#) )

Opcodes Flag

**PARSEKIT\_RESULT\_JMP\_ADDR** ( [int](#) )

Opcodes Flag

**PARSEKIT\_RESULT\_OPARRAY** ( [int](#) )

Opcodes Flag

**PARSEKIT\_RESULT\_OPLINE** ( [int](#) )

Opcodes Flag

**PARSEKIT\_RESULT\_VAR** ( [int](#) )

Opcodes Flag

**PARSEKIT\_USAGE\_UNKNOWN** ( [int](#) )

Opcodes Flag

**PARSEKIT\_ZEND\_INTERNAL\_CLASS** ( [int](#) )

Class Type

**PARSEKIT\_ZEND\_USER\_CLASS** ( [int](#) )

Class Type

**PARSEKIT\_ZEND\_EVAL\_CODE** ( [int](#) )

Function Type

**PARSEKIT\_ZEND\_INTERNAL\_FUNCTION** ( [int](#) )

Function Type

**PARSEKIT\_ZEND\_OVERLOADED\_FUNCTION** ( [int](#) )

Function Type

**PARSEKIT\_ZEND\_OVERLOADED\_FUNCTION\_TEMPORARY** ( [int](#) ) PHP >= 5.0.0  
Function Type

**PARSEKIT\_ZEND\_USER\_FUNCTION** ( [int](#) )  
Function Type

**PARSEKIT\_IS\_CONST** ( [int](#) )  
Node Type

**PARSEKIT\_IS\_TMP\_VAR** ( [int](#) )  
Node Type

**PARSEKIT\_IS\_UNUSED** ( [int](#) )  
Node Type

**PARSEKIT\_IS\_VAR** ( [int](#) )  
Node Type

**PARSEKIT\_ZEND\_ADD** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ADD\_ARRAY\_ELEMENT** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ADD\_CHAR** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ADD\_INTERFACE** ( [int](#) ) PHP >= 5.0.0  
Opcode

**PARSEKIT\_ZEND\_ADD\_STRING** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ADD\_VAR** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ASSIGN** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ASSIGN\_ADD** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ASSIGN\_BW\_AND** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ASSIGN\_BW\_OR** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ASSIGN\_BW\_XOR** ( [int](#) )  
Opcode

**PARSEKIT\_ZEND\_ASSIGN\_CONCAT** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_DIM** ( [int](#) ) PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_DIV** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_MOD** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_MUL** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_OBJ** ( [int](#) ) PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_REF** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_SL** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_SR** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_ASSIGN\_SUB** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BEGIN\_SILENCE** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BOOL** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BOOL\_NOT** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BOOL\_XOR** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BRK** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BW\_AND** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BW\_NOT** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BW\_OR** ( [int](#) )

Opcode

**PARSEKIT\_ZEND\_BW\_XOR ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_CASE ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_CAST ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_CATCH ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_CLONE ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_CONCAT ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_CONT ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_DECLARE\_CLASS ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_DECLARE\_FUNCTION ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_DECLARE\_INHERITED\_CLASS ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_DIV ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_DO\_FCALL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_DO\_FCALL\_BY\_NAME ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_ECHO ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_END\_SILENCE ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_EXIT ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_EXT\_FCALL\_BEGIN ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_EXT\_FCALL\_END ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_EXT\_NOP ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_EXT\_STMT ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_CLASS ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_FETCH\_CONSTANT ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_FUNC\_ARG ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_IS ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_R ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_RW ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_TMP\_VAR ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_UNSET ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_DIM\_W ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_FUNC\_ARG ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_IS ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_OBJ\_FUNC\_ARG ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_OBJ\_IS ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_OBJ\_R ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_OBJ\_RW ( [int](#) )**

Opcode



**PARSEKIT\_ZEND\_FETCH\_OBJ\_UNSET ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_OBJ\_W ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_R ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_RW ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_UNSET ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FETCH\_W ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FE\_FETCH ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FE\_RESET ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_FREE ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_HANDLE\_EXCEPTION ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_IMPORT\_CLASS ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_IMPORT\_CONST ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_IMPORT\_FUNCTION ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_INCLUDE\_OR\_EVAL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_INIT\_ARRAY ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_INIT\_CTOR\_CALL ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_INIT\_FCALL\_BY\_NAME ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_INIT\_METHOD\_CALL ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_INIT\_STATIC\_METHOD\_CALL ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_INIT\_STRING ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_INSTANCEOF ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_ISSET\_ISEMPTY ( [int](#) )** PHP < 5.0.0

Opcode

**PARSEKIT\_ZEND\_ISSET\_ISEMPTY\_DIM\_OBJ ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_ISSET\_ISEMPTY\_PROP\_OBJ ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_ISSET\_ISEMPTY\_VAR ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_IS\_EQUAL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_IS\_IDENTICAL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_IS\_NOT\_EQUAL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_IS\_NOT\_IDENTICAL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_IS\_SMALLER ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_IS\_SMALLER\_OR\_EQUAL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMP ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMPNZ ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMPNZ\_EX ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMPZ ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMPZNZ ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMPZ\_EX ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_JMP\_NO\_CTOR ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_MOD ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_MUL ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_NEW ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_NOP ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_OP\_DATA ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_POST\_DEC ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_POST\_DEC\_OBJ ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_POST\_INC ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_POST\_INC\_OBJ ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_PRE\_DEC ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_PRE\_DEC\_OBJ ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_PRE\_INC ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_PRE\_INC\_OBJ ( [int](#) )** PHP >= 5.0.0

Opcode

**PARSEKIT\_ZEND\_PRINT ( [int](#) )**

Opcode

**PARSEKIT\_ZEND\_QM\_ASSIGN ( [int](#) )**

Opcod

**PARSEKIT\_ZEND\_RAISE\_ABSTRACT\_ERROR ( int )** PHP >= 5.0.0

Opcod

**PARSEKIT\_ZEND\_RECV ( int )**

Opcod

**PARSEKIT\_ZEND\_RECV\_INIT ( int )**

Opcod

**PARSEKIT\_ZEND\_RETURN ( int )**

Opcod

**PARSEKIT\_ZEND\_SEND\_REF ( int )**

Opcod

**PARSEKIT\_ZEND\_SEND\_VAL ( int )**

Opcod

**PARSEKIT\_ZEND\_SEND\_VAR ( int )**

Opcod

**PARSEKIT\_ZEND\_SEND\_VAR\_NO\_REF ( int )**

Opcod

**PARSEKIT\_ZEND\_SL ( int )**

Opcod

**PARSEKIT\_ZEND\_SR ( int )**

Opcod

**PARSEKIT\_ZEND\_SUB ( int )**

Opcod

**PARSEKIT\_ZEND\_SWITCH\_FREE ( int )**

Opcod

**PARSEKIT\_ZEND\_THROW ( int )** PHP >= 5.0.0

Opcod

**PARSEKIT\_ZEND\_TICKS ( int )**

Opcod

**PARSEKIT\_ZEND\_UNSET\_DIM\_OBJ ( int )**

Opcod

**PARSEKIT\_ZEND\_UNSET\_VAR ( int )**

Opcod

**PARSEKIT\_ZEND\_VERIFY\_ABSTRACT\_CLASS ( int )** PHP >= 5.0.0

Opcod

# Parsekit Functions

# parsekit\_compile\_file

parsekit\_compile\_file -- Compile a string of PHP code and return the resulting op array

## Description

array **parsekit\_compile\_file** ( string \$filename [, array &\$errors [, int \$options ] ] )

### Warning

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

## Parameters

*filename*

A string containing the name of the file to compile. Similar to the argument to **include()**.

*errors*

A 2D hash of errors (including fatal errors) encountered during compilation. Returned by reference.

*options*

One of either **PARSEKIT\_QUIET** or **PARSEKIT\_SIMPLE**. To produce varying degrees of verbosity in the returned output.

## Return Values

Returns a complex multi-layer array structure as detailed below.

## Examples

### Example #1 - [parsekit\\_compile\\_file\(\)](#) example

```
<?php
var_dump(parsekit_compile_file('hello_world.php', $errors,
PARSEKIT_SIMPLE));
?>
```

The above example will output:

```
array(5) {  
  [0]=>  
  string(37) "ZEND_ECHO UNUSED 'Hello World' UNUSED"  
  [1]=>  
  string(30) "ZEND_RETURN UNUSED NULL UNUSED"  
  [2]=>  
  string(42) "ZEND_HANDLE_EXCEPTION UNUSED UNUSED UNUSED"  
  ["function_table"]=>  
  NULL  
  ["class_table"]=>  
  NULL  
}
```

## See Also

- [parsekit\\_compile\\_string\(\)](#)

# parsekit\_compile\_string

parsekit\_compile\_string -- Compile a string of PHP code and return the resulting op array

## Description

array **parsekit\_compile\_string** ( string \$phpcode [, array &\$errors [, int \$options ] ] )

### Warning

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

## Parameters

*phpcode*

A string containing phpcode. Similar to the argument to [eval\(\)](#).

*errors*

A 2D hash of errors (including fatal errors) encountered during compilation. Returned by reference.

*options*

One of either **PARSEKIT\_QUIET** or **PARSEKIT\_SIMPLE**. To produce varying degrees of verbosity in the returned output.

## Return Values

Returns a complex multi-layer array structure as detailed below.

## Examples

### Example #2 - [parsekit\\_compile\\_string\(\)](#) example

```
<?php
    $ops = parsekit_compile_string('
echo "Foo\n";
', $errors, PARSEKIT_QUIET);

    var_dump($ops);
?>
```

The above example will output:



```

array(20) {
  ["type"]=>
  int(4)
  ["type_name"]=>
  string(14) "ZEND_EVAL_CODE"
  ["fn_flags"]=>
  int(0)
  ["num_args"]=>
  int(0)
  ["required_num_args"]=>
  int(0)
  ["pass_rest_by_reference"]=>
  bool(false)
  ["uses_this"]=>
  bool(false)
  ["line_start"]=>
  int(0)
  ["line_end"]=>
  int(0)
  ["return_reference"]=>
  bool(false)
  ["refcount"]=>
  int(1)
  ["last"]=>
  int(3)
  ["size"]=>
  int(3)
  ["T"]=>
  int(0)
  ["last_brk_cont"]=>
  int(0)
  ["current_brk_cont"]=>
  int(-1)
  ["backpatch_count"]=>
  int(0)
  ["done_pass_two"]=>
  bool(true)
  ["filename"]=>
  string(17) "Parsekit Compiler"
  ["opcodes"]=>
  array(3) {
    [8594800]=>
    array(5) {
      ["opcode"]=>
      int(40)
      ["opcode_name"]=>
      string(9) "ZEND_ECHO"
      ["flags"]=>
      int(768)
      ["op1"]=>
      array(3) {
        ["type"]=>
        int(1)
        ["type_name"]=>
        string(8) "IS_CONST"
        ["constant"]=>
        &string(4) "Foo"
      }
    }
  }
  ["lineno"]=>

```

```

        int(2)
    }
    ["859484C"]=>
    array(6) {
        ["opcode"]=>
        int(62)
        ["opcode_name"]=>
        string(11) "ZEND_RETURN"
        ["flags"]=>
        int(16777984)
        ["op1"]=>
        array(3) {
            ["type"]=>
            int(1)
            ["type_name"]=>
            string(8) "IS_CONST"
            ["constant"]=>
            &NULL
        }
        ["extended_value"]=>
        int(0)
        ["lineno"]=>
        int(3)
    }
    [8594898]=>
    array(4) {
        ["opcode"]=>
        int(149)
        ["opcode_name"]=>
        string(21) "ZEND_HANDLE_EXCEPTION"
        ["flags"]=>
        int(0)
        ["lineno"]=>
        int(3)
    }
}
}

```

## See Also

- [parsekit\\_compile\\_file\(\)](#)

# parsekit\_func\_arginfo

parsekit\_func\_arginfo -- Return information regarding function argument(s)

## Description

array **parsekit\_func\_arginfo** ( [mixed](#) \$function )

### Warning

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

## Parameters

*function*

A string describing a function, or an array describing a class/method.

## Return Values

Returns an array containing argument information.

## Examples

### Example #3 - [parsekit\\_func\\_arginfo\(\)](#) example

```
<?php
function foo($bar, stdClass $baz, &$bomb, $bling = false) {
}

var_dump(parsekit_func_arginfo('foo'));
?>
```

The above example will output:

```
array(4) {
  [0]=>
  array(3) {
    ["name"]=>
    string(3) "bar"
    ["allow_null"]=>
    bool(true)
    ["pass_by_reference"]=>
    bool(false)
```

```
}
[1]=>
array(4) {
    ["name"]=>
    string(3) "baz"
    ["class_name"]=>
    string(8) "stdClass"
    ["allow_null"]=>
    bool(false)
    ["pass_by_reference"]=>
    bool(false)
}
[2]=>
array(3) {
    ["name"]=>
    string(4) "bomb"
    ["allow_null"]=>
    bool(true)
    ["pass_by_reference"]=>
    bool(true)
}
[3]=>
array(3) {
    ["name"]=>
    string(5) "bling"
    ["allow_null"]=>
    bool(true)
    ["pass_by_reference"]=>
    bool(false)
}
}
```