

Object property and method call overloading

Introduction

The purpose of this extension is to allow overloading of object property access and method calls. Only one function is defined in this extension, [overload\(\)](#) which takes the name of the class that should have this functionality enabled. The class named has to define appropriate methods if it wants to have this functionality: `__get()`, `__set()` and `__call()` respectively for getting/setting a property, or calling a method. This way overloading can be selective. Inside these handler functions the overloading is disabled so you can access object properties normally.

Warning
This extension is <i>EXPERIMENTAL</i> . 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.

Warning
This extension is not a part of PHP 5. PHP 5 supports <code>__get()</code> , <code>__set()</code> and <code>__call()</code> natively. See the Overloading in PHP 5 page for more information.

Installing/Configuring

Requirements

No external libraries are needed to build this extension.

Installation

In order to use these functions, you must compile PHP with the *--enable-overload* option. Starting with PHP 4.3.0 this extension is enabled by default. You can disable overload support with *--disable-overload*.

The Windows version of PHP has built-in support for this extension. You do not need to load any additional extensions in order to use these functions.

Note
Builtin support for overload is available with PHP 4.3.0.

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.

Examples

Some simple examples on using the [overload\(\)](#) function:

Example #1 - Overloading a PHP class

```
<?php

class OO {
    var $a = 111;
    var $elem = array('b' => 9, 'c' => 42);

    // Callback method for getting a property
    function __get($prop_name, &$prop_value)
    {
        if (isset($this->elem[$prop_name])) {
            $prop_value = $this->elem[$prop_name];
            return true;
        } else {
            return false;
        }
    }

    // Callback method for setting a property
    function __set($prop_name, $prop_value)
    {
        $this->elem[$prop_name] = $prop_value;
        return true;
    }
}

// Here we overload the OO object
overload('OO');

$o = new OO;
echo "\$o->a: $o->a\n"; // print: $o->a: 111
echo "\$o->b: $o->b\n"; // print: $o->b: 9
echo "\$o->c: $o->c\n"; // print: $o->c: 42
echo "\$o->d: $o->d\n"; // print: $o->d:

// add a new item to the $elem array in OO
$o->x = 56;

// instantiate stdClass (it is built-in in PHP 4)
// $val is not overloaded!
$val = new stdClass;
$val->prop = 555;

// Set "a" to be an array with the $val object in it
// But __set() will put this in the $elem array
$o->a = array($val);
var_dump($o->a[0]->prop);

?>
```

Object overloading Functions

overload

overload -- Enable property and method call overloading for a class

Description

void overload (string `$class_name`)

The [overload\(\)](#) function will enable property and method call overloading for a class identified by `class_name`.

Parameters

class_name

The overloaded class name, as a string

Return Values

No value is returned.

Examples

See an [example](#) in the introductory section of this part.