

Classkit

Introduction

These functions allow the dynamic manipulation of PHP classes, at runtime.

Note
This extension has been replaced by runkit , which is not limited to class manipulation but has function manipulation, as well.

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/classkit](#).

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.

CLASSKIT_ACC_PRIVATE ([int](#))

Marks the method *private*

CLASSKIT_ACC_PROTECTED ([int](#))

Marks the method *protected*

CLASSKIT_ACC_PUBLIC ([int](#))

Marks the method *public*

Classkit Functions

classkit_import

classkit_import -- Import new class method definitions from a file

Description

array **classkit_import** (string \$filename)

Note

This function cannot be used to manipulate the currently running (or chained) method.

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

The filename of the class method definitions to import

Return Values

Associative array of imported methods

Examples

Example #1 - [classkit_import\(\)](#) example

```
<?php
// file: newclass.php
class Example {
    function foo() {
        return "bar!\n";
    }
}
?>

<?php
// requires newclass.php (see above)
```

```
class Example {
    function foo() {
        return "foo!\n";
    }
}

$e = new Example();

// output original
echo $e->foo();

// import replacement method
classkit_import('newclass.php');

// output imported
echo $e->foo();

?>
```

The above example will output:

```
foo!
bar!
```

See Also

- [classkit_method_add\(\)](#)
- [classkit_method_copy\(\)](#)

classkit_method_add

classkit_method_add -- Dynamically adds a new method to a given class

Description

bool **classkit_method_add** (string \$classname, string \$methodname, string \$args, string \$code [, int \$flags])

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

classname

The class to which this method will be added

methodname

The name of the method to add

args

Comma-delimited list of arguments for the newly-created method

code

The code to be evaluated when *methodname* is called

flags

The type of method to create, can be **CLASSKIT_ACC_PUBLIC**, **CLASSKIT_ACC_PROTECTED** or **CLASSKIT_ACC_PRIVATE**

Note

This parameter is only used as of PHP 5, because, prior to this, all methods were public.

Return Values

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

Examples

Example #2 - [classkit_method_add\(\)](#) example

```
<?php
class Example {
    function foo() {
        echo "foo!\n";
    }
}

// create an Example object
$e = new Example();

// Add a new public method
classkit_method_add(
    'Example',
    'add',
    '$num1, $num2',
    'return $num1 + $num2;',
    CLASSKIT_ACC_PUBLIC
);

// add 12 + 4
echo $e->add(12, 4);
?>
```

The above example will output:

16

See Also

- [classkit_method_copy\(\)](#)
- [classkit_method_redefine\(\)](#)
- [classkit_method_remove\(\)](#)
- [classkit_method_rename\(\)](#)
- [create_function\(\)](#)

classkit_method_copy

classkit_method_copy -- Copies a method from class to another

Description

bool **classkit_method_copy** (string \$dClass, string \$dMethod, string \$sClass [, string \$sMethod])

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

dClass

Destination class for copied method

dMethod

Destination method name

sClass

Source class of the method to copy

sMethod

Name of the method to copy from the source class. If this parameter is omitted, the value of *dMethod* is assumed.

Return Values

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

Examples

Example #3 - [classkit_method_copy\(\)](#) example

```
<?php
class Foo {
    function example() {
        return "foo!\n";
    }
}
```

```
}  
  
class Bar {  
    // initially, no methods  
}  
  
// copy the example() method from the Foo class to the Bar class, as baz()  
classkit_method_copy('Bar', 'baz', 'Foo', 'example');  
  
// output copied function  
echo Bar::baz();  
?>
```

The above example will output:

```
foo!
```

See Also

- [classkit_method_add\(\)](#)
- [classkit_method_redefine\(\)](#)
- [classkit_method_remove\(\)](#)
- [classkit_method_rename\(\)](#)

classkit_method_redefine

classkit_method_redefine -- Dynamically changes the code of the given method

Description

bool **classkit_method_redefine** (string \$classname, string \$methodname, string \$args, string \$code [, int \$flags])

Note

This function cannot be used to manipulate the currently running (or chained) method.

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

classname

The class in which to redefine the method

methodname

The name of the method to redefine

args

Comma-delimited list of arguments for the redefined method

code

The new code to be evaluated when *methodname* is called

flags

The redefined method can be **CLASSKIT_ACC_PUBLIC**, **CLASSKIT_ACC_PROTECTED** or **CLASSKIT_ACC_PRIVATE**

Note

This parameter is only used as of PHP 5, because, prior to this, all methods were public.

Return Values

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

Examples

Example #4 - [classkit_method_redefine\(\)](#) example

```
<?php
class Example {
    function foo() {
        return "foo!\n";
    }
}

// create an Example object
$e = new Example();

// output Example::foo() (before redefine)
echo "Before: " . $e->foo();

// Redefine the 'foo' method
classkit_method_redefine(
    'Example',
    'foo',
    '',
    'return "bar!\n";',
    CLASSKIT_ACC_PUBLIC
);

// output Example::foo() (after redefine)
echo "After: " . $e->foo();
?>
```

The above example will output:

```
Before: foo!
After: bar!
```

See Also

- [classkit_method_add\(\)](#)
- [classkit_method_copy\(\)](#)
- [classkit_method_remove\(\)](#)
- [classkit_method_rename\(\)](#)

classkit_method_remove

classkit_method_remove -- Dynamically removes the given method

Description

bool **classkit_method_remove** (string \$classname, string \$methodname)

Note

This function cannot be used to manipulate the currently running (or chained) method.

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

classname

The class in which to remove the method

methodname

The name of the method to remove

Return Values

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

Examples

Example #5 - [classkit_method_remove\(\)](#) example

```
<?php
class Example {
    function foo() {
        return "foo!\n";
    }

    function bar() {
        return "bar!\n";
    }
}
```

```
    }  
}  
  
// Remove the 'foo' method  
classkit_method_remove(  
    'Example',  
    'foo'  
);  
  
echo implode(' ', get_class_methods('Example'));  
  
?>
```

The above example will output:

```
bar
```

See Also

- [classkit_method_add\(\)](#)
- [classkit_method_copy\(\)](#)
- [classkit_method_redefine\(\)](#)
- [classkit_method_rename\(\)](#)

classkit_method_rename

classkit_method_rename -- Dynamically changes the name of the given method

Description

bool **classkit_method_rename** (string \$classname, string \$methodname, string \$newname)

Note

This function cannot be used to manipulate the currently running (or chained) method.

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

classname

The class in which to rename the method

methodname

The name of the method to rename

newname

The new name to give to the renamed method

Return Values

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

Examples

Example #6 - [classkit_method_rename\(\)](#) example

```
<?php
class Example {
    function foo() {
        return "foo!\n";
    }
}
```



```
    }  
}  
  
// Rename the 'foo' method to 'bar'  
classkit_method_rename(  
    'Example',  
    'foo',  
    'bar'  
);  
  
// output renamed function  
echo Example::bar();  
?>
```

The above example will output:

```
foo!
```

See Also

- [classkit_method_add\(\)](#)
- [classkit_method_copy\(\)](#)
- [classkit_method_redefine\(\)](#)
- [classkit_method_remove\(\)](#)