W32api

Introduction

This extension is a generic extension API to DLLs. This was originally written to allow access to the Win32 API from PHP, although you can also access other functions exported via other DLLs.

Currently supported types are generic PHP types (strings, booleans, floats, integers and nulls) and types you define with w32api_deftype().

Note
This extension has been moved to the » PECL repository and is no longer bundled with PHP as of PHP 5.1.0.

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.

Installing/Configuring

Requirements

This extension will only work on Windows systems.

Installation

There is no installation needed to use these functions; they are part of the PHP core.

Runtime Configuration

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

Resource Types

This extension defines one resource type, used for user defined types. The name of this resource is *"dynaparm"*.

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.

DC_MICROSOFT (integer)

DC_BORLAND (integer)

DC_CALL_CDECL (integer)

DC_CALL_STD (integer)

DC_RETVAL_MATH4 (integer)

DC_RETVAL_MATH8 (integer)

DC_CALL_STD_BO (integer)

DC_CALL_STD_MS (integer)

DC_CALL_STD_M8 (integer)

DC_FLAG_ARGPTR (integer)

Examples

This example gets the amount of time the system has been running and displays it in a message box.

Example #1 - Get the uptime and display it in a message box

```
<?php
// Define constants needed, taken from
// Visual Studio/Tools/Winapi/WIN32API.txt
define("MB_OK", 0);

// Load the extension in
dl("php_w32api.dll");

// Register the GetTickCount function from kernel32.dll
w32api_register_function("kernel32.dll",
                         "GetTickCount",
                         "long");

// Register the MessageBoxA function from User32.dll
w32api_register_function("User32.dll",
                         "MessageBoxA",
                         "long");

// Get uptime information
$ticks = GetTickCount();

// Convert it to a nicely displayable text
$secs  = floor($ticks / 1000);
$mins  = floor($secs / 60);
$hours = floor($mins / 60);

$str = sprintf("You have been using your computer for:" .
               "\r\n %d Milliseconds, or \r\n %d Seconds" .
               "or \r\n %d mins or\r\n %d hours %d mins.",
               $ticks,
               $secs,
               $mins,
               $hours,
               $mins - ($hours*60));

// Display a message box with only an OK button and the uptime text
MessageBoxA(NULL,
           $str,
           "Uptime Information",
           MB_OK);
?>
```

W32api Functions

w32api_deftype

w32api_deftype -- Defines a type for use with other w32api_functions

Description

bool **w32api_deftype** (string `$typename`, string `$member1_type`, string `$member1_name` [, string `$...` [, string `$...`]])

You need to call this function if you would like to define a type for a w32api call.

Parameters

`typename`
The name of the type.

`member1_type`
A member type can be a user defined type. All the type names are case sensitive. Built in type names should be provided in lowercase.

`member1_name`
The member name of *`member1_type`*.

`...`

`...`

This function takes 2n+1 arguments, where n is the number of members the type has. After that is the type of the member followed by the members name (in pairs).

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.

w32api_init_dtype

w32api_init_dtype -- Creates an instance of the data type typename and fills it with the values passed

Description

resource **w32api_init_dtype** (string `$typename`, mixed `$value` [, mixed `$...`])

This function creates an instance of the data type named *`typename`*, filling in the values of the data type.

Parameters

`typename`
: The *`typename`* parameter is case sensitive.

`value`
: You should give the values in the same order as you defined the data type with w32api_deftype().

`...`

Return Values

Returns a *dynaparm* resource.

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.

w32api_invoke_function

w32api_invoke_function -- Invokes function funcname with the arguments passed after the function name

Description

mixed **w32api_invoke_function** (string `$funcname`, mixed `$argument` [, mixed `$...`])

w32api_invoke_function() tries to find the previously registered function, passing the parameters you provided.

Parameters

`funcname`
The function name.

`argument`
Any of the arguments can be of any PHP type or w32api_deftype() defined type, as needed.

`...`

Return Values

The return type is the one you set when you registered the function, the value is the one returned by the function itself.

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.

w32api_register_function

w32api_register_function -- Registers function function_name from library with PHP

Description

bool **w32api_register_function** (string `$library`, string `$function_name`, string `$return_type`)

This function tries to find the *`function_name`* function in *`libary`*, and tries to import it into PHP.

Parameters

`library`
The library name, as a string.

`function_name`
The function name, as a string.

`return_type`
The function will be registered with the given *`return_type`*. This type can be a generic PHP type, or a type defined with w32api_deftype(). All type names are case sensitive. Built in type names should be provided in lowercase.

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.

w32api_set_call_method

w32api_set_call_method -- Sets the calling method used

Description

void **w32api_set_call_method** (int `$method`)

This function sets the method call type.

Parameters

`method`
Can be one of **DC_CALL_CDECL** or **DC_CALL_STD** (the extension default).

Return Values

No value is returned.

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.