

# SNMP

# Introduction

SNMP Functions.

# Installing/Configuring

## Requirements

In order to use the SNMP functions on Unix you need to install the » [NET-SNMP](#) package. On Windows these functions are only available on NT and not on Win95/98.

## Installation

Important: In order to use the UCD SNMP package, you need to define `NO_ZEROLENGTH_COMMUNITY` to `1` before compiling it. After configuring UCD SNMP, edit `config.h` or `acconfig.h` and search for `NO_ZEROLENGTH_COMMUNITY`. Uncomment the `#define` line. It should look like this afterwards:

```
#define NO_ZEROLENGTH_COMMUNITY 1  
Now compile PHP --with-snmp[=DIR].
```

If you see strange segmentation faults in combination with SNMP commands, you did not follow the above instructions. If you do not want to recompile UCD SNMP, you can compile PHP with the `--enable-ucd-snmp-hack` switch which will work around the misfeature.

The Windows distribution contains support files for SNMP in the `mibs` directory. This directory should be moved to `DRIVE:\usr\mibs`, where `DRIVE` must be replaced with the driveletter where PHP is installed on, e.g. `c:\usr\mibs`.

## 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.

**SNMP\_OID\_OUTPUT\_FULL** ( [integer](#) )

As of 5.2.0

**SNMP\_OID\_OUTPUT\_NUMERIC** ( [integer](#) )

As of 5.2.0

**SNMP\_VALUE\_LIBRARY** ( [integer](#) )

**SNMP\_VALUE\_PLAIN** ( [integer](#) )

**SNMP\_VALUE\_OBJECT** ( [integer](#) )

**SNMP\_BIT\_STR** ( [integer](#) )

**SNMP\_OCTET\_STR** ( [integer](#) )

**SNMP\_OPAQUE** ( [integer](#) )

**SNMP\_NULL** ( [integer](#) )

**SNMP\_OBJECT\_ID** ( [integer](#) )

**SNMP\_IPADDRESS** ( [integer](#) )

**SNMP\_COUNTER** ( [integer](#) )

**SNMP\_UNSIGNED** ( [integer](#) )

**SNMP\_TIMETICKS** ( [integer](#) )

**SNMP\_INTEGER** ( [integer](#) )

**SNMP\_INTEGER** ( [integer](#) )

**SNMP\_COUNTER64** ( [integer](#) )

# SNMP Functions

# snmp\_get\_quick\_print

snmp\_get\_quick\_print -- Fetches the current value of the UCD library's quick\_print setting

## Description

bool **snmp\_get\_quick\_print** ( void )

Returns the current value stored in the UCD Library for quick\_print. quick\_print is off by default.

## Return Values

Returns **TRUE** if quick\_print is on, **FALSE** otherwise.

## Examples

<b>Example #1 - <a href="#">snmp_get_quick_print()</a> example</b>
<pre>&lt;?php \$quickprint = snmp_get_quick_print(); ?&gt;</pre>

## See Also

- [snmp\\_set\\_quick\\_print\(\)](#) for a full description of what quick\_print does.

# snmp\_get\_valueretrieval

snmp\_get\_valueretrieval -- Return the method how the SNMP values will be returned

## Description

int **snmp\_get\_valueretrieval** ( void )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## See Also

- [snmp\\_set\\_valueretrieval\(\)](#)



# snmp\_read\_mib

snmp\_read\_mib -- Reads and parses a MIB file into the active MIB tree

## Description

bool **snmp\_read\_mib** ( string `$filename` )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

# snmp\_set\_enum\_print

snmp\_set\_enum\_print -- Return all values that are enums with their enum value instead of the raw integer

## Description

`void snmp_set_enum_print ( int $enum_print )`

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Notes

<b>Note</b>
<a href="#"><code>snmp_set_enum_print()</code></a> is only available when using the UCD SNMP library. This function is not available when using the Windows SNMP library.

# snmp\_set\_oid\_numeric\_print

snmp\_set\_oid\_numeric\_print -- Return all objects including their respective object id within the specified one

## Description

`void snmp_set_oid_numeric_print ( int $oid_numeric_print )`

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## ChangeLog

Version	Description
5.2.0	Since PHP 5.2.0, This function is an alias of: <a href="#">snmp_set_oid_output_format()</a> .

## See Also

- [snmp\\_set\\_oid\\_output\\_format\(\)](#)

# snmp\_set\_oid\_output\_format

snmp\_set\_oid\_output\_format -- Set the OID output format

## Description

**void** **snmp\_set\_oid\_output\_format** ( int \$oid\_format )

[snmp\\_set\\_oid\\_output\\_format\(\)](#) sets the output format to be full or numeric.

## Parameters

*oid\_format*

Set it to **SNMP\_OID\_OUTPUT\_FULL** if you want a full output,  
**SNMP\_OID\_OUTPUT\_NUMERIC** otherwise.

## Return Values

No value is returned.

## Notes

Note
<a href="#"><u>snmp_set_oid_output_format()</u></a> is only available when using the UCD SNMP library. This function is not available when using the Windows SNMP library.

# snmp\_set\_quick\_print

snmp\_set\_quick\_print -- Set the value of quick\_print within the UCD SNMP library

## Description

**void** snmp\_set\_quick\_print ( bool \$quick\_print )

Sets the value of quick\_print within the UCD SNMP library. When this is set (1), the SNMP library will return 'quick printed' values. This means that just the value will be printed. When quick\_print is not enabled (default) the UCD SNMP library prints extra information including the type of the value (i.e. IpAddress or OID). Additionally, if quick\_print is not enabled, the library prints additional hex values for all strings of three characters or less.

By default the UCD SNMP library returns verbose values, quick\_print is used to return only the value.

Currently strings are still returned with extra quotes, this will be corrected in a later release.

## Parameters

*quick\_print*

## Return Values

No value is returned.

## Examples

Setting quick\_print is often used when using the information returned rather than displaying it.

### Example #2 - Using [snmp\\_set\\_quick\\_print\(\)](#)

```
<?php
snmp_set_quick_print(0);
$a = snmpget("127.0.0.1", "public", ".1.3.6.1.2.1.2.2.1.9.1");
echo "$a\n";
snmp_set_quick_print(1);
$a = snmpget("127.0.0.1", "public", ".1.3.6.1.2.1.2.2.1.9.1");
echo "$a\n";
?>
```

The above example will output something similar to:

```
'Timeticks: (0) 0:00:00.00'  
'0:00:00.00'
```

## See Also

- [snmp\\_get\\_quick\\_print\(\)](#)

# snmp\_set\_valueretrieval

snmp\_set\_valueretrieval -- Specify the method how the SNMP values will be returned

## Description

`void snmp_set_valueretrieval ( int $method )`

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## See Also

- [snmp\\_get\\_valueretrieval\(\)](#)

# snmpget

snmpget -- Fetch an SNMP object

## Description

```
string snmpget ( string $hostname, string $community, string $object_id [, int $timeout [,  
int $retries ] ] )
```

The [snmpget\(\)](#) function is used to read the value of an SNMP object specified by the *object\_id*.

## Parameters

*hostname*  
The SNMP agent.

*community*  
The read community.

*object\_id*  
The SNMP object.

*timeout*

*retries*

## Return Values

Returns SNMP object value on success or **FALSE** on error.

## Examples

### Example #3 - Using [snmpget\(\)](#)

```
<?php  
$syscontact = snmpget("127.0.0.1", "public", "system.SysContact.0");  
?>
```

## See Also



- [snmpset\(\)](#)

# snmpgetnext

snmpgetnext -- Fetch a SNMP object

## Description

string **snmpgetnext** ( string \$host, string \$community, string \$object\_id [, int \$timeout [, int \$retries ] ] )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

# snmprealwalk

snmprealwalk -- Return all objects including their respective object ID within the specified one

## Description

array **snmprealwalk** ( string \$host, string \$community, string \$object\_id [, int \$timeout [, int \$retries ] ] )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

# snmpset

snmpset -- Set an SNMP object

## Description

```
bool snmpset ( string $hostname, string $community, string $object_id, string $type,  
mixed $value [, int $timeout [, int $retries ] ] )
```

[snmpset\(\)](#) is used to set the value of an SNMP object specified by the *object\_id*.

## Parameters

*hostname*  
The SNMP agent.

*community*  
The read community.

*object\_id*  
The SNMP object.

*type*

*value*

*timeout*

*retries*

## Return Values

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

## See Also

- [snmpget\(\)](#)

# snmpwalk

snmpwalk -- Fetch all the SNMP objects from an agent

## Description

array **snmpwalk** ( string \$hostname, string \$community, string \$object\_id [, int \$timeout  
[, int \$retries ] ] )

[snmpwalk\(\)](#) function is used to read all the values from an SNMP agent specified by the *hostname*.

## Parameters

*hostname*

The SNMP agent.

*community*

The read community.

*object\_id*

If **NULL**, *object\_id* is taken as the root of the SNMP objects tree and all objects under that tree are returned as an array. If *object\_id* is specified, all the SNMP objects below that *object\_id* are returned.

*timeout*

*retries*

## Return Values

Returns an array of SNMP object values starting from the *object\_id* as root or **FALSE** on error.

## Examples

### Example #4 - [snmpwalk\(\)](#) Example

```
<?php
$a = snmpwalk("127.0.0.1", "public", "");

foreach ($a as $val) {
    echo "$val\n";
}
```

```
}  
?>
```

Above function call would return all the SNMP objects from the SNMP agent running on localhost. One can step through the values with a loop

## See Also

- [snmpwalkoid\(\)](#)

# snmpwalkoid

snmpwalkoid -- Query for a tree of information about a network entity

## Description

array **snmpwalkoid** ( string \$hostname, string \$community, string \$object\_id [, int \$timeout [, int \$retries ] ] )

[snmpwalkoid\(\)](#) function is used to read all object ids and their respective values from an SNMP agent specified by *hostname*.

The existence of [snmpwalkoid\(\)](#) and [snmpwalk\(\)](#) has historical reasons. Both functions are provided for backward compatibility. Use [snmprealwalk\(\)](#) instead.

## Parameters

*hostname*

The SNMP agent.

*community*

The read community.

*object\_id*

If **NULL**, *object\_id* is taken as the root of the SNMP objects tree and all objects under that tree are returned as an array. If *object\_id* is specified, all the SNMP objects below that *object\_id* are returned.

*timeout*

*retries*

## Return Values

Returns an associative array with object ids and their respective object value starting from the *object\_id* as root or **FALSE** on error.

## Examples

Example #5 - <a href="#">snmpwalkoid()</a> Example
--

<pre>&lt;?php \$a = snmpwalkoid("127.0.0.1", "public", "");</pre>
---

```
for (reset($a); $i = key($a); next($a)) {  
    echo "$i: $a[$i]<br />\n";  
}  
?>
```

Above function call would return all the SNMP objects from the SNMP agent running on localhost. One can step through the values with a loop

## See Also

- [snmpwalk\(\)](#)