libxml

Introduction

These functions/constants are available since PHP 5.1.0 and if you have compiled one of the extensions based on libxml, like DOM, SimpleXML and XSLT.

Installing/Configuring

Requirements

This extension requires » libxml >= 2.6.0.

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 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.

LIBXML_COMPACT (integer)
Activate small nodes allocation optimization. This may speed up your application without needing to change the code.

Note
Only available in Libxml >= 2.6.21

LIBXML_DTDATTR (integer)
Default DTD attributes

LIBXML_DTDLOAD (integer)
Load the external subset

LIBXML_DTDVALID (integer)
Validate with the DTD

LIBXML_NOBLANKS (integer)
Remove blank nodes

LIBXML_NOCDATA (integer)
Merge CDATA as text nodes

LIBXML_NOEMPTYTAG (integer)
Expand empty tags (e.g. <*br*/> to <*br*></*br*>)

Note
This option is currently just available in the DOMDocument::save and DOMDocument::saveXML functions.

LIBXML_NOENT (integer)
Substitute entities

LIBXML_NOERROR (integer)
Suppress error reports

LIBXML_NONET (integer)
Disable network access when loading documents

LIBXML_NOWARNING (integer)

Suppress warning reports

LIBXML_NOXMLDECL (integer)
Drop the XML declaration when saving a document

Note
Only available in Libxml >= 2.6.21

LIBXML_NSCLEAN (integer)
Remove redundant namespaces declarations

LIBXML_XINCLUDE (integer)
Implement XInclude substitution

LIBXML_ERR_ERROR (integer)
A recoverable error

LIBXML_ERR_FATAL (integer)
A fatal error

LIBXML_ERR_NONE (integer)
No errors

LIBXML_ERR_WARNING (integer)
A simple warning

LIBXML_VERSION (integer)
libxml version like 20605 or 20617

LIBXML_DOTTED_VERSION (string)
libxml version like 2.6.5 or 2.6.17

libxml Functions

Predefined Classes

LibXMLError

Properties

- code - the error's code
- column - the column where the error occurred. Please note that this property isn't entirely implemented in libxml and therefore *0* is often returned.
- file - the filename, or empty if the XML was loaded from a string
- level - the severity of the error (one of the following constants: **LIBXML_ERR_WARNING**, **LIBXML_ERR_ERROR** or **LIBXML_ERR_FATAL**)
- line - the line where the error occurred
- message - the error message

libxml_clear_errors

libxml_clear_errors -- Clear libxml error buffer

Description

void **libxml_clear_errors** (void)

libxml_clear_errors() clears the libxml error buffer.

Return Values

No value is returned.

See Also

- libxml_get_errors()
- libxml_get_last_error()

libxml_get_errors

libxml_get_errors -- Retrieve array of errors

Description

array **libxml_get_errors** (void)

Retrieve array of errors.

Return Values

Returns an array with LibXMLError objects if there are any errors in the buffer, or an empty array otherwise.

Examples

Example #1 - A libxml_get_errors() example

This example demonstrates how to build a simple libxml error handler.

```
<?php

libxml_use_internal_errors(true);

$xmlstr = <<< XML
<?xml version='1.0' standalone='yes'?>
<movies>
<movie>
 <titles>PHP: Behind the Parser</title>
</movie>
</movies>
XML;

$doc = simplexml_load_string($xmlstr);
$xml = explode("\n", $xmlstr);

if (!$doc) {
   $errors = libxml_get_errors();

   foreach ($errors as $error) {
       echo display_xml_error($error, $xml);
   }

   libxml_clear_errors();
}


function display_xml_error($error, $xml)
{
   $return  = $xml[$error->line - 1] . "\n";
```

```
    $return .= str_repeat('-', $error->column) . "^\n";

    switch ($error->level) {
        case LIBXML_ERR_WARNING:
            $return .= "Warning $error->code: ";
            break;
         case LIBXML_ERR_ERROR:
            $return .= "Error $error->code: ";
            break;
        case LIBXML_ERR_FATAL:
            $return .= "Fatal Error $error->code: ";
            break;
    }

    $return .= trim($error->message) .
               "\n  Line: $error->line" .
               "\n  Column: $error->column";

    if ($error->file) {
        $return .= "\n  File: $error->file";
    }

    return "$return\n\n--------------------------------------------\n\n";
}

?>
```

The above example will output:

```
<titles>PHP: Behind the Parser</title>
^
Fatal Error 76: Opening and ending tag mismatch: titles line 4 and title
 Line: 4
 Column: 0

--------------------------------------------
```

See Also

- libxml_get_last_error()
- libxml_clear_errors()

libxml_get_last_error

libxml_get_last_error -- Retrieve last error from libxml

Description

LibXMLError **libxml_get_last_error** (void)

Retrieve last error from libxml.

Return Values

Returns a LibXMLError object if there is any error in the buffer, **FALSE** otherwise.

See Also

- libxml_get_errors()
- libxml_clear_errors()

libxml_set_streams_context

libxml_set_streams_context -- Set the streams context for the next libxml document load or write

Description

void **libxml_set_streams_context** (resource `$streams_context`)

Sets the streams context for the next libxml document load or write.

Parameters

`streams_context`
The stream context resource (created with stream_context_create())

Return Values

No value is returned.

Examples

Example #2 - A libxml_set_streams_context() example

```
<?php

$opts = array(
   'http' => array(
       'user_agent' => 'PHP libxml agent',
   )
);

$context = stream_context_create($opts);
libxml_set_streams_context($context);

// request a file through HTTP
$doc = DOMDocument::load('http://www.example.com/file.xml');

?>
```

See Also

- stream_context_create()

libxml_use_internal_errors

libxml_use_internal_errors -- Disable libxml errors and allow user to fetch error information as needed

Description

bool **libxml_use_internal_errors** ([bool `$use_errors`])

libxml_use_internal_errors() allows you to disable standard libxml errors and enable user error handling.

Parameters

`use_errors`
Whether to enable user error handling. Defaults to **FALSE**.

Return Values

This function returns the previous value of *`use_errors`*.

Examples

Example #3 - A libxml_use_internal_errors() example

This example demonstrates the basic usage of libxml errors and the value returned by this function.

```
<?php

// enable user error handling
var_dump(libxml_use_internal_errors(true));

$doc = DOMDocument::load('file.xml');

if (!$doc) {
   $errors = libxml_get_errors();
   foreach ($errors as $error) {
       // handle errors here
   }

   libxml_clear_errors();
}

?>
```

The above example will output:

```
bool(false)
```

See Also

- libxml_clear_errors()
- libxml_get_errors()