

## Bulletin Board Code

# Introduction

This extension aims to help parse BBCode text in order to convert it to HTML or another markup language. It uses one pass parsing and provides great speed improvement over the common approach based on regular expressions. Further more, it helps provide valid HTML by reordering open / close tags and by automatically closing unclosed tags.

Since 0.10.1 It supports argument quoting with single quotes, double quotes and HTML escaped double quotes.

# Installing/Configuring

## Requirements

No external libraries are needed to build this extension.

## Installation

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/bbcode](http://pecl.php.net/package/bbcode)

An alternative solution, written in PHP, is the PEAR package [» HTML\\_BBCodeParser](#).

## Runtime Configuration

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

## Resource Types

One resource is used in the BBCode extension: a BBCode\_Container returned by [bbcode\\_create\(\)](#).

# 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.

**BBCODE\_TYPE\_NOARG** ( [integer](#) )

This BBCode tag does not accept any arguments.

**BBCODE\_TYPE\_SINGLE** ( [integer](#) )

This BBCode tag does not have a corresponding close tag.

**BBCODE\_TYPE\_ARG** ( [integer](#) )

This BBCode tag need an argument.

**BBCODE\_TYPE\_OPTARG** ( [integer](#) )

This BBCode tag accept an optional argument.

**BBCODE\_TYPE\_ROOT** ( [integer](#) )

This BBCode tag is the special tag root (nesting level 0).

**BBCODE\_FLAGS\_ARG\_PARSING** ( [integer](#) )

This BBCode tag require argument sub-parsing (the argument is also parsed by the BBCode extension). As Of 0.10.2 another parser can be used as argument parser.

**BBCODE\_FLAGS\_CDATA\_NOT\_ALLOWED** ( [integer](#) )

This BBCode Tag does not accept content (it voids it automatically).

**BBCODE\_FLAGS\_SMILEYS\_ON** ( [integer](#) ) - since 0.10.2

This BBCode Tag accepts smileys.

**BBCODE\_FLAGS\_SMILEYS\_OFF** ( [integer](#) ) - since 0.10.2

This BBCode Tag does not accept smileys.

**BBCODE\_FLAGS\_ONE\_OPEN\_PER\_LEVEL** ( [integer](#) ) - since 0.10.2

This BBCode Tag automatically closes if another tag of the same type is found at the same nesting level.

**BBCODE\_FLAGS\_REMOVE\_IF\_EMPTY** ( [integer](#) ) - since 0.10.2

This BBCode Tag is automatically removed if content is empty it allows to produce lighter HTML.

**BBCODE\_FLAGS\_DENY\_REOPEN\_CHILD** ( [integer](#) ) - since 0.10.3

This BBCode Tag does not allow unclosed childs to reopen when automatically closed.

**BBCODE\_ARG\_DOUBLE\_QUOTE** ( [integer](#) ) - since 0.10.2

This is a parser option allowing argument quoting with double quotes (")

**BBCODE\_ARG\_SINGLE\_QUOTE** ( [integer](#) ) - since 0.10.2

This is a parser option allowing argument quoting with single quotes (')

**BBCODE\_ARG\_HTML\_QUOTE** ( [integer](#) ) - since 0.10.2

This is a parser option allowing argument quoting with HTML version of double quotes (&quot;)

**BBCODE\_AUTO\_CORRECT** ( [integer](#) ) - since 0.10.2

This is a parser option changing the way errors are treated. It automatically closes tag in the order they are opened. And treat tags with only an open tag as if there were a close tag present.

**BBCODE\_CORRECT\_REOPEN\_TAGS** ( [integer](#) ) - since 0.10.2

This is a parser option changing the way errors are treated. It automatically reopens tag if close tags are not in the good order.

**BBCODE\_DISABLE\_TREE\_BUILD** ( [integer](#) ) - since 0.10.2

This is a parser option disabling the BBCode parsing it can be useful if only the "smiley" replacement must be used.

**BBCODE\_DEFAULT\_SMILEYS\_ON** ( [integer](#) ) - since 0.10.2

This is a parser option setting smileys to ON if no flag is given at tag level.

**BBCODE\_DEFAULT\_SMILEYS\_OFF** ( [integer](#) ) - since 0.10.2

This is a parser option setting smileys to OFF if no flag is given at tag level.

**BBCODE\_FORCE\_SMILEYS\_OFF** ( [integer](#) ) - since 0.10.2

This is a parser option disabling completely the smileys parsing.

**BBCODE\_SMILEYS\_CASE\_INSENSITIVE** ( [integer](#) ) - since 0.10.3

Use a case insensitive Detection for smileys instead of a simple binary search.

**BBCODE\_SET\_FLAGS\_SET** ( [integer](#) ) - since 0.10.2

This permits to SET the complete flag set on a parser.

**BBCODE\_SET\_FLAGS\_ADD** ( [integer](#) ) - since 0.10.2

This permits to switch a flag set ON on a parser.

**BBCODE\_SET\_FLAGS\_REMOVE** ( [integer](#) ) - since 0.10.2

This permits to switch a flag set OFF on a parser.

## BBCode Functions

# bbcode\_add\_element

bbcode\_add\_element -- Adds a bbcode element

## Description

bool **bbcode\_add\_element** ( resource \$bbcode\_container, string \$tag\_name, array \$tag\_rules )

Adds a tag to an existing BBCode\_Container tag\_set using tag\_rules.

## Parameters

*bbcode\_container*

BBCode\_Container resource, returned by [bbcode\\_create\(\)](#).

*tag\_name*

The new tag to add to the BBCode\_Container tag\_set.

*tag\_rules*

An associative array containing the parsing rules; see [bbcode\\_create\(\)](#) for the available keys.

## Return Values

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

# bbcode\_add\_smiley

bbcode\_add\_smiley -- Adds a smiley to the parser

## Description

bool **bbcode\_add\_smiley** ( resource \$bbcode\_container, string \$smiley, string \$replace\_by )

Adds a smiley to the parser

## Parameters

*bbcode\_container*

BBCode\_Container resource, returned by [bbcode\\_create\(\)](#).

*smiley*

The string that will be replaced when found.

*replace\_by*

The string that replace smiley when found.

## Return Values

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

## Examples

### Example #1 - [bbcode\\_add\\_smiley\(\)](#) usage example

```
<?php
/*
 * Prepare the rule set
 */
$arrayBBCode=array(
    '='=>      array( 'type'=>BBCODE_TYPE_ROOT,
                    'childs'=>'!i' ),
    'b'=>      array( 'type'=>BBCODE_TYPE_NOARG,
                    'open_tag'=>'<b>',
                    'close_tag'=>'</b>' ),
    'u'=>      array( 'type'=>BBCODE_TYPE_NOARG,
                    'open_tag'=>'<u>',
                    'close_tag'=>'</u>',
                    'flags'=>BBCODE_FLAGS_SMILEYS_OFF ),
    'i'=>      array( 'type'=>BBCODE_TYPE_NOARG,
                    'open_tag'=>'<i>',
                    'close_tag'=>'</i>' ,
```



```

        'childs'=>'b'),
    );
/*
 * Parsed Text
 */
$text=<<<EOF
[i] No parse Test [/i] :)
[b] Parsed, with smiley :( [/b]
[u] Parsed, with no smiley :D [/u]
EOF;
/*
 * Init the parser
 */
$BBHandler=bbcode_create($arrayBBCode);
/*
 * Add Smiley rules to parser
 */
bbcode_add_smiley($BBHandler, ":", "<img src=\"smiley.gif\" alt=\":)\" />");
bbcode_add_smiley($BBHandler, ":(", "<img src=\"sad.gif\" alt=\":(\" />");
bbcode_add_smiley($BBHandler, ":D", "<img src=\"happy.gif\" alt=\":D\" />");
bbcode_add_smiley($BBHandler, ":p", "<img src=\"tong.gif\" alt=\":p\" />");
bbcode_add_smiley($BBHandler, ":|", "<img src=\"special.gif\" alt=\":|\" />");
bbcode_add_smiley($BBHandler, ":6:", "<img src=\"six.gif\" alt=\":6:\" />");
/*
 * Parse the text
 */
echo bbcode_parse($BBHandler,$text);
?>

```

The above example will output:

```

[i] No parse Test [/i] 
<b> Parsed, with smiley  </b>
<u> Parsed, with no smiley :D </u>

```

# bbcode\_create

bbcode\_create -- Create a BBCode Resource

## Description

resource **bbcode\_create** ( [ array \$bbcode\_initial\_tags ] )

This function returns a new BBCode Resource used to parse BBCode strings.

## Parameters

*bbcode\_initial\_tags*

An associative array containing the tag names as keys and parameters required to correctly parse BBCode as their value. The following key/value pairs are supported:

- *flags* optional - a flag set based on the BBCode\_FLAGS\_\* constants.
- *type* required - an int indicating the type of tag. Use the BBCode\_TYPE\_\* constants.
- *open\_tag* required - the HTML replacement string for the open tag.
- *close\_tag* required - the HTML replacement string for the close tag.
- *default\_arg* optional - use this value as the default argument if none is provided and tag\_type is of type OPTARG.
- *content\_handling* optional - Gives the callback used for modification of the content. Object Oriented Notation supported only since 0.10.1 callback prototype is string name(string \$content, string \$argument)
- *param\_handling* optional - Gives the callback used for modification of the argument. Object Oriented Notation supported only since 0.10.1 callback prototype is string name(string \$content, string \$argument)
- *childs* optional - List of accepted child for the tag. The format of the list is a comma separated string. If the list starts with ! it will be the list of rejected child for the tag.
- *parent* optional - List of accepted child for the tag. The format of the list is a comma separated string.

## Return Values

Returns a BBCode\_Container

## Examples

## Example #2 - [bbcode\\_create\(\)](#) example

```
<?php
$arrayBBCode=array(
    ''=>        array('type'=>BBCODE_TYPE_ROOT,  'childs'=>'!i'),
    'i'=>        array('type'=>BBCODE_TYPE_NOARG, 'open_tag'=>'<i>',
                      'close_tag'=>'</i>', 'childs'=>'b'),
    'url'=>       array('type'=>BBCODE_TYPE_OPTARG,
                      'open_tag'=>'<a href="{PARAM}">', 'close_tag'=>'</a>',
                      'default_arg'=>' {CONTENT}',
                      'childs'=>'b,i'),
    'img'=>       array('type'=>BBCODE_TYPE_NOARG,
                      'open_tag'=>'<img src="", 'close_tag'=>'>' />',
                      'childs'=>''),
    'b'=>        array('type'=>BBCODE_TYPE_NOARG, 'open_tag'=>'<b>',
                      'close_tag'=>'</b>'),
);
$text=<<<EOF
[b]Bold Text[/b]
[i]Italic Text[/i]
[url]http://www.php.net[/url]
[url=http://pecl.php.net/][b]Content Text[/b][/url]
[img]http://static.php.net/www.php.net/images/php.gif[/img]
[url=http://www.php.net/]
[img]http://static.php.net/www.php.net/images/php.gif[/img]
[/url]
EOF;
$BBHandler=bbcode_create($arrayBBCode);
echo bbcode_parse($BBHandler,$text);
?>
```

The above example will output:

```
<b>Bold Text</b>
<i>Italic Text</i>
<a href="http://www.php.net/">http://www.php.net/</a>
<a href="http://pecl.php.net/"><b>Content Text</b></a>

<a href="http://www.php.net/">
[img]http://static.php.net/www.php.net/images/php.gif[/img]
</a>
```

# bbcode\_destroy

bbcode\_destroy -- Close BBCode\_container resource

## Description

bool **bbcode\_destroy** ( resource \$bbcode\_container )

This function closes the resource opened by [bbcode\\_create\(\)](#).

## Parameters

*bbcode*

BBCode\_Container resource returned by [bbcode\\_create\(\)](#).

## Return Values

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

# bbcode\_parse

bbcode\_parse -- Parse a string following a given rule set

## Description

string **bbcode\_parse** ( resource \$bbcode\_container, string \$to\_parse )

This function parse the string to\_parse following the rules in the bbcode\_container created by [bbcode\\_create\(\)](#).

## Parameters

*bbcode\_container*

BBCode\_Container resource returned by [bbcode\\_create\(\)](#).

*to\_parse*

The string we need to parse.

## Return Values

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



```

BBCODE_FLAGS_REMOVE_IF_EMPTY | BBCODE_FLAGS_SMILEYS_OFF),
    'i'=>
        array('type'=>BBCODE_TYPE_NOARG,
                'open_tag'=>'<i>', 'close_tag'=>'</i>',
                'flags'=>BBCODE_FLAGS_REMOVE_IF_EMPTY),
);
/*
* Generating bbcode ruleset for argument parser
*/
$arrayBBCode_arg=array(
    'b'=>
        array('type'=>BBCODE_TYPE_NOARG,
                'open_tag'=>'<b class="sub">', 'close_tag'=>'</b>',
                'flags'=>BBCODE_FLAGS_REMOVE_IF_EMPTY),
    'u'=>
        array('type'=>BBCODE_TYPE_NOARG,
                'open_tag'=>'<u>', 'close_tag'=>'</u>',
                'flags'=>BBCODE_FLAGS_SMILEYS_OFF |
BBCODE_FLAGS_REMOVE_IF_EMPTY | BBCODE_FLAGS_SMILEYS_OFF),
    'i'=>
        array('type'=>BBCODE_TYPE_NOARG,
                'open_tag'=>'<i>', 'close_tag'=>'</i>',
                'flags'=>BBCODE_FLAGS_REMOVE_IF_EMPTY),
);
/*
* Text we are going to parse
*/
$text=<<<EOF
[quote="[b]Test[/b]"]
Foo :)
[/quote]
[b]Bar example :)[/b] :)
EOF;
/*
* Init the two parsers
*/
$BBHandler=bbcode_create($arrayBBCode);
$BBArgHandler=bbcode_create($arrayBBCode_arg);
/*
* Setting Flags on the parsers
*/
bbcode_set_flags($BBHandler,

BBCODE_CORRECT_REOPEN_TAGS|BBCODE_DEFAULT_SMILEYS_ON|BBCODE_ARG_DOUBLE_QUOTE
|

BBCODE_ARG_SINGLE_QUOTE|BBCODE_ARG_HTML_QUOTE,BBCODE_SET_FLAGS_SET);
bbcode_set_flags($BBArgHandler,

BBCODE_CORRECT_REOPEN_TAGS|BBCODE_DEFAULT_SMILEYS_ON|BBCODE_ARG_DOUBLE_QUOTE
|

BBCODE_ARG_SINGLE_QUOTE|BBCODE_ARG_HTML_QUOTE,BBCODE_SET_FLAGS_SET);
/*
* Setting $BBArgHandler as the BBHandler argument parser
*/
bbcode_set_arg_parser($BBHandler,$BBArgHandler);
/*
* Adding Smileys handling rules to Main parser
*/
bbcode_add_smiley($BBHandler, " :) ", "<img src=\"smiley.gif\" alt=\" :) \"
/>");
/*
* Use the main parser to parse text

```

```
*/  
echo bbcode_parse($BBHandler,$text);  
?>
```

The above example will output:

```
<quote><h4>Source: <b class="sub">Test</b></h4>  
Foo   
</quote>  
<b>Bar example :)</b> 
```



# bbcode\_set\_flags

bbcode\_set\_flags -- Set or alter parser options

## Description

bool **bbcode\_set\_flags** ( resource \$bbcode\_container, int \$flags [, int \$mode ] )

Set or alter parser options

## Parameters

*bbcode\_container*

BBCode\_Container resource, returned by [bbcode\\_create\(\)](#).

*flags*

The flag set that must be applied to the bbcode\_container options

*mode*

One of the BBCODE\_SET\_FLAGS\_\* constant to set, unset a specific flag set or to replace the flag set by flags.

## Return Values

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

## Examples

### Example #4 - [bbcode\\_set\\_flags\(\)](#) usage example

```
<?php
/*
 * Preparing RuleSet
 */
$arrayBBCode=array(
    'b'=>      array('type'=>BBCODE_TYPE_NOARG,
                    'open_tag'=>'<b>', 'close_tag'=>'</b>'),
    'u'=>      array('type'=>BBCODE_TYPE_NOARG,
                    'open_tag'=>'<u>', 'close_tag'=>'</u>'),
    'i'=>      array('type'=>BBCODE_TYPE_NOARG,
                    'open_tag'=>'<i>', 'close_tag'=>'</i>'),
);
/*
 * Paired incorrectly nested BBCode
 */
$text="[i] Parser [b] Auto Correction [/i] at work [/b]\n";
$BBHandler=bbcode_create($arrayBBCode);
```

```
echo bbcode_parse($BBHandler,$text);
// Enabling reopening of automatically closed elements
bbcode_set_flags($BBHandler,BBCODE_CORRECT_REOPEN_TAGS,
    BBCODE_SET_FLAGS_SET);
echo bbcode_parse($BBHandler,$text);

/*
* Unpaired incorrectly nested BBCode
*/
$text="[i] Parser [b] Auto Correction [/i] at work\n";
echo bbcode_parse($BBHandler,$text);
// Enabling automatic close of pending tags
bbcode_set_flags($BBHandler,
    BBCODE_CORRECT_REOPEN_TAGS|BBCODE_AUTO_CORRECT,
    BBCODE_SET_FLAGS_SET);
echo bbcode_parse($BBHandler,$text);
?>
```

The above example will output:

```
<i> Parser <b> Auto Correction </b></i> at work
<i> Parser <b> Auto Correction </b></i><b> at work </b>
<i> Parser [b] Auto Correction </i> at work
<i> Parser <b> Auto Correction </b></i><b> at work
</b>
```