mb-scrub

mb_scrub

Replace ill-formed byte sequences with the substitute character

ENTITY

string

mb_scrub

string

string

string

null

encoding

ENTITY

Perform a character set conversion from the specified encoding, or the default encoding if no encoding was specified, to the same encoding. This has the effect of replacing any invalid byte sequences with the substitute character.

ENTITY

`string`
The input string.

`encoding`
The encoding used to interpret `string`. If it is omitted or ENTITY, the mbstring.internal_encoding setting will be used if set, otherwise the default_charset setting will be used.

ENTITY

The ENTITY result with invalid byte sequences replaced.

ENTITY ENTITY

ENTITY

ENTITY	ENTITY

ENTITY

Byte-level replacement performed by `mb_scrub`

`bin2hex` is used here because terminals, browsers and fonts may render an ill-formed byte sequence with a replacement character of their own, which hides what the string actually contains.

```
<?php

// The byte 0xFF cannot appear in a valid UTF-8 string.
$input = "A\xFFB";
echo bin2hex($input), "\n";

// The default substitute character is "?" (0x3F).
echo bin2hex(mb_scrub($input, 'UTF-8')), "\n";

// U+FFFD REPLACEMENT CHARACTER is encoded as EF BF BD in UTF-8.
mb_substitute_character(0xFFFD);
echo bin2hex(mb_scrub($input, 'UTF-8')), "\n";

?>
```

ENTITY

```
41ff42
413f42
41efbfbd42
```

Using `mb_scrub` before UTF-8 aware processing

PCRE patterns using the `u` modifier reject subjects that are not well-formed UTF-8. Scrubbing the input first makes it acceptable.

```
<?php

$input = "A\xFFB";

var_dump(preg_match_all('/./us', $input));
echo preg_last_error_msg(), "\n";

$clean = mb_scrub($input, 'UTF-8');

var_dump(preg_match_all('/./us', $clean));

?>
```

ENTITY

```
bool(false)
Malformed UTF-8 characters, possibly incorrectly encoded
int(3)
```

ENTITY

mb_substitute_character

mb_check_encoding

mb_convert_encoding

□□□ #2
□ □□□ □□□ 2026-10-31 17:53:47 UTC
□ □□□ □□□ 2026-10-31 17:53:49 UTC