xdiff

Introduction

xdiff extension creates and applies patches to both text and binary files.

Installing/Configuring

Requirements

To use xdiff, you will need libxdiff installed, available on the libxdiff homepage » http://www.xmailserver.org/xdiff-lib.html.

Note
You'll need at least libxdiff 0.7 for these functions to be aware of memory_limit.

Installation

xdiff is currently available through PECL » http://pecl.php.net/package/xdiff.

If » PEAR is available on your *nix-like system you can use the pear installer to install the xdiff extension, by the following command: *pear -v install xdiff*.

You can always download the tar.gz package and install xdiff by hand:

Example #1 - xdiff install by hand

```
gunzip xdiff-xxx.tgz
tar -xvf xdiff-xxx.tar
cd xdiff-xxx
phpize
./configure && make && make install
```

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.

XDIFF_PATCH_NORMAL (integer)

XDIFF_PATCH_REVERSE (integer)

xdiff Functions

xdiff_file_diff_binary

xdiff_file_diff_binary -- Make binary diff of two files

Description

bool **xdiff_file_diff_binary** (string `$file1`, string `$file2`, string `$dest`)

Makes a binary diff of two files and stores the result in a file. This function works with both text and binary files.

Parameters

`file1`

`file2`

`dest`
Path of the resulting file. Resulting file is in binary format.

Return Values

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

Examples

Example #2 - xdiff_file_diff_binary() **example**

The following code makes binary diff of two archives.

```
<?php
$old_version = 'my_script_1.0.tgz';
$new_version = 'my_script_1.1.tgz';

xdiff_file_diff_binary($old_version, $new_version, 'my_script.bdiff');
?>
```

Notes

Note
Both files will be loaded into memory so ensure that your memory_limit is set high enough.

See Also

- xdiff_string_diff_binary()

xdiff_file_diff

xdiff_file_diff -- Make unified diff of two files

Description

bool **xdiff_file_diff** (string `$file1`, string `$file2`, string `$dest` [, int `$context` [, bool `$minimal`]])

Makes a diff of two files and stores the result in a file.

Parameters

`file1`

`file2`

`dest`
Path of the resulting file.

`context`
Indicates how many lines of context you want to include in diff result.

`minimal`
Set this parameter to **TRUE** if you want to minimalize size of diff (can take a long time). Resulting file is human-readable.

Return Values

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

Examples

Example #3 - xdiff_file_diff() **example**

The following code makes unified diff of two php files.

```
<?php
$old_version = 'my_script.php';
$new_version = 'my_new_script.php';

xdiff_file_diff($old_version, $new_version, 'my_script.diff', 2);
?>
```

Notes

Note
This function doesn't work well with binary files. To make diff of binary files use xdiff_file_diff_binary().

See Also

- xdiff_string_diff()

xdiff_file_merge3

xdiff_file_merge3 -- Merge 3 files into one

Description

mixed **xdiff_file_merge3** (string `$file1`, string `$file2`, string `$file3`, string `$dest`)

Merges three files into one and stores the result in a file.

Parameters

`file1`

`file2`

`file3`

`dest`
Path of the resulting file.

Return Values

Returns **TRUE** if merge was successful, string with rejected chunks if it was not or **FALSE** if an internal error happened.

Examples

Example #4 - xdiff_file_merge3() example

The following code merges three files into one.

```
<?php
$old_version = 'original_script.php';
$fix1 = 'script_with_fix1.php';
$fix2 = 'script_with_fix2.php';

$errors = xdiff_file_merge3($old_version, $fix1, $fix2, 'fixed_script.php');
if (is_string($errors)) {
   echo "Rejects:\n";
   echo $errors;
}
?>
```

See Also

- xdiff_string_merge3()

xdiff_file_patch_binary

xdiff_file_patch_binary -- Patch a file with a binary diff

Description

bool **xdiff_file_patch_binary** (string `$file`, string `$patch`, string `$dest`)

Patches a *`file`* with a binary *`patch`* and stores the result in a file.

Parameters

`file`
: The original file.

`patch`
: The binary patch file.

`dest`
: Path of the resulting file.

Return Values

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

Examples

Example #5 - xdiff_file_patch_binary() **example**

The following code applies binary diff to a file.

```
<?php
$old_version = 'archive-1.0.tgz';
$patch = 'archive.bpatch';

$result = xdiff_file_patch_binary($old_version, $patch, 'archive-1.1.tgz');
if ($result) {
  echo "File patched";
} else {
  echo "File couldn't be patched";
}

?>
```

Notes

Note
Both files (*file* and *patch*) will be loaded into memory so ensure that your memory_limit is set high enough.

See Also

- xdiff_string_patch_binary()

xdiff_file_patch

xdiff_file_patch -- Patch a file with an unified diff

Description

mixed **xdiff_file_patch** (string `$file`, string `$patch`, string `$dest` [, int `$flags`])

Patches a *`file`* with a unified *`patch`* and stores the result in a file.

Parameters

`file`
: The original file.

`patch`
: The patch file.

`dest`
: Path of the resulting file.

`flags`
: Can be either **XDIFF_PATCH_NORMAL** (default mode, normal patch) or **XDIFF_PATCH_REVERSE** (reversed patch).

Return Values

Returns **FALSE** if an internal error happened, string with rejected chunks of patch or **TRUE** if patch has been successfully applied.

Examples

Example #6 - xdiff_file_patch() example

The following code applies unified diff to a file.

```
<?php
$old_version = 'my_script-1.0.php';
$patch = 'my_script.patch';

$errors = xdiff_file_patch($old_version, $patch, 'my_script-1.1.php');
if (is_string($errors)) {
  echo "Rejects:\n";
  echo $errors;
}
```

```
?>
```

Example #7 - Patch reversing example

The following code reverses a patch.

```
<?php
$new_version = 'my_script-1.1.php';
$patch = 'my_script.patch';

$errors = xdiff_file_patch($new_version, $patch, 'my_script-1.0.php',
XDIFF_PATCH_REVERSE);
if (is_string($errors)) {
  echo "Rejects:\n";
  echo $errors;
}

?>
```

See Also

- xdiff_string_patch()

xdiff_string_diff_binary

xdiff_string_diff_binary -- Make binary diff of two strings

Description

string **xdiff_string_diff_binary** (string `$str1`, string `$str2`)

Makes a binary diff of two strings.

Parameters

`str1`

`str2`

Return Values

Returns string with result or **FALSE** if an internal error happened.

See Also

- xdiff_file_diff_binary()

xdiff_string_diff

xdiff_string_diff -- Make unified diff of two strings

Description

string **xdiff_string_diff** (string `$str1`, string `$str2` [, int `$context` [, bool `$minimal`]])

Makes a unified diff of two strings.

Parameters

`str1`

`str2`

`context`
Indicates how many lines of context you want to include in the diff result.

`minimal`
Set this parameter to **TRUE** if you want to minimalize the size of the diff (can take a long time).

Return Values

Returns string with result or **FALSE** if an internal error happened.

Examples

Example #8 - xdiff_string_diff() **example**

The following code makes unified diff of two articles.

```
<?php
$old_article = file_get_contents('./old_article.txt');
$new_article = $_REQUEST['article']; /* Let's say that someone pasted a new
article to html form */

$diff = xdiff_string_diff($old_article, $new_article, 1);
if (is_string($diff)) {
   echo "Differences between two articles:\n";
   echo $diff;
}
```

```
?>
```

Notes

Note
This function doesn't work well with binary strings. To make diff of binary strings use xdiff_string_diff_binary().

See Also

- xdiff_file_diff()

xdiff_string_merge3

xdiff_string_merge3 -- Merge 3 strings into one

Description

mixed **xdiff_string_merge3** (string `$str1`, string `$str2`, string `$str3` [, string `&$error`])

Merges three strings into one.

Parameters

`str1`

`str2`

`str3`

`error`
If provided then rejected parts are stored inside this variable.

Return Values

Returns merged string, **FALSE** if an internal error happened, or **TRUE** if merged string is empty.

See Also

- xdiff_file_merge3()

xdiff_string_patch_binary

xdiff_string_patch_binary -- Patch a string with a binary diff

Description

string **xdiff_string_patch_binary** (string `$str`, string `$patch`)

Patches a string with a binary *`patch`*.

Parameters

`str`
: The original binary string.

`patch`
: The binary patch string.

Return Values

Returns the patched string, or **FALSE** on error.

See Also

- xdiff_file_patch_binary()

xdiff_string_patch

xdiff_string_patch -- Patch a string with an unified diff

Description

string **xdiff_string_patch** (string `$str`, string `$patch` [, int `$flags` [, string `&$error`]])

Patches a string with a unified *`patch`* string.

Parameters

`str`
The original string.

`patch`
The unified patch string.

`flags`
`flags` can be either **XDIFF_PATCH_NORMAL** (default mode, normal patch) or **XDIFF_PATCH_REVERSE** (reversed patch).

`error`
If provided then rejected parts are stored inside this variable.

Return Values

Returns the patched string, or **FALSE** on error.

Examples

Example #9 - xdiff_string_patch() **example**

The following code applies changes to some article.

```
<?php
$old_article = file_get_contents('./old_article.txt');
$diff = $_SERVER['patch']; /* Let's say that someone pasted a patch to html
form */

$errors = '';

$new_article = xdiff_string_patch($old_article, $diff, XDIFF_PATCH_NORMAL,
$errors);
if (is_string($new_article)) {
   echo "New article:\n";
```

```
   echo $new_article;
}

if (strlen($errors)) {
   echo "Rejects: \n";
   echo $errors;
}

?>
```

See Also

- xdiff_file_patch()