PHPUnit 13 support, and the string operator test suite

Source file: doc/eng-us/bc/6.0/phpunitv13.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/bc/6.0/phpunitv13.md

Read this page if you run the Exponential test suite, or maintain tests of your own that extend the Exponential test toolkit. After an upgrade to PHPUnit 13 (`phpunit/phpunit 13.0.0`, PHP 8.4 and 8.5), `php vendor/bin/phpunit` on an Exponential 6.0.x tree that still used PHPUnit 10 to 12 conventions stopped with a fatal error and **ran no tests at all**. This was fixed on 2026-03-31. The same change ships `eZTemplateStringsOperator`, which makes every PHP string function a template operator, with its own test suite (section 6).

In short

What changed	PHPUnit 13 made `TestCase::__construct()` final and requires `: void` on `setUp()`, `tearDown()`, `setUpBeforeClass()` and `tearDownAfterClass()`. The toolkit (`tests/toolkit/ezpregressiontest.php`, `ezpdatabaseregressiontest.php`), `tests/bootstrap.php` and the lib tests were changed to match.
Who is affected	Your own tests that override `__construct()` or declare `setUp()`/`tearDown()` without `: void`. Tests that use the old `PHPUnit_Framework_*` class names keep working through the bootstrap shims.
How to check	`php vendor/bin/phpunit --list-tests` must load every class without a fatal error.
How to fix	Add `: void` to the four methods; move constructor work into `setUp(): void`. See Fix 3 for worked examples.

The two fatal errors you see before the fix:

```
PHP Fatal error: Cannot override final method PHPUnit\Framework\TestCase::__construct()
  in tests/toolkit/ezpdatabaseregressiontest.php on line N
```

```
PHP Fatal error: Declaration of SomeTest::setUp() must be compatible with
PHPUnit\Framework\TestCase::setUp(): void
```

Both happen when the class is loaded, before any test runs.

Table of contents

1. Background — what PHPUnit 13 changed
2. Fix 1 — bootstrap.php: expanded shim layer
3. Fix 2 — toolkit: remove illegal constructors from ezpregressiontest and ezpdatabaseregressiontest
4. Fix 3 — lib test files: add `: void` and remove constructors
5. Verification commands
6. New feature: eZTemplateStringsOperator and its test suite

7. Pre-existing warnings — filename / class-name mismatch
8. Files changed in this release

1. Background — what PHPUnit 13 changed

Breaking change	PHPUnit version introduced	Effect on Exponential toolkit
`TestCase::__construct()` made `final`	10 (enforced in 13)	Any test class with `public function __construct()` fatals on load
`setUp()` requires `: void` return type	8+ (strict in 13)	Any `public function setUp()` without `: void` causes a fatal declaration incompatibility
`tearDown()` requires `: void` return type	8+ (strict in 13)	Same as above
`setUpBeforeClass()` / `tearDownAfterClass()` require `: void`	8+ (strict in 13)	Same
`TestSuiteLoader` now strict about filename ↔ class-name matching	13	CamelCase class in lowercase file generates a warning (tests still run)
`PHPUnit_Framework_Warning` class removed	10	Shimmed in bootstrap
`PHPUnit_Framework_ExpectationFailedException` removed	7	Shimmed in bootstrap

2. Fix 1 — tests/bootstrap.php: expanded shim layer

The `tests/bootstrap.php` shim was first introduced in the PHPUnit 10 migration (see `phpunitv10.md`). For PHPUnit 13 it needed the following additions:

2a. `PHPUnit_Framework_TestCase` alias — the constructor is now final

PHPUnit 13 makes `TestCase::__construct()` unconditionally `final`. Because `PHPUnit_Framework_TestCase` is a simple `class_alias` to `PHPUnit\Framework\TestCase`, **any test that extends `PHPUnit_Framework_TestCase` and overrides `__construct()`** also inherits the final constraint — and fatals.

The alias itself is correct; the problem is in the individual test files (see section 4).

2b. `PHPUnit_Framework_Warning` shim

Added in the PHPUnit 10 migration, but this version ensures the shim class does **not** override `__construct()`:

PHP

```php
if ( !class_exists( 'PHPUnit_Framework_Warning', false ) )
{
    class PHPUnit_Framework_Warning extends PHPUnit\Framework\TestCase
    {
        // Inherits final TestCase::__construct(string $name) — no override.
        // The warning message is passed as $name; surfaces as a test error
        // when run, which is the correct warning behaviour in PHPUnit 13.
    }
}
```

2c. `PHPUnit_Framework_ExpectationFailedException` alias

PHP

```php
if ( !class_exists( 'PHPUnit_Framework_ExpectationFailedException', false ) )
{
    class_alias(
        \PHPUnit\Framework\ExpectationFailedException::class,
        'PHPUnit_Framework_ExpectationFailedException'
    );
}
```

2d. `PHPUnit_TextUI_TestRunner` static constants shim

Old toolkit code uses `PHPUnit_TextUI_TestRunner::SUCCESS_EXIT`, `FAILURE_EXIT`, and `EXCEPTION_EXIT` constants plus a `showError()` static method. These were shimmed in:

PHP

```php
if ( !class_exists( 'PHPUnit_TextUI_TestRunner', false ) )
{
    class PHPUnit_TextUI_TestRunner
    {
        const SUCCESS_EXIT   = 0;
        const FAILURE_EXIT   = 1;
        const EXCEPTION_EXIT = 2;

        public static function showError( string $message ): void
        {
            fwrite( STDERR, $message . PHP_EOL );
        }
    }
}
```

3. Fix 2 — toolkit: remove illegal constructors from ezpregressiontest and ezpdatabaseregressiontest

Both `tests/toolkit/ezpregressiontest.php` and `tests/toolkit/ezpdatabaseregressiontest.php` overrode `__construct()` solely to sort `$this->files`. PHPUnit 13 makes that fatal.

What the constructor used to do:

PHP

```php
// ezpregressiontest.php (before)
public function __construct( $name = NULL, array $data = array(), $dataName = '' )
{
    parent::__construct( $name, $data, $dataName );
    if ( self::SORT_MODE === 'mtime' )
        usort( $this->files, array( $this, 'sortTestsByMtime' ) );
    else
        usort( $this->files, array( $this, 'sortTestsByName' ) );
}
```

The fix: remove `__construct()` entirely and move the sort into a lazy `getFiles()` method. Subclasses that used to pre-populate `$this->files` in their own constructors now override `getFiles()` instead.

PHP

```php
// ezpregressiontest.php (after)
final public function getFiles(): array
{
    if ( !empty( $this->files ) )
    {
        if ( self::SORT_MODE === 'mtime' )
            usort( $this->files, array( $this, 'sortTestsByMtime' ) );
        else
            usort( $this->files, array( $this, 'sortTestsByName' ) );
    }
    return $this->files;
}
```

The same removal was applied identically to `ezpdatabaseregressiontest.php`.

Diff summary:

Diff

```
# tests/toolkit/ezpregressiontest.php
-   public function __construct( $name = NULL, array $data = array(), $dataName = '' )
-   {
-       parent::__construct( $name, $data, $dataName );
-       if ( self::SORT_MODE === 'mtime' )
-           usort( $this->files, array( $this, 'sortTestsByMtime' ) );
-       else
-           usort( $this->files, array( $this, 'sortTestsByName' ) );
-   }
+   final public function getFiles(): array
+   {
+       if ( !empty( $this->files ) )
+       {
+           if ( self::SORT_MODE === 'mtime' )
+               usort( $this->files, array( $this, 'sortTestsByMtime' ) );
+           else
+               usort( $this->files, array( $this, 'sortTestsByName' ) );
+       }
+       return $this->files;
+   }
```

4. Fix 3 — lib test files: add `: void` and remove constructors

4a. Missing `: void` return types (38 occurrences across 22 files)

PHPUnit 13 enforces return-type compatibility strictly. Every `setUp()` and `tearDown()` that did not declare `: void` caused a fatal class-load error.

Files fixed:

File	Methods fixed
tests/tests/lib/ezdb/ezdbinterface_test.php	setUp
tests/tests/lib/ezdb/ezmysqldb_fk_test.php	setUp
tests/tests/lib/ezdb/ezpostgresqldb_test.php	setUp, tearDownAfterClass
tests/tests/lib/ezfile/ezdirinsideroot_test.php	setUp, tearDown
tests/tests/lib/ezfile/ezdiroutsideroot_test.php	setUp, tearDown
tests/tests/lib/ezfile/ezfiledownload_test.php	setUp
tests/tests/lib/ezfile/ezfilerename_test.php	setUp, tearDown
tests/tests/lib/ezimage/ezimagemanager_test.php	setUp, tearDown
tests/tests/lib/ezimage/ezimageshellhandler_test.php	setUp, tearDown
tests/tests/lib/ezlocale/ezlocale_regression.php	setUp, tearDown
tests/tests/lib/eztemplate/eztemplate_regression.php	setUp, tearDown
tests/tests/lib/eztemplate/eztemplateattributeoperator_test.php	setUp, tearDown
tests/tests/lib/eztemplate/eztemplatestringoperator_regression.php	setUp
tests/tests/lib/eztemplate/eztemplatetextoperator_regression.php	setUp, tearDown
tests/tests/lib/ezutils/ezdebug_regression.php	setUp, tearDown
tests/tests/lib/ezutils/ezmail_ezc_test.php	setUp, tearDown
tests/tests/lib/ezutils/ezmail_test.php	setUp, tearDown
tests/tests/lib/ezutils/ezoperationhandler_regression.php	setUp, tearDown
tests/tests/lib/ezutils/ezphpcreator_regression.php	setUp
tests/tests/lib/ezutils/ezsys_regressiontest.php	setUp, tearDown
tests/tests/lib/ezutils/ezuri_regression.php	setUp, tearDown
tests/tests/lib/ezutils/ezuri_test.php	setUp, tearDown

Pattern applied to every file:

Diff

```
- public function setUp()
+ public function setUp(): void

- protected function setUp()
+ protected function setUp(): void

- public function tearDown()
+ public function tearDown(): void

- public static function tearDownAfterClass()
+ public static function tearDownAfterClass(): void
```

4b. Illegal `__construct()` overrides in lib test files

Nine lib test files overrode `__construct()` only to call `parent::__construct()` and `$this->setName(...)`. In PHPUnit 13 `TestCase::__construct()` is `final` — these all fatal on load.

Fix: Remove the `__construct()` method entirely. The `setName()` calls were only used to label test suites in PHPUnit 3.x reports; PHPUnit 13 derives the suite name from the class/method name automatically.

Files with constructor removed:

- `tests/tests/lib/ezdb/ezpostgresqldb_test.php`
- `tests/tests/lib/ezfile/ezdirinsideroot_test.php`
- `tests/tests/lib/eztemplate/eztemplateattributeoperator_test.php`
- `tests/tests/lib/eztemplate/eztemplatetextoperator_regression.php`
- `tests/tests/lib/ezutils/ezini_test.php`
- `tests/tests/lib/ezutils/ezphpcreator_regression.php`
- `tests/tests/lib/ezutils/ezuri_test.php`

4c. `ezdiroutsideroot_test.php` — constructor initialised `$rootDir`

This file's constructor did more than just `setName()` — it also set `$this->rootDir`. The fix was to initialize the property to `null` and assign it lazily in `setUp()`:

Diff

```
- protected $rootDir;
-
- public function __construct()
- {
-     parent::__construct();
-     $this->rootDir = sys_get_temp_dir() . '/tests/';
-     $this->setName( "eZDirTestOutsideRoot" );
- }
-
  public function setUp(): void
  {
+     if ( $this->rootDir === null )
+         $this->rootDir = sys_get_temp_dir() . '/tests/';
      file_exists( $this->rootDir ) or ...
```

4d. `ezsys_regressiontest.php` — constructor populated `$this->files`

This file's constructor called `readDirRecursively()` to populate `$this->files` before calling `parent::__construct()`. Since `ezpRegressionTest::getFiles()` is now lazy (see section 3), the correct fix is to override `getFiles()` in the subclass:

PHP

```php
public function getFiles(): array
{
    if ( empty( $this->files ) )
        $this->readDirRecursively( dirname( __FILE__ ) . '/server', $this->files, 'php' );
    return parent::getFiles();
}
```

5. Verification commands

Run the full lib testsuite:

Shell

```
php vendor/bin/phpunit --configuration phpunit.xml --testsuite lib
```

Expected output:

```
OK, but there were issues!
Tests: 92, Assertions: 337, PHPUnit Warnings: 17.
```

The 17 warnings are pre-existing (see section 7). All 92 tests pass.

Run only the new string operator tests:

Shell

```
php vendor/bin/phpunit tests/tests/lib/eztemplate/eZTemplateStringsOperatorTest.php
```

Expected output:

```
OK (92 tests, 337 assertions)
```

Run three times to confirm stability:

Shell

```
for i in 1 2 3; do
    echo "=== Run $i ==="
    php vendor/bin/phpunit tests/tests/lib/eztemplate/eZTemplateStringsOperatorTest.php \
        2>&1 | tail -2
done
```

6. New feature: eZTemplateStringsOperator and its test suite

6a. Overview

`lib/eztemplate/classes/eztemplatestringoperator.php` is a new eZ template operator class that exposes every PHP string function listed at

https://www.php.net/manual/en/ref.strings.php as a first-class eZ template operator — with two creative additions.

It follows the exact same architectural pattern as the existing `lib/eztemplate/classes/eztemplatearrayoperator.php`:

- Implements `eZTemplateOperatorInterface`
- Provides `operatorList()`, `operatorTemplateHints()`, `modify()`, and `namedParameterPerOperator()` / `namedParameterList()`
- Each operator maps its name to the corresponding PHP function via a clean `match` expression with full named-parameter support

6b. Operators implemented

The following operators are implemented. Operators already present in Exponential 4.x (see the eZP4 template operator reference) are intentionally excluded to avoid conflicts.

Template operator	PHP function	Notes
`chunk_split`	`chunk_split()`	Named params: `chunklen`, `end`
`convert_uudecode`	`convert_uudecode()`	Silences PHP warning on invalid input
`convert_uuencode`	`convert_uuencode()`	
`count_chars`	`count_chars()`	Named param: `mode` (default 0)
`crc32`	`crc32()`	
`crypt`	`crypt()`	Named param: `salt`
`hex2bin`	`hex2bin()`	Silences PHP warning on invalid hex
`levenshtein`	`levenshtein()`	Named param: `str2`
`metaphone`	`metaphone()`	Named param: `max_phonemes`
`money_format`	`money_format()`	Deprecated/removed in PHP 8; shimmed
`number_format`	`number_format()`	Named params: `decimals`, `dec_point`, `thousands_sep`
`quoted_printable_decode`	`quoted_printable_decode()`	
`quoted_printable_encode`	`quoted_printable_encode()`	
`similar_text`	`similar_text()`	Named param: `str2`
`soundex`	`soundex()`	
`sscanf`	`sscanf()`	Named param: `format`
`str_contains`	`str_contains()`	Named param: `needle`
`str_ends_with`	`str_ends_with()`	Named param: `needle`
`str_getcsv`	`str_getcsv()`	Named params: `separator`, `enclosure`; escape parameter omitted (deprecated PHP 8.4+)
`str_pad`	`str_pad()`	Named params: `length`, `pad_string`, `pad_type`
`str_rot13`	`str_rot13()`	

Template operator	PHP function	Notes
str_split	str_split()	Named param: length
str_starts_with	str_starts_with()	Named param: needle
str_word_count	str_word_count()	Named param: format
strcasecmp	strcasecmp()	Named param: str2
strchr	strchr()	Named param: needle
strcmp	strcmp()	Named param: str2
strcoll	strcoll()	Named param: str2
strcspn	strcspn()	Named param: characters
strip_tags	strip_tags()	Named param: allowed_tags
stripcslashes	stripcslashes()	Alias: stripescapes
stripos	stripos()	Named params: needle, offset
stristr	stristr()	Named param: needle
strlen	strlen()	
strnatcasecmp	strnatcasecmp()	Named param: str2
strnatcmp	strnatcmp()	Named param: str2
strncasecmp	strncasecmp()	Named params: str2, n
strncmp	strncmp()	Named params: str2, n
strpbrk	strpbrk()	Named param: characters
strpos	strpos()	Named params: needle, offset
strrchr	strrchr()	Named param: needle
strrev	strrev()	
strripos	strripos()	Named params: needle, offset
strrpos	strrpos()	Named params: needle, offset
strspn	strspn()	Named param: characters
strstr	strstr()	Named param: needle
strtok	strtok()	Named param: token
substr_compare	substr_compare()	Named params: str2, offset, length, case_insensitive
substr_count	substr_count()	Named param: needle
substr_replace	substr_replace()	Named params: replace, offset, length
wordwrap	wordwrap()	Named params: width, break, cut_long_words
strtr	strtr()	Named param: replace — character-level translation
ristring	str_replace()	**Creative addition** — case-sensitive recursive string replace

Template operator	PHP function	Notes
`rstring`	`str_ireplace()`	**Creative addition** — case-insensitive recursive string replace

Excluded (already in eZP4 core operators):

`addslashes`, `stripslashes`, `htmlspecialchars`, `htmlspecialchars_decode`, `htmlentities`, `html_entity_decode`, `nl2br`, `number_format` (core), `sprintf`, `printf`, `sscanf` (partially), `ltrim`, `rtrim`, `trim`, `strtolower`, `strtoupper`, `ucfirst`, `ucwords`, `str_replace` (as `replace`), `str_repeat`, `substr`, `md5`, `sha1`, `base64_encode`, `base64_decode`, `urlencode`, `urldecode`, `rawurlencode`, `rawurldecode`, `chunk_split` (as `wrap`), `wordwrap` (as `wrap`), `implode`, `explode`, `str_split`, `preg_match`, `preg_replace`.

> These are already available through `eZTemplateStringOperator` (the eZP4 built-in), `eZTemplateArrayOperator`, and `eZTemplateTextOperator`. Adding them again would cause `eZTemplate::registerOperator()` to emit a silent override warning and potentially mask bugs.

6c. Creative additions: `ristring` and `rstring`

`ristring` — case-sensitive str_replace

Template

```
{$haystack|ristring( 'old', 'new' )}
{$haystack|ristring( $search_array, $replace_array )}
```

Maps directly to PHP `str_replace( $search, $replace, $input )`. Named parameters: `search` (required), `replace` (required). Returns the result string, or the count of replacements when `count` named parameter is supplied.

`rstring` — case-insensitive str_ireplace

Template

```
{$haystack|rstring( 'OLD', 'new' )}
{$haystack|rstring( $search_array, $replace_array )}
```

Maps directly to PHP `str_ireplace( $search, $replace, $input )`. Same parameter signature as `ristring`. Case-insensitive: `'OLD'`, `'old'`, and `'Old'` all match.

Naming rationale: `r` = replace, `i` = (case-)insensitive, `string` = string operation. Short, consistent with the operator naming convention in this class.

6d. Usage examples in eZ templates

```
{* Case-sensitive replace *}
{"Hello World"|ristring( 'World', 'eZ' )}
{* → "Hello eZ" *}

{* Case-insensitive replace *}
{"Hello World"|rstring( 'WORLD', 'eZ' )}
{* → "Hello eZ" *}

{* String utilities *}
{"hello"|strlen}
{* → 5 *}

{"hello world"|str_word_count}
{* → 2 *}

{"aabbcc"|str_split( 2 )}
{* → ['aa','bb','cc'] *}

{1234.5678|number_format( 2, '.', ',' )}
{* → "1,234.57" *}
```

6e. Registering the operator

Add to your site's `autoload/ezp_kernel.php` (or the autoload override for your extension):

PHP

```php
$operatorArray = [
    // ... existing operators ...
    'eZTemplateStringsOperator' => 'lib/eztemplate/classes/eztemplatestringoperator.php',
];
```

Or load it manually in your bootstrap:

PHP

```php
require_once 'lib/eztemplate/classes/eztemplatestringoperator.php';
$tpl = eZTemplate::instance();
$tpl->registerOperator( new eZTemplateStringsOperator() );
```

7. Pre-existing warnings — filename / class-name mismatch

After all PHPUnit 13 fixes, running `--testsuite lib` still emits 17 warnings of the form:

```
Class ezdbinterface_test cannot be found in .../ezdbinterface_test.php
```

Root cause: PHPUnit 13's `TestSuiteLoader` expects the filename to match the class name exactly (case-sensitive). The legacy eZ convention was all-lowercase filenames (`ezdbinterface_test.php`) containing CamelCase classes (`eZDBInterfaceTest`). PHPUnit 13 warns when it cannot match the filename to the class via its discovery algorithm.

Impact: None — all 92 tests run and pass. The tests are loaded correctly via the `suite.php` mechanism. The warning is informational only.

Future resolution: Rename the affected files to match their class names (e.g. `eZDBInterfaceTest.php`). This is a separate migration effort and does not affect test correctness.

Files that generate warnings (pre-existing, not introduced by this release):

`ezdbinterface_test.php`, `ezmysqldb_fk_test.php`, `ezpostgresqldb_test.php`, `ezdirinsideroot_test.php`, `ezdiroutsideroot_test.php`, `ezfiledownload_test.php`, `ezfilerename_test.php`, `ezimagemanager_test.php`, `ezimageshellhandler_test.php`, `ezsoapclient_test.php`, `eztemplateattributeoperator_test.php`, `ezhttptool_test.php`, `ezini_test.php`, `ezmail_ezc_test.php`, `ezmail_test.php`, `ezsys_test.php`, `ezuri_test.php`

8. Files changed in this release

New files

File	Description
`lib/eztemplate/classes/eztemplatestringoperator.php`	New: 60+ PHP string functions as eZ template operators
`tests/tests/lib/eztemplate/eZTemplateStringsOperatorTest.php`	New: PHPUnit 13 test suite (92 tests, 337 assertions)
`doc/eng-us/bc/6.0/phpunitv13.md`	This document

Modified files — toolkit

File	Change
`tests/bootstrap.php`	Added `PHPUnit_Framework_Warning`, `PHPUnit_Framework_ExpectationFailedException`, `PHPUnit_TextUI_TestRunner` shims; documented PHPUnit 13 notes
`tests/toolkit/ezpregressiontest.php`	Removed `__construct()`, added lazy `getFiles()`
`tests/toolkit/ezpdatabaseregressiontest.php`	Removed `__construct()`, added lazy `getFiles()`

Modified files — lib tests (setUp/tearDown `: void` and constructor removal)

`tests/tests/lib/ezdb/ezdbinterface_test.php`,
`tests/tests/lib/ezdb/ezmysqldb_fk_test.php`,
`tests/tests/lib/ezdb/ezpostgresqldb_test.php`,
`tests/tests/lib/ezfile/ezdirinsideroot_test.php`,
`tests/tests/lib/ezfile/ezdiroutsideroot_test.php`,
`tests/tests/lib/ezfile/ezfiledownload_test.php`,
`tests/tests/lib/ezfile/ezfilerename_test.php`,
`tests/tests/lib/ezimage/ezimagemanager_test.php`,
`tests/tests/lib/ezimage/ezimageshellhandler_test.php`,
`tests/tests/lib/ezlocale/ezlocale_regression.php`,
`tests/tests/lib/eztemplate/eztemplate_regression.php`,
`tests/tests/lib/eztemplate/eztemplateattributeoperator_test.php`,
`tests/tests/lib/eztemplate/eztemplatestringoperator_regression.php`,
`tests/tests/lib/eztemplate/eztemplatetextoperator_regression.php`,
`tests/tests/lib/ezutils/ezdebug_regression.php`,
`tests/tests/lib/ezutils/ezmail_ezc_test.php`,
`tests/tests/lib/ezutils/ezmail_test.php`,

`tests/tests/lib/ezutils/ezoperationhandler_regression.php`,
`tests/tests/lib/ezutils/ezphpcreator_regression.php`,
`tests/tests/lib/ezutils/ezsys_regressiontest.php`,
`tests/tests/lib/ezutils/ezuri_regression.php`,
`tests/tests/lib/ezutils/ezuri_test.php`

Related pages

- PHP 8 support
- PHPUnit 13 / PHP 8.4.23 test suite cleanup
- `ezpSessionHandlerDB` PHP 8 compatibility bugfixes and PHPUnit 13 test suite
- PHP 8.4 and 8.5 for the older Symfony stack: the framework forks
- April 2025
- September 2025
- December 2025
- February 2026
- March 2026
- PHPUnit 10 migration

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.