<https://github.com/PHPMailer/PHPMailer>

**PHPMailer - A full-featured email creation and transfer class for PHP**

Build status:

**Class Features**

* Probably the world's most popular code for sending email from PHP!
* Used by many open-source projects: WordPress, Drupal, 1CRM, SugarCRM, Yii, Joomla! and many more
* Integrated SMTP support - send without a local mail server
* Send emails with multiple To, CC, BCC and Reply-to addresses
* Multipart/alternative emails for mail clients that do not read HTML email
* Add attachments, including inline
* Support for UTF-8 content and 8bit, base64, binary, and quoted-printable encodings
* SMTP authentication with LOGIN, PLAIN, CRAM-MD5 and XOAUTH2 mechanisms over SSL and SMTP+STARTTLS transports
* Validates email addresses automatically
* Protect against header injection attacks
* Error messages in over 50 languages!
* DKIM and S/MIME signing support
* Compatible with PHP 5.5 and later
* Namespaced to prevent name clashes
* Much more!

**Why you might need it**

Many PHP developers utilize email in their code. The only PHP function that supports this is the mail() function. However, it does not provide any assistance for making use of popular features such as HTML-based emails and attachments.

Formatting email correctly is surprisingly difficult. There are myriad overlapping RFCs, requiring tight adherence to horribly complicated formatting and encoding rules - the vast majority of code that you'll find online that uses the mail() function directly is just plain wrong! *Please* don't be tempted to do it yourself - if you don't use PHPMailer, there are many other excellent libraries that you should look at before rolling your own - try [SwiftMailer](https://swiftmailer.symfony.com/), [Zend/Mail](https://zendframework.github.io/zend-mail/), [eZcomponents](https://github.com/zetacomponents/Mail) etc.

The PHP mail() function usually sends via a local mail server, typically fronted by a sendmail binary on Linux, BSD and OS X platforms, however, Windows usually doesn't include a local mail server; PHPMailer's integrated SMTP implementation allows email sending on Windows platforms without a local mail server.

**License**

This software is distributed under the [LGPL 2.1](http://www.gnu.org/licenses/lgpl-2.1.html) license. Please read LICENSE for information on the software availability and distribution.

**Installation & loading**

PHPMailer is available on [Packagist](https://packagist.org/packages/phpmailer/phpmailer) (using semantic versioning), and installation via [Composer](https://getcomposer.org/) is the recommended way to install PHPMailer. Just add this line to your composer.json file:

"phpmailer/phpmailer": "~6.0"

or run

composer require phpmailer/phpmailer

Note that the vendor folder and the vendor/autoload.php script are generated by Composer; they are not part of PHPMailer.

If you want to use the Gmail XOAUTH2 authentication class, you will also need to add a dependency on the league/oauth2-client package in your composer.json.

Alternatively, if you're not using Composer, copy the contents of the PHPMailer folder into one of the include\_path directories specified in your PHP configuration and load each class file manually:

<?php

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

require 'path/to/PHPMailer/src/Exception.php';

require 'path/to/PHPMailer/src/PHPMailer.php';

require 'path/to/PHPMailer/src/SMTP.php';

If you're not using the SMTP class explicitly (you're probably not), you don't need a use line for the SMTP class.

If you don't speak git or just want a tarball, click the 'zip' button on the right of the project page in GitHub, though note that docs and examples are not included in the tarball.

**Legacy versions**

PHPMailer 5.2 (which is compatible with PHP 5.0 - 7.0) is no longer being supported for feature updates, and will only be receiving security updates from now on. You will find the latest version of 5.2 in the [5.2-stable branch](https://github.com/PHPMailer/PHPMailer/tree/5.2-stable), and future versions of 5.2 will be tagged with 5.2.x version numbers, so existing Composer configs should remain working. If you're using PHP 5.5 or later, we recommend you make the necessary changes to switch to the 6.0 release.

The 5.2 branch will not receive security updates after December 31st 2018.

**Upgrading from 5.2**

The biggest changes are that source files are now in the src/ folder, and PHPMailer now declares the namespace PHPMailer\PHPMailer. This has several important effects – [read the upgrade guide](https://github.com/PHPMailer/PHPMailer/tree/master/UPGRADING.md) for more details.

**Minimal installation**

While installing the entire package manually or with Composer is simple, convenient and reliable, you may want to include only vital files in your project. At the very least you will need [src/PHPMailer.php](https://github.com/PHPMailer/PHPMailer/tree/master/src/PHPMailer.php). If you're using SMTP, you'll need [src/SMTP.php](https://github.com/PHPMailer/PHPMailer/tree/master/src/SMTP.php), and if you're using POP-before SMTP, you'll need [src/POP3.php](https://github.com/PHPMailer/PHPMailer/tree/master/src/POP3.php). You can skip the [language](https://github.com/PHPMailer/PHPMailer/tree/master/language/) folder if you're not showing errors to users and can make do with English-only errors. If you're using XOAUTH2 you will need [src/OAuth.php](https://github.com/PHPMailer/PHPMailer/tree/master/src/OAuth.php) as well as the Composer dependencies for the services you wish to authenticate with. Really, it's much easier to use Composer!

**A Simple Example**

<?php

// Import PHPMailer classes into the global namespace

// These must be at the top of your script, not inside a function

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

//Load Composer's autoloader

require 'vendor/autoload.php';

$mail = new PHPMailer(true); // Passing `true` enables exceptions

try {

//Server settings

$mail->SMTPDebug = 2; // Enable verbose debug output

$mail->isSMTP(); // Set mailer to use SMTP

$mail->Host = 'smtp1.example.com;smtp2.example.com'; // Specify main and backup SMTP servers

$mail->SMTPAuth = true; // Enable SMTP authentication

$mail->Username = 'user@example.com'; // SMTP username

$mail->Password = 'secret'; // SMTP password

$mail->SMTPSecure = 'tls'; // Enable TLS encryption, `ssl` also accepted

$mail->Port = 587; // TCP port to connect to

//Recipients

$mail->setFrom('from@example.com', 'Mailer');

$mail->addAddress('joe@example.net', 'Joe User'); // Add a recipient

$mail->addAddress('ellen@example.com'); // Name is optional

$mail->addReplyTo('info@example.com', 'Information');

$mail->addCC('cc@example.com');

$mail->addBCC('bcc@example.com');

//Attachments

$mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments

$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name

//Content

$mail->isHTML(true); // Set email format to HTML

$mail->Subject = 'Here is the subject';

$mail->Body = 'This is the HTML message body <b>in bold!</b>';

$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

$mail->send();

echo 'Message has been sent';

} catch (Exception $e) {

echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;

}

You'll find plenty more to play with in the [examples](https://github.com/PHPMailer/PHPMailer/tree/master/examples) folder.

That's it. You should now be ready to use PHPMailer!

**Localization**

PHPMailer defaults to English, but in the [language](https://github.com/PHPMailer/PHPMailer/tree/master/language/) folder you'll find numerous (48 at the time of writing!) translations for PHPMailer error messages that you may encounter. Their filenames contain [ISO 639-1](http://en.wikipedia.org/wiki/ISO_639-1) language code for the translations, for example fr for French. To specify a language, you need to tell PHPMailer which one to use, like this:

// To load the French version

$mail->setLanguage('fr', '/optional/path/to/language/directory/');

We welcome corrections and new languages - if you're looking for corrections to do, run the [PHPMailerLangTest.php](https://github.com/PHPMailer/PHPMailer/tree/master/test/PHPMailerLangTest.php) script in the tests folder and it will show any missing translations.

**Documentation**

Start reading at the [GitHub wiki](https://github.com/PHPMailer/PHPMailer/wiki). If you're having trouble, this should be the first place you look as it's the most frequently updated.

Examples of how to use PHPMailer for common scenarios can be found in the [examples](https://github.com/PHPMailer/PHPMailer/tree/master/examples) folder. If you're looking for a good starting point, we recommend you start with [the Gmail example](https://github.com/PHPMailer/PHPMailer/tree/master/examples/gmail.phps).

Note that in order to reduce PHPMailer's deployed code footprint, the examples are no longer included if you load PHPMailer via Composer or via [GitHub's zip file download](https://github.com/PHPMailer/PHPMailer/archive/master.zip), so you'll need to either clone the git repository or use the above links to get to the examples directly.

Complete generated API documentation is [available online](http://phpmailer.github.io/PHPMailer/).

You can generate complete API-level documentation by running phpdoc in the top-level folder, and documentation will appear in the docs folder, though you'll need to have [PHPDocumentor](http://www.phpdoc.org/) installed. You may find [the unit tests](https://github.com/PHPMailer/PHPMailer/tree/master/test/phpmailerTest.php) a good source of how to do various operations such as encryption.

If the documentation doesn't cover what you need, search the [many questions on Stack Overflow](http://stackoverflow.com/questions/tagged/phpmailer), and before you ask a question about "SMTP Error: Could not connect to SMTP host.", [read the troubleshooting guide](https://github.com/PHPMailer/PHPMailer/wiki/Troubleshooting).

**Tests**

There is a PHPUnit test script in the [test](https://github.com/PHPMailer/PHPMailer/tree/master/test/) folder. PHPMailer uses PHPUnit 4.8 - we would use 5.x but we need to run on PHP 5.5.

Build status:

If this isn't passing, is there something you can do to help?

**Security**

Please disclose any vulnerabilities found responsibly - report any security problems found to the maintainers privately.

PHPMailer versions prior to 5.2.22 (released January 9th 2017) have a local file disclosure vulnerability, [CVE-2017-5223](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-5223). If content passed into msgHTML() is sourced from unfiltered user input, relative paths can map to absolute local file paths and added as attachments. Also note that addAttachment (just like file\_get\_contents, passthru, unlink, etc) should not be passed user-sourced params either! Reported by Yongxiang Li of Asiasecurity.

PHPMailer versions prior to 5.2.20 (released December 28th 2016) are vulnerable to [CVE-2016-10045](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-10045) a remote code execution vulnerability, responsibly reported by [Dawid Golunski](https://legalhackers.com/advisories/PHPMailer-Exploit-Remote-Code-Exec-CVE-2016-10045-Vuln-Patch-Bypass.html), and patched by Paul Buonopane (@Zenexer).

PHPMailer versions prior to 5.2.18 (released December 2016) are vulnerable to [CVE-2016-10033](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-10033) a critical remote code execution vulnerability, responsibly reported by [Dawid Golunski](http://legalhackers.com/advisories/PHPMailer-Exploit-Remote-Code-Exec-CVE-2016-10033-Vuln.html).

See [SECURITY](https://github.com/PHPMailer/PHPMailer/tree/master/SECURITY.md) for more detail on security issues.

**Contributing**

Please submit bug reports, suggestions and pull requests to the [GitHub issue tracker](https://github.com/PHPMailer/PHPMailer/issues).

We're particularly interested in fixing edge-cases, expanding test coverage and updating translations.

If you found a mistake in the docs, or want to add something, go ahead and amend the wiki - anyone can edit it.

If you have git clones from prior to the move to the PHPMailer GitHub organisation, you'll need to update any remote URLs referencing the old GitHub location with a command like this from within your clone:

git remote set-url upstream https://github.com/PHPMailer/PHPMailer.git

Please *don't* use the SourceForge or Google Code projects any more; they are obsolete and no longer maintained.

**Sponsorship**