

Requirements for doing a normal installation

eZ Publish makes use of and depends on five important things:

1. A web server
2. The server-side PHP scripting engine
3. A database server
4. An image conversion system (optional)
5. The Zeta Components library

The first three things listed above should be in place before an eZ Publish installation is deployed. The image conversion system is optional and is only needed if you're planning to use eZ Publish with images. The web server and the server-side PHP scripting engine have to run on the same machine. The database server may run on a different computer.

[[Web server](#)] [[Server-side PHP scripting engine](#)] [[Zlib extension](#)] [[DOM extension](#)] [[PHP CLI](#)] [[CURL](#)] [[PHP memory limit](#)] [[PHP timezone](#)] [[Session parameters](#)] [[Other php configuration settings](#)] [[Database server](#)] [[MySQL](#)] [[PostgreSQL](#)] [[Oracle compatibility](#)] [[Image conversion system \(optional\)](#)] [[Limitation on some file systems when storing large number of content files](#)] [[eZ Publish on NFS file systems](#)] [[Network connectivity](#)]



Please visit the [requirements page](#) to check if your platform is fully supports installing eZ Publish, then come back for the extra details which follow.

Web server

See the [The web servers page](#)

Server-side PHP scripting engine

PHP is free software and can be downloaded from <http://www.php.net>. The full list of extensions needed can be found on the [requirements page](#). Info on some caveats can be found below.

Zlib extension

Make sure that [zlib](#) support in PHP is **enabled**, otherwise the [setup wizard](#) will not be able to unpack downloaded packages during the installation process.

DOM extension

In most cases, [DOM functions](#) are **enabled** by default as they are included in the PHP core. However, some Linux distributions have PHP without compiled-in support for DOM. Instead, they provide DOM as a shared module in a separate RPM package called "php-xml".

PHP CLI

Since version 5.0, it is necessary to **have PHP CLI installed**, as using php from the command line is needed during the setup process. Also, some features like [notifications](#), delayed search indexing, upgrade scripts, the collaboration system (content approval), clearing caches from within the command line, etc. will not work without php cli access.

CURL

It is **recommended** to **enable** [CURL](#) support, otherwise some features like [outbound connections via proxy](#) and SSL support for eZSoap will not work.

PHP memory limit

eZ Publish 5.3-alpha1/2004.01 and higher needs "memory_limit = 256M" in order to complete the setup wizard, or to execute its tests. Earlier releases needed "memory_limit= 128MB" as is the default in PHP.

php.ini

```
memory_limit = 256M

#prior to eZ Publish 5.3-alpha1/2004.01
#memory_limit = 128M
```

PHP timezone

You need to set the "`date.timezone`" value in the "`php.ini`" configuration file. If this setting is not specified, you will most likely receive error messages like "*It is not safe to rely on the system's timezone settings*" when running eZ Publish on PHP 5. The following example shows how the corresponding line in "`php.ini`" looks like:

php.ini

```
date.timezone = <timezone>
```



Refer to the [PHP documentation](#) for the list of supported timezones.

Don't forget to restart Apache after editing "`php.ini`".

Session parameters

eZ Publish sessions are handled by the **Symfony** stack, through **session handlers**. For that to be set up, additional `YAML` configuration are required, which you can find in the [Session chapter](#).

Other php configuration settings

See the [Symfony documentation](#) for the recommended `php.ini` settings for Symfony, as well as the [Vhost example](#) for the recommended `php.ini` settings for eZ Publish.

Database server

eZ Publish stores data structures and content using a **relational database**. This means that a database server has to be available for eZ Publish at all times. Follow this link to the [eZ Publish requirements page](#) to find which database solutions eZ Publish is compatible with.

eZ Publish 5 requires a **UTF-8 database** and support for **transactions**, which for MySQL means using the [InnoDB](#) storage engine.

The [Setup Wizard](#) will automatically detect the database server during operation of the setup wizard if it is running on the same computer that operates as web server.



Note that eZ Publish 5 does not support [clustering](#) mode for PostgreSQL databases. The clustering code is optimized for best performance on MySQL databases using the [InnoDB](#) storage engine.

MySQL

Even if you are not going to run eZ Publish in a clustered environment, the use of `InnoDB` is required. This storage engine makes it possible to use transaction-safe tables in a MySQL database.



Database transaction support is enabled by default in eZ Publish. This feature makes the system less vulnerable to database errors and inconsistencies due to aborted requests.


Contact your database administrator if you are unsure about whether `InnoDB` is available on your server.

MySQL can be tweaked with a lot of settings, but one setting which is required to **set to a higher value is innodb_buffer_pool_size**, by default it is set to 8MB, but it needs at least to be set to 128MB, or as the MySQL doc says up to 80% of system memory on dedicated database server.


Known issue with running PHP5.3 on MySQL: Some people (like Windows users with both IPv4 and IPv6 installed) experience problems connecting to the database server using host names like "localhost". If you experience problems, try using IPv4 address like "127.0.0.1". This is due to a connectivity problem when running PHP5.3 on MySQL. So, please replace the database server name "localhost" with the IP address of the machine, or "127.0.0.1", which is reserved for the local host.

PostgreSQL

If you want to use PostgreSQL, make sure the "pgcrypto" module is installed. On Linux/UNIX, you may need to install a separate package called "postgresql-contrib" which contains the "pgcrypto" module. The "pgcrypto" module provides cryptographic functions for PostgreSQL, including the "digest" function, which is needed for eZ Publish.


 Refer to the [PostgreSQL documentation](#) for more information

When setting up a PostgreSQL database for eZ Publish, you will have to **register these functions in the database**.

 Refer to the "Setting up a database" part of the "Installing eZ Publish on a Linux-UNIX based system" and "Installing eZ Publish on Windows" documentation pages (depending on the target OS) for more information.

Oracle compatibility

To be able to use eZ Publish on oracle you will need the [eZ Publish Extension for Oracle® Database](#) extension, as well as the php oci8 extension.

 Please note that installing eZ Publish via the setup wizard directly on an Oracle database is currently not supported.


 Supported only on eZ Publish 5.0, which is also the last version supporting Oracle on full legacy installations, which support does not include REST API usage or new "Symfony / eZ Platform" stack. Support is planned to return with one of the first releases of "eZ Platform", aka "6.x".


Image conversion system (optional)

In order to resize, convert or modify images, eZ Publish needs to make use of **an image conversion system**. One of the following software packages (both are free) can be used:

- **GD2** (comes with PHP)
- **ImageMagick**

ImageMagick supports more formats than GD and usually produces better results (better scaling, etc.). The setup wizard will automatically detect the pre-installed image conversion system(s).

The installation and setup of required software solutions (outlined above) is far beyond the scope of this document. Please refer to the homepage and documentation of the different software solutions.

 Composer will need GD to run.

Limitation on some file systems when storing large number of content files

eZ Publish stores all binary content (e.g. images, PDFs, etc.) in `var/storage` using a similar folder structure to the content tree, creating one folder for each object.

In most file systems used under Linux (especially ext2 + ext3) there is a hard limit of 32.000 sub-folders to the maximum which can be created in one folder. This means that it **is not possible to store more than 31999 objects under one parent object**.

To get around this limitation without changing the file system, you can **split your content tree** so that you don't have more than 32k content files (example: images) in the same folder.

Other file systems support more file/folder entries per folder:

- ext4: 64.000
- ReiserFS: roughly 1.2 million
- ZFS: 2^{48} (a really big number: 281474976710656)!



Note that those filesystems might not be fully supported by eZ Publish, please check out the [requirements page](#) for details.

eZ Publish on NFS file systems

Please be aware that it's not advisable to run eZ Publish on NFS file systems as you may experience issues. The cause of the issues may be **performance**, as NFS will slow down on heavy network traffic, slow access to files, or file access concurrency regarding file lock. Also, eZ Publish currently uses [the flock\(\) PHP function](#), which is not considered stable for NFS shares.



NFS should only be used to store distributed data such as cache, or binary files, in clustered environments.

Network connectivity

During execution of [the setup wizard](#), the web server will need to download some content from the internet.



If the web server can not access directly the internet, or if it has to go through a proxy, workarounds have to be taken. See the [setup wizard](#) documentation page for possible workarounds