

# Installation from source

This page will give you some advise how to install Z-Push from source ([GIT](#)) or from the distributed tarball.

## Requirements

Z-Push 2.3 runs only on PHP 5.4 or later  
A PEAR dependency as in previous versions does not exist in Z-Push 2.

The PHP version requirement is met in these distributions and versions (or later).

Distribution	Version
Debian	7.0 and later
Ubuntu	14.04 and later
RHEL/CentOS	6 and later
Fedora	23 and later
OpenSuse	13.2 and later
SLES	12 and later

If your distribution is not listed here, you can check which PHP version is default for it at <http://distrowatch.com/>.

For other relevant information related to distributions supported by Kopano/Zarafa please check the according admin manual.

## Additional php packages

To use the full featureset of Z-Push 2.3 and the z-push-top command line utility, additional php packages are required. These provide SOAP support, access to process control and shared memory.

Depending on the features you want to use, different packages are required. These packages vary in names between the distributions.

Dependencies for Z-Push

Distro	Packages
All	php-cli php-soap
Suse (SLES & OpenSuse)	php-soap php-pcntl php-posix php-mbstring
RHEL based systems (incl. CentOS)	php-cli php-soap php-process php-mbstring

\* In order to install these packages you need to add an extra channel subscription.

To use the shared memory IPC Provider, install:

Distro	Packages
Debian, Ubuntu, RHEL/Centos	no additional packages (but the above)

On Suse (SLES & OpenSuse)	php-sysvshm
	php-sysvsem

To use the memcached IPC Provider, install:

Distro	Packages
Debian, Ubuntu, Suse (SLES & OpenSuse)	php-memcached
RHEL based systems (incl. CentOS)	php-pecl-memcached

The memcache daemon is also required, normally it's available in the memcached package.

Most backends have own requirements, like Kopano/Zarafa depends on `php-mapi` while the IMAP backend depends on `php-imap`. See dependencies for each backend you plan to use individually.

## How to install

### 1. Extraction / getting sources

To install Z-Push, simply untar the z-push archive, e.g. with:

```
tar -xzf z-push-[version].tar.gz
```

The tar contains a folder which has the following structure:

```
z-push-[version]
```

The contents of the archive are similar to the content of a GIT checkout.

The only difference is that in GIT you will find three directories "src", "tools" and "config".

Contents of the "src" directory are located directly in the `z-push-[version]` folder, "tools" is a subdirectory of it.

The "config" directory of GIT is not being distributed in the tarball.

### 2. Directories

The contents of this folder should be copied to `/usr/share/z-push`.

In a case that `/usr/share/z-push` does not exist yet, create it with:

```
mkdir -p /usr/share/z-push
```

The directory should be owned by the webuser, e.g. `apache` or `www-data` depending on your distribution.

```
chown -R apache: /usr/share/z-push
```

```
cp -R z-push-[version]/* /usr/share/z-push/
```

By default the state directory is `/var/lib/z-push`, the log directory `/var/log/z-push`.

The locations can be changed in the configuration file. See next section on other options.

Make sure that these directories exist and are writable for your webserver process.

If you don't want to use the file statemachine (default) you don't need to create the state directory. More information on the different state machines is available here: [State Machines](#)

Either change the owner of these directories to the UID of your apache process or make the directories world writable :

```
chmod 755 /var/lib/z-push /var/log/z-push
```

```
chown apache:apache /var/lib/z-push /var/log/z-push
```

For the default webserver user please refer to your distribution's manual.

### 3. Z-Push configuration

Edit the `config.php` file in the Z-Push directory to fit your needs.

Here you can define the location of the states (if using the file statemachine) or alternatively configure the sql statemachine (file states are not required in this case).

If you intend to use Z-Push with Kopano backend and Kopano is installed on the same server, it should work out of the box without changing anything. Please also set your timezone in the `config.php` file.

The parameters and their roles are also explained in the `config.php` file.

By default the parameter `IPC_PROVIDER` (InterProcessCommunication) is empty in the configuration. This will lead to Z-Push using the first available provider.

Two providers are currently available:

- Shared memory provider (might require additional shared memory packages - same used in Z-Push 2.2.x)
- Memcache provider (requires php-memcached and a memcache server installed and configured)

The shared memory provider is the preferred IPC provider. If the `IPC_Provider` is not configured, the code will try to use shared memory (if available), the same as in Z-Push 2.2.x.

To force the usage of the memcached provider, set the `IPC_PROVIDER` parameter in the config to:

```
define('IPC_PROVIDER', 'IpcMemcachedProvider');
```

### 4. Webserver configuration

You must configure Apache to redirect the URL

`Microsoft-Server-ActiveSync` to the `index.php` file in the Z-Push directory. This can be done by adding the line:

```
Alias /Microsoft-Server-ActiveSync /usr/share/z-push/index.php
```

to your `httpd.conf` file. Make sure that you are adding the line to the correct part of your Apache configuration, taking care of virtual hosts and other Apache configurations.

Another possibility is to add this line to `z-push.conf` file inside the directory which contents are automatically processed during the webserver start (by default it is `conf.d` inside the `/etc/apache2` or `/etc/httpd` depending on your distribution).

You have to reload your webserver after making these configurations.

It's known that other webserver like nginx also work with Z-Push. Please feel free to contribute configuration steps and file in our wiki.

#### WARNING



#### Warning

You CANNOT simply rename the z-push directory to `Microsoft-Server-ActiveSync`. This will cause Apache to send redirects to the mobile device, which will definitely break your mobile device synchronisation.

Lastly, make sure that PHP has the following settings:

```
php_flag magic_quotes_gpc off
php_flag register_globals off
php_flag magic_quotes_runtime off
php_flag short_open_tag on
```

You can set this in the `httpd.conf`, in `php.ini` or in an `.htaccess` file in the root of z-push.

If you have several php applications on the same system, you could specify the z-push directory so these settings are considered only there.

```
<Directory /usr/share/z-push>
    php_flag magic_quotes_gpc off
    php_flag register_globals off
    php_flag magic_quotes_runtime off
    php_flag short_open_tag on
</Directory>
```

If you don't set this up correctly, you will not be able to login correctly via z-push.

Please also set a `memory_limit` for php to 128M or higher in your `php.ini`.

Z-Push writes files to your file system like logs or data from the FileStateMachine (which is default). In order to make this possible, you either need to disable the `php-safe-mode` in `php.ini` or `.htaccess` with

```
php_admin_flag safe_mode off
```

or configure it accordingly, so Z-Push is allowed to write to the log and state directories.

After doing this, you should be able to synchronize with your mobile device.

## Tools

To use the command line tools, access the installation directory (usually `/usr/share/z-push`) and execute:

```
./z-push-top.php          and/or
./z-push-admin.php
```

To facilitate the access symbolic links can be created, by executing:

```
ln -s /usr/share/z-push/z-push-admin.php /usr/sbin/z-push-admin
ln -s /usr/share/z-push/z-push-top.php /usr/sbin/z-push-top
```

With these symlinks in place the cli tools can be accessed from any directory and without the php file extension.

The usage of the tools is explained here:

[Tools: z-push-top](#)

[Tools: z-push-admin](#)

## Upgrade

Upgrading to a newer Z-Push version follows the same path as the initial installation.

### Upgrading from 1.4 to 1.4.1

When upgrading to a new minor version e.g. from Z-Push 1.4 to Z-Push 1.4.1, the existing Z-Push directory can be overwritten when extracting the archive. When installing a new major version it is recommended to extract the tarball to another directory and to copy the state from the existing installation.

#### Important

It is crucial to always keep the data of the state directory in order to ensure data consistency on already synchronized mobiles.

Without the state information mobile devices, which already have an ActiveSync profile, will receive duplicate items or the synchronization will break completely.

### Upgrading to 2.x from 1.x

Upgrading to Z-Push 2.X from 1.X it is not necessary to copy the state directory because states are not compatible. However Z-Push 2 implements a fully automatic re-synchronizing of devices in the case states are missing or faulty.

#### Important

Downgrading from Z-Push 2.X to 1.X is not simple. As the states are not compatible you would have to follow the procedure for a new installation and re-create profiles on every device.

#### Important

States of Z-Push 2.0 and Z-Push 2.1 are not compatible. A state migration script called `migrate-2.0.x-2.1.0.php` is available in the tools folder.

### Upgrade from 2.2.x to 2.3

When upgrading your states from Z-Push 2.2.x to 2.3 it's required to run `z-push-admin -a fixstates` once to ensure state compatibility.

See detailed instruction here: [Upgrade to Z-Push 2.3](#)

#### Important

When running Z-Push separately from your Kopano installation you had in the past to configure `MAPI_SERVER` directly in the `config.php` of Z-Push. This setting has now moved to the `config.php` file of the Kopano backend (`backend/kopano/config.php`).

Please also observe the published release notes of the new Z-Push version.  
For some releases it is necessary to e.g. re-synchronize the mobile.

## S/MIME

Z-Push supports signing and en-/decrypting of emails on mobile devices since the version 2.0.7.

### Important

Currently only Android 4.X and higher and iOS 5 and higher devices are known to support encryption/signing of emails.

It might be possible that PHP functions require CA information in order to validate certs. Therefore the `CAINFO` parameter in the `config.php` must be configured properly.

The major part of S/MIME deployment is the PKI setup. It includes the public-private key/certificate obtaining, their management in directory service and roll-out to the mobile devices. Individual certificates can either be obtained from a local (company intern) or a public CA. There are various public CAs offering certificates: commercial ones e.g. Symantec or Comodo or community-driven e.g. [CAcert.org](http://CAcert.org).

Both most popular directory services Microsoft Active Directory (MS AD) and free open source solution OpenLDAP allow to save certificates. Private keys /certificates reside in user's directory or on a smartcard. Public certificates are saved in directory. MS AD and OpenLDAP both use `userCertificate` attribute to save it.

In Active Directory the public key for contacts from GAB is saved in `PR_EMS_AB_TAGGED_X509_CERT (0x8C6A1102)` property and if you save a key in a contact it's `PR_USER_X509_CERTIFICATE (0x3A701102)`.

In LDAP public key for contacts from GAB is saved in `userCertificate` property. It should be mapped to `0x3A220102` in `ldap.propmap.cfg (0x3A220102 = userCertificate)`.

Make sure it looks like this in LDAP:

```
userCertificate;binary
MIIFGjCCBAKgAwIBAgIQbRnqpxlPaâ€|
```

### Important

It is strongly recommended to use MS AD or LDAP to manage certificates. Other user plugin options like db or unix might not work correctly and are not supported.

For in-depth information please refer to: <http://www.zarafa.com/blog/post/2013/05/smime-z-push-signing-and-en-decrypting-emails-mobile-devices>

## Setting up your mobile device

This is simply a case of adding an 'exchange server' to your ActiveSync server list, specifying the IP address of the Z-Push's apache server, disabling SSL, unless you have already setup SSL on your Apache server, setting the correct username and password (the domain is ignored, you can

simply specify 'domain' or some other random string), and then going through the standard ActiveSync settings.

Once you have done this, you should be able to synchronise your mobile simply by clicking the 'Sync' button in ActiveSync on your mobile.

**NOTE:** Using the synchronisation without SSL is not recommended because your private data is transmitted in clear text over the net. Configuring SSL on Apache is beyond of the scope of this document. Please refer to Apache documentation available at <http://httpd.apache.org/docs/>

## Troubleshooting

Most problems will be caused by incorrect Apache settings. To test whether your Apache setup is working correctly, you can simply type the Z-Push URL in your browser, to see if apache is correctly redirecting your request to Z-Push. You can simply use:

```
http://<serverip>/Microsoft-Server-ActiveSync
```

If correctly configured, you should see a username/password request and when you specify a valid username and password, you should see a Z-Push information page, saying that this kind of requests is not supported. Without authentication credentials Z-Push displays general information.

If not then check your PHP and Apache settings and Apache error logs.

If you have other synchronisation problems, you can increase the `LOGLEVEL` parameter in the config e.g. to `LOGLEVEL_DEBUG` or `LOGLEVEL_WBXML`.

The z-push.log file will then collect detailed debug information from your synchronisation.

**NOTE:** This setting will set Z-Push to log the detailed information for **every** user on the system. You can set a different log level for particular users by adding them comma separated to `$specialLogUsers` in the `config.php`

```
$specialLogUsers = array("user1", "user2", "user3");
```

**NOTE:** Be aware that if you are using `LOGLEVEL_DEBUG` and `LOGLEVEL_WBXML` Z-Push will be quite talkative, so it is advisable to use log-rotate on the log file.

## Repeated incorrect password messages

If a password contains characters which are encoded differently in `ISO-8859-1` and `Windows-1252` encodings (e.g. "Â§") the login might fail with Z-Push but it works fine with the WebApp/Webaccess. The solution is to add:

```
setlocale(LC_CTYPE, "en_US.UTF-8");
```

to the `config.php` file.