

Installing Totara

Prerequisites

It is recommended to place Totara web servers into [De-militarized Zones \(DMZs\)](#) instead of putting them directly into internal networks. This helps mitigate certain types of security attacks by limiting access from the site.



We strongly recommend that production sites are installed from scratch rather than being cloned from a database backup of another site.

You can read more in our [policy documents](#).

On this page

- [Prerequisites](#)
- [Install process](#)

Before installing Totara you should have a server with:

- A supported OS - Windows or Linux (Ubuntu, Debian, Suse, Redhat), Solaris10, other Unix variant.
- System clock synchronised with internet time servers.
- Functional outbound email.
- Supported webserver software (Apache 2.x.x or IIS only at this stage although nginx should also work).
- PHP (5.5.9 minimum).
- Required PHP modules: iconv, mbstring, curl, ctype, zip, simplexml, spl, pcre, dom, xml, json, gd, intl, zlib.
- Recommended PHP modules: openssl, tokenizer, xmlrpc, soap, opcache.
- PHP Memory limit of at least 256MB.
- Access to a supported database (PostgreSQL 9.2 or greater, MySQL 5.5.31 or greater, MariaDB 5.5.31, MSSQL 2008 or greater).



- If your database is Postgres you must set `standard_conforming_strings` to "off" and `escape_string_warning` to "off" in your **postgresql.conf** file.
- If your database is MSSQL, your database user must have the permissions: "ALTER SETTINGS(SERVER)".
- If your database is MySQL:
 - Increase the 'max_allowed_packet' setting to at least 4 megabytes.
 - Make sure InnoDB storage engine is available, legacy MyISAM engine is not supported any more.
 - Set 'innodb_file_per_table=1' and 'innodb_file_format=Barracuda' in [mysqld] section - these settings are required for compression of large database records.

It is recommended that Totara sites are installed via **SSL**.

- A domain name that resolves to your server.
- Recommended minimum browser: Recent Google Chrome, Recent Firefox, Safari 9, Internet Explorer 9 (IE 10 required for drag and drop of files from outside the browser into Totara).

Install process

1. If you are planning to use Totara in multiple user languages, you should ensure that the server is properly configured with the correct locales. This is important to ensure that date, number and currency data is displayed correctly for each language. See the [developer documentation](#) for more information on server locales.
2. Create directories on your server for www files ("wwwroot" - allow for 1GB) and sitedata files ("sitedata" - allow for 10GB). Ensure the sitedata directory is outside the wwwroot. Ensure that the sitedata directory is writeable by the web server process.
3. Configure your webserver to serve requests for your domain from the wwwroot directory.
4. Move the Totara source code into the wwwroot directory.
5. Create an empty database using the UTF8 character set.
 - For Postgres: `createdb -E utf8 [dbname]`
 - For MySQL/MariaDB: `CREATE DATABASE [dbname] DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;`

Ensure that you have a non-root database user with write access to the database.

6. Using your web browser, visit your domain and step through the installation process.
7. Set up a cron task to execute the file **admin/cli/cron.php** every minute.
8. Setting PHP variables. There are some PHP configuration variables that may need to be customised, depending on your needs. The variables that are commonly changed include: "memory_limit", "upload_max_filesize", "post_max_size", "max_input_vars" and "upload_tmp_dir".
 - **memory_limit** - This will need to be increased for restoring large course backups or unzipping large files within Totara.
 - **upload_max_filesize** - This setting in conjunction with **post_max_size** will determine how high you can set the max upload size within Totara.
 - **max_input_vars** - This setting will need to be increased to a minimum of '2000' if you would like to customise any of the Totara language packs.
 - **upload_tmp_dir** - Some customers may wish to enable this setting and specifically set a directory where files are temporarily stored during the upload process. Note the web server user must have permissions to write files in this directory, or all file uploads will fail.

If the opcache extension is installed then enabling opcache is recommended for performance reasons, so also set `opcache.enable = 1`

These settings are all located in **php.ini**. See <http://php.net/manual/en/ini.core.php> for more information (note that you will need to restart your webserver for these settings to take effect).