

Upgrading from Moodle

Prerequisites

Minimum requirements in Totara 9.0:

- PHP version 5.5.9 (7.1 and higher is not supported)
- PostgreSQL 9.2, MySQL 5.5.31, MariaDB 5.5.31 or MSSQL 2008.
- MSSQL only: Database user requires additional permissions: 'ALTER SETTINGS(SERVER)'.
- Oracle database is not supported.



Additional plugins, custom themes, or customised Moodle code are likely to cause conflicts in the upgrade process. Review possible conflicts before upgrading to prevent site stability issues.

As of Totara 9.0 you can no longer upgrade from versions of Moodle prior to 2.2.11. Ensure the Moodle site is updated to Moodle version 2.2.11+ before upgrading to Totara 9.0.

On this page

Upgrade process

1. Check the live logs to check if any users are currently using the site. The site will be offline while the upgrades are performed (*Site administration > Reports > Live Logs*).
2. Enable maintenance mode in Moodle (*Site administration > Server > Maintenance Mode*).
3. Back up the Moodle database.
4. Back up the moodledata directory.
5. Back up the Moodle source code directory.
6. Replace or merge the Moodle code.

- If your Moodle site is not customised you can replace your Moodle code folder with the new Totara code folder and copy your old **config.php** file to the new folder.



Do not copy the new code on top of the existing code folder.

- If you have customised Moodle you will need to merge your existing Moodle source code and new Totara code. This is best accomplished using a version control tool like GIT or SVN. You will need to:
 - Reuse the existing Moodle's **config.php** file.
 - Ensure any themes are updated to work with Totara.
 - Include any existing modules and filters remain in the code.
 - Resolve any merge conflicts caused by existing local customisations.
7. Navigate to **admin/index.php** using a web browser and perform the install process.
 8. Click **Yes** on the Upgrade page.
 9. Confirm the server status checks.
 10. Confirm the plugin status checks.
 11. Proceed with the initial database upgrades.
 12. Edit the **New Admin Settings** page if required.
 13. Continue the database upgrades.
 14. Check for warnings or errors on the upgrade review page.
 15. Edit the **New Site Settings** page if required.
 16. Test the site functionality. The site should now be displayed with the Totara logos and theme.
 17. Disable server maintenance mode.
 18. Congratulations, your site is now running Totara.

Password salting and module security

Totara automatically generates and adds a different salt for each user. A site-wide variable for the salt is no longer required for new installations of 2.5 or greater.

Backward compatibility for site upgrades



If you are upgrading a site from 2.4 or below and you are already using a site-wide salt in your configuration file, you need to keep using it to ensure your existing users can still log in.

A user's password hash will be converted to the new scheme when they login, however it might take a while until all users have logged in. Therefore you can force all your users to the new scheme by doing a force password reset as a bulk action.

Sites running PHP version below 5.3.7

Password hashing relies on bcrypt support from PHP, this is normally only available in PHP version 5.3.7 or greater. Therefore, if you are using a version of PHP which doesn't properly support bcrypt, Totara will fall back to the old password hashing scheme. It is recommended that you continue to use a site-wide salt until you are able to upgrade PHP.



An important fix bcrypt in the hashing algorithm was added PHP v5.3.7 which has also been backported in some distributions of Linux to earlier versions of PHP. This means that some earlier versions of PHP may still work.