

Cloning Totara sites from a database backup

When setting up a new site it is common to want to start with some default configuration and setup. One way to achieve this is to clone the database and sitedata from an existing site. When this is done, often everything will appear to work fine, but there are some subtle issues caused by cloning which can cause significant problems on a production site.

Therefore we strongly recommend doing a fresh install for any site which will be used in production. See below for more information on our recommended approach and things to consider if you are cloning from a backup.

Importance of resetting the site identifier

One significant issue that can occur when a site is cloned from a database backup is that you can end up with the same site identifier on multiple sites. The site identifier is stored in the `$CFGsiteidentifier` config setting and is required to be a unique value across sites. Many areas of code rely on the uniqueness of this variable (see below). If you restore a new site from a backup for any reason other than to move the site to a new URL you **MUST** unset the siteidentifier configuration variable. The system will automatically generate a new unique identifier when it is next required.

Potential issues with site identifier reuse

Below are some potential issues that we have identified that could occur on sites using the same site identifiers. This is not an exhaustive list, just some cases that highlight the issues:

Caching

The built-in caching system makes use of the site identifier to generate a unique key (see `get_cache_identifier()` in `/cache/classes/definition.php`). If two sites with the same identifier are making use of the same cache store, there is potential for the cached data from one site to be incorrectly passed to the other site. This could include sensitive information, or system critical information such as configuration data. Incorrect use of cached data from another site has the potential to critically damage the site(s) - for example if cached configuration or database table structure data from one site is used by code on another.

Backup and restore

Backup and restore uses the site identifier to tell if a backup is being restored on the same site as it was generated on or not. The system behaves differently depending on if the source and destination match or not, for example instead of creating a new user the system may assign restored data to an existing user if it believes the user matches (see comments above `precheck_user()` in `/util/dbops/restore_dbops.class.php`). While there are additional checks in place (the user id and username must both match) there is still the potential for incorrect assignment of data if a backup from one site is restored on a different site when they share the same site identifier, particularly if users are cloned as part of the setup of the new site.

LTI

LTI makes use of the site identifier to generate secret tokens which are used to authenticate access to LTI packages (see `generate_cartridge_token()` in `/enrol/lti/classes/helper.php`). Theoretically a user with access to one site could authenticate to a course on any other site that shares the same site identifier, if there is a course which shares the same course id as the course they have access to on their site. This would represent a significant security hole.

Recommended approach to creating site instances

Our recommended approach to setting up a new site would be as follows:

- Create a new site from scratch with an empty database and data root.
- Install the site using the normal install procedure (CLI or via the web).
- Consider creating a flavour plugin (see comments in `/totara/flavour/classes/definition.php` for how to create a flavour) to set initial configuration and default settings.
- To set up any default content, create a local plugin and use `db/install.php` or CLI scripts to create any default content.

Considerations if restoring from database backups

While we strongly recommend against doing this for the reasons above, and cannot provide support for any issues arising due to sites set up in this way, if you do choose to create a site this way we recommend you consider the following carefully:

- You **MUST** unset the site identifier configuration setting on the new site, unless:
 1. You are moving the site to a new URL (in which case it should stay the same).
 2. You are copying a production database to a staging or test server for testing purposes (you should be fine as long as your test server is properly isolated from the production instance)
- Your database may include references to the site URL which you will likely want to replace. There is a script which does this (`/admin/cli/replace.php`) but we do not support its use and cannot make any guarantees that it will work correctly.
- You should truncate any site logs immediately after restoring the site.
- We recommend keeping the amount of content in the site being backed up to an absolute minimum. The less content there is the less chance there will be a problem.

What to do if you have existing sites that share a site identifier

We would recommend that if at all possible you should unset the site identifier and generate a new one by calling the moodlelib.php function `get_site_identifier()`. However this will need to be assessed on a case by case basis taking into account the following possible impacts:

- Ensure you have a full backup of your database and site data prior to unsetting the site identifier.
- Caches will be invalidated. Clear your cache immediately after unsetting the site identifier.
- Be aware that any backups generated on the site that you restore to the same site will now treat the site as if it is a different site instead of the same one.
- Carefully review the code of any third party plugins to identify any dependencies on the site identifier and determine how they will behave.