

Migrating to an External Database

For details on using an external database from the first TeamCity start, as well as the general external database information and the database-specific configuration steps, refer to the [Setting up an External Database](#) page.

The current section covers the steps required to migrate TeamCity data from one database to another. The most typical case is when you evaluated TeamCity with the default internal database and need to switch to an external database to prepare your TeamCity installation for production use. The steps here are also applicable when switching from one external database to another.



Database migration cannot be combined with the server upgrade. If you want to upgrade at the same time, you should first [upgrade](#), run the new version of TeamCity, and then migrate to another database.

There are several ways to migrate data into a new database:

- [Switch](#) with no data migration: build configurations settings will be preserved, but not the historical builds data or users.
- [Full Migration](#): all the data is preserved except for any database-stored data provided by the third-party plugins.
- [Backup and then restore](#): the same as full migration, but using the two-step approach.

Switch with No Data Migration

If you want a fast switch to an external database and do not want to preserve existing data like users and builds on the server, follow the steps below. See [#Full Migration](#) for preserving all the data.

After the switch, the server will start with an empty database, but preserve all the settings stored under TeamCity Data Directory (see [details](#) on what is stored where).

Steps to perform the switch:

1. [Create and configure an external database](#) to be used by TeamCity.
2. Shut down the TeamCity server.
3. [Create a backup copy](#) of the `<TeamCity Data Directory>` used by the server.
4. Clean up the `system` folder: you must remove the `messages` and `artifacts` folders from the `/system` folder of your `<TeamCity data directory>`; you may delete the old HSQLDB files: `buildserver.*` to remove the no longer needed internal storage data.
5. Start the TeamCity server.



If you see the TeamCity Maintenance screen, click the "I'm a server administrator, show me the details" link and enter the [Maintenance Authentication Token](#). Follow the instructions to create a new TeamCity database.

Full Migration

These steps describe switching to another database preserving all data. The TeamCity migration tool, the `maintainDB` command line utility, is available for this purpose.

The `maintainDB.[cmd|sh]` shell/batch script is located in the `<TeamCity Home>/bin` directory and is used for migrating as well as for [backing up](#) and [restoring](#) TeamCity data.

The utility is only available in the TeamCity `.tar.gz` and `.exe` distributions. If you installed TeamCity using the `.war` distribution and need to perform data migration, use the tool from the `.tar.gz` distribution.

TeamCity supports HSQLDB, MySQL, Oracle, PostgreSQL and Microsoft SQL Server; the migration is possible between any of these databases.



The target database must be empty before the migration process (it must NOT contain any tables).




If an error occurs during migration, do not use the new database as it may result in the database data corruption or various errors that can uncover later. In case of an error, investigate the reason logged into the console or in the migration logs (see below), and, if a solution is found, clean the target database and repeat the migration process.

To migrate all your existing data to a new external database:

1. [Create and configure an external database](#) to be used by TeamCity and install the database driver into TeamCity. Do not modify any TeamCity settings at this stage.
2. Shut down the TeamCity server.
3. Create a temporary properties file with a custom name (for example, `database.<database_type>.properties`) for the target database using the corresponding template (`<TeamCity Data Directory>/config/database.<database_type>.properties.dist`). Configure the properties and place the file into any temporary directory. Do not modify the original `database.<database_type>.properties` file.
4. Run the `maintainDB` tool with the `migrate` command and specify the absolute path to the newly created target database properties file with the `-T` option:

```
maintainDB.[cmd|sh] migrate [-A <path to TeamCity Data Directory>] -T <path to database.properties file>
```

 If you have the `TEAMCITY_DATA_PATH` environment set (pointing to the [TeamCity Data Directory](#)), you do not need the `-A <path to TeamCity Data Directory>` parameter of the tool.

Upon the successful completion of the database migration, the temporary file will be copied to (`<TeamCity Data Directory>/config/database.properties`) file which will be used by TeamCity. The temporary file can be safely deleted.

If you are migrating between external databases, the original `database.properties` file for the source database will be replaced with the file specified via the `-T` option. The original `database.properties` file will be automatically re-named to `database.properties.before.<timestamp>`.

5. Start the TeamCity server. This must be the same TeamCity version that was run last time (TeamCity [upgrade](#) must be performed as a separate procedure).

After you make sure the migration succeeded, you may delete the old HSQLDB files: `buildserver.*` to remove the no longer needed internal storage data.

Backup and Restore

You can [create a backup](#) and then [restore it](#) using different target database settings. You will probably need to specify restore options to restore only the database data.

Troubleshooting

- Extended information during migration execution is logged into the `logs\teamcity-maintenance.log` file. Also, `logs\teamcity-maintenance-truncation.log` contains extended information on possible data truncation during the migration process.
- If you encounter an "out of memory" error, try increasing the number in the `-Xmx512m` parameter in the `maintainDB` script. On a 32-bit platform, the maximum is about 1300 megabytes. Alternatively, run HSQLDB in the standalone mode via

```
java -Xmx256M -cp ..\webapps\ROOT\WEB-INF\lib\hsqldb.jar org.hsqldb.Server -database.0 <TeamCity data directory>\system\buildserver -dbname.0 buildserver
```

and then run the Migration tool pointing to the database as the source: `jdbc:hsqldb:hsq1://localhost/buildserver sa ''`

- If you get "The input line is too long." error while running the tool (e.g. this may happen on Windows 2000), change the script to use an alternative classpath method. For `maintainDB.bat`, remove the lines below "Add all JARs from WEB-INF\lib to classpath" comment and uncomment the lines below "Alternative classpath: Add only necessary JARs" comment.

See also:

Installation and Upgrade: [Common database-related problems](#)

Installation and Upgrade: [Setting up an External Database](#)

Concepts: [TeamCity Data Directory](#)

Administrator's Guide: [TeamCity Data Backup](#)