Setting up an External Database

This page covers external database setup for the first use with TeamCity 8.1 and above. For earlier versions, select the documentation corresponding to your TeamCity version.

TeamCity stores build history, users, build results and some run time data in an SQL database. See also the description of what is stored where on the Manual Backup and Restore page.

If you evaluated TeamCity with the internal database which is not recommended for production, please refer to Migrating to an External Database.

On this page:

Default Internal Database

On the first TeamCity run, using an internal database based on the HSQLDB database engine is suggested by default. The internal database suits evaluation purposes only; it works out of the box and requires no additional setup.

However, we strongly recommend using an external database as a back-end TeamCity database in a production environment. An external database is usually more reliable and provides better performance: the internal database may crash and lose all your data (e.g. on the "out of disk space" condition). Also, the internal database may become extremely slow on large data sets (say, database storage files over 200Mb). Please also note that our support does not cover any performance or database data loss issues if you are using the internal database.

In short, do not EVER use internal HSQLDB database for production TeamCity instances. Migrate to an external database the moment you start to rely on the data stored in TeamCity server.

Selecting External Database Engine

As a general rule you should use the database that better suits your environment and that you can maintain/configure better in your organization.
While we strive to make sure TeamCity functions equally well under all of the supported databases, issues can surface in some of them under high TeamCity-generated load.

You may also want to estimate the required database capacity.

Supported Databases

TeamCity supports the following databases:

- MySQL
- PostgreSQL
- Oracle
- MS SQL

General Steps

1. Configure the external database to be used by TeamCity (see the database-specific sections below).
2. Run the TeamCity server for the first time.
3. Select an external database to be used and specify the database connection settings.
   If required, you can later manually modify your database connection settings.
   ⚠ Note that TeamCity creates its own database schema on the first start and actively modifies it during the upgrade. The schema is not changed when TeamCity is working normally.
   The user account used by TeamCity should have permissions to create new, modify and delete existing tables in its schema, in addition to usual read/write permissions on all tables.
4. You may also need to download the JDBC driver for your database.
   Due to licensing terms, TeamCity does not bundle driver jars for external databases. You will need to download the Java JDBC driver and put the appropriate `.jar` files (see driver-specific sections below) from it into the `<TeamCity Data Directory>/lib/jdbc` directory.
   Please note that the `.jar` files should be compiled for the Java version not greater than the one used to run TeamCity, otherwise you might see "Unsupported major.minor version" errors related to the database driver classes.

⚠ See additional information if Amazon Aurora DB Cluster is used as the TeamCity database server.

Database-specific Steps

The section below describes the required configuration on the database server and the TeamCity server.

MySQL

Supported versions

On MySQL server side

Recommended database server settings:

- use InnoDB storage engine
- use UTF-8 character set
- use case-sensitive collation
- see also recommendations for MySQL server settings

The MySQL user account that will be used by TeamCity must be granted all permissions on the TeamCity database.
This can be done by executing the following SQL commands from the MySQL console:

```
create database <database-name> collate utf8_bin;
create user <user-name> identified by '<password>';
grant all privileges on <database-name>.* to <user-name>;
grant process on *.* to <user-name>;
```

On TeamCity server side (with MySQL)

JDBC driver installation:

1. Download the MySQL JDBC driver from http://dev.mysql.com/downloads/connector/j/. If the MySQL server version is 5.5 or newer, the JDBC driver version should be 5.1.23 or newer.
2. Place `mysql-connector-java-*-bin.jar` from the downloaded archive into the `<TeamCity Data Directory>/lib/jdbc`. Proceed with the TeamCity setup.

PostgreSQL

Supported versions

On PostgreSQL server side

1. Create an empty database for TeamCity in PostgreSQL.
    - Make sure to set up the database to use UTF8.
    - Grant permissions to modify this database to the user account used by TeamCity to work with the database.
2. See also recommendations for PostgreSQL server settings

TeamCity does not specify which schema will be used for its tables. By default, PostgreSQL creates tables in the 'public' schema ('public' is the name of the schema). TeamCity can also work with other PostgreSQL schemas. To switch to a different schema, do the following:
Create a schema named exactly as the user name: this can be done using the `pgAdmin` tool or with the following SQL:

```
create schema teamcity authorization teamcity;
```

The schema has to be empty (it must not contain any tables).

On TeamCity server side (with PostgreSQL)

Download the required PostgreSQL JDBC42 driver and place it into the `<TeamCity Data Directory>/lib/jdbc`. Proceed with the TeamCity setup.

Set the `connectionProperties.autosave=conservative` in the database properties file to address this issue with PostgreSQL JDBC driver.

Oracle

Supported versions

On Oracle server side

1. Create an Oracle user account/schema for TeamCity.

    TeamCity uses the primary character set (char, varchar, clob) for storing internal text data and the national character set (nchar2, nvarchar, nclob) to store the user input and data from external systems, like VCS, NTLM, etc.

    - Make sure that the national character set of the database instance is UTF or Unicode.
    - Grant the CREATE SESSION, CREATE TABLE, permissions to a user whose account will be used by TeamCity to work with this database.TeamCity, on the first connect, creates all necessary tables and indices in the user's schema. (Note: TeamCity never attempts to access other schemas even if they are accessible)

        Make sure TeamCity user has quota for accessing table space.

On TeamCity server side (with Oracle)

1. Get the Oracle JDBC driver.
   Supported driver versions are 11.1 and higher.
   Place the following files:
    - ojdbc7.jar (or ojdbc6.jar if ojdbc7.jar is not available for your database version)
    - orai18n.jar (can be omitted if missing in the driver version) into the `<TeamCity Data Directory>/lib/jdbc` directory.

        The Oracle JDBC driver must be compatible with the Oracle server.

        It is strongly recommended to locate the driver in your Oracle server installation. Contact your DBA for the files if required.
        Alternatively, download the Oracle JDBC driver from the Oracle web site. Make sure the driver version is compatible with your Oracle server.
2. Proceed with the TeamCity setup.

Microsoft SQL Server

Supported versions

For step-by-step instructions, see the dedicated page. The current section provides key details required for the setup.

On MS SQL server side

TeamCity uses the primary character set (char, varchar, text) for storing internal text data and the national character set (nchar, nvarchar, ntext) to store the user input and data from external systems, like VCS, NTLM, etc.

1. Create a new database. As the primary collation, use the case-sensitive collation (collation name ending with '_CS_AS') corresponding to your locale.
2. Create a TeamCity user and ensure that this user is the owner of the database (grant the user dbo rights), which will give the user the ability to modify the database schema.
   For SSL connections, ensure that the version of MS SQL server and the TeamCity version of java are compatible. We recommend using the latest update of SQL server.
3. Allocate sufficient transaction log space depending on how intensively the server will be used. The recommended setup is not less then 1Gb.
4. Make sure SQL Server Browser is running.
5. Make sure TCP/IP protocol is enabled for SQL Server instance.
6. Make sure that the "no count" setting is disabled: the enabled "no count" setting prevents TeamCity from starting queued builds.

On TeamCity server side (with MS SQL)

1. Download the Microsoft JDBC driver v6.0+ (sqljdbc_6.0.x package, .exe or .tar.gz depending on your TeamCity server platform) from the Microsoft Download Center.
2. Unpack the downloaded package into a temporary directory.
3. Copy the `sqljdbc42.jar` corresponding to jre8 from the just downloaded package into the TeamCity Data Directory/lib/jdbc directory. For driver version 6.2 use `mssql-jdbc-*.jre8.jar` file instead. MS SQL integrated security (Windows authentication) requires installing `sqljdbc_auth.dll` from the driver package as per instructions.
4. Proceed with the TeamCity setup.

jTDS driver

It is not recommended to use jTDS JDBC driver. There are at least known issues with using Unicode characters when the driver is in use.

If you use the driver ("jtds" text appears in the connectionUrl of database.properties), it is highly recommended to switch the native driver.

The due procedure for the switch is to:

- create the server backup including the database
- stop the server and configure the server to use native Microsoft JDBC driver as noted in the section above
- restore the database form the backup into a new MS SQL database
- run the server

Database Configuration Properties

The database connection settings are stored in `<TeamCity Data Directory>\config\database.properties` file. The file is a Java properties file. You can modify it to specify required properties for your database connections.

TeamCity uses Apache DBCP for database connection pooling. Please refer to http://commons.apache.org/dbcp/configuration.html for detailed description of configuration properties.

For all supported databases there are template files with database-specific properties located in the `<TeamCity Data Directory>/config` directory. The files have the following naming format: `database.<database_type>.properties.dist` and can be used as a reference on the required settings.

See also:

Installation and Upgrade: Common database-related problems | Migrating to an External Database
Concepts: TeamCity Data Directory
Administrator's Guide: TeamCity Data Backup