

Configuring the Database

Overview

Artifactory comes with a built-in embedded Derby database that can be reliably used to store data (metadata) for production-level repositories up to hundreds of gigabytes in size.

However, Artifactory supports pluggable database implementations allowing you to change the default to use other popular databases.

Artifactory currently supports the following databases:

- Derby (The default embedded database)
- [MySQL v5.5 and above with InnoDB](#)
- [Oracle version 10g and above](#)
- [Microsoft SQL Server 2008 and above](#)
- [PostgreSQL v9.2 and above](#)
- [MariaDB v10.2.9 and above](#)

For each of the supported databases you can find the corresponding properties file inside `$ARTIFACTORY_HOME/misc/db`.

Choosing the Right Database

As the default database, Derby provides good performance since it runs in the same process as Artifactory, however, under intensive usage or high load, performance may be degraded since Artifactory and the database compete for shared JVM resources such as caches and memory. Therefore, for Artifactory servers that need to support heavy load, you may consider using an external database such as MySQL or PostgreSQL which are very common choices in many Artifactory installations.

Any of the other supported databases is also a fair choice and may be the practical choice to make if your organization is already using one of them.



Accessing a Remote Database

When using an external database, you need a reliable, stable and low-latency network connection to ensure proper functioning of your system.

When using a [fullDB](#) configuration, we strongly recommend a high-bandwidth to accommodate the transfer of large BLOBs over the network.

Modes of Operation

Artifactory supports two modes of operation:

- Metadata in the database and binaries stored on the file system (This is the default and recommended configuration).
- Metadata and binaries stored as BLOBs in the database

Checksum-Based Storage

Artifactory uniquely stores artifacts using checksum-based storage. For details, please refer to [Checksum-Based Storage](#).

Page Contents

- [Overview](#)
 - [Choosing the Right Database](#)
 - [Modes of Operation](#)
 - [Checksum-Based Storage](#)
- [Before You Start](#)
 - [Backup Your Current Installation](#)
- [Setup the New Database](#)
 - [Advanced Settings](#)

Read more

- [MySQL](#)
- [Oracle](#)
- [Microsoft SQL Server](#)
- [PostgreSQL](#)
- [MariaDB](#)

Before You Start



Preprocessing

Changing the database does not automatically transfer your data to the new database. Please follow the steps below to backup your data so that you can restore it after the change.

Backup Your Current Installation

When changing the database for an existing installation you must first perform a [Full System Export](#) using the "Exclude Content" option. Once your new database is set up and configured, you will import this data to re-populate your Artifactory metadata content.

Make sure to [backup](#) your current Artifactory system before updating to a new database. You will need your Artifactory instance to be disconnected from the network to avoid usage during this procedure.

Setup the New Database

To setup your new database you need to perform the following steps:

- Create a database instance
- Create an Artifactory user for the database
- Install the appropriate JDBC driver
- Copy the relevant database configuration file
- Configure the corresponding `db.properties` file.
- Start Artifactory
- Import the metadata using [Full System Import](#)

These steps are fully detailed in the specific documentation page for each of the supported databases listed in the [Overview](#).

Advanced Settings

Once you have setup your database, you can configure it to support your expected load with the following two parameters:

<code>pool.max.active</code>	The maximum number of pooled database connections (default: 100).
<code>pool.max.idle</code>	The maximum number of pooled idle database connections (default: 10).