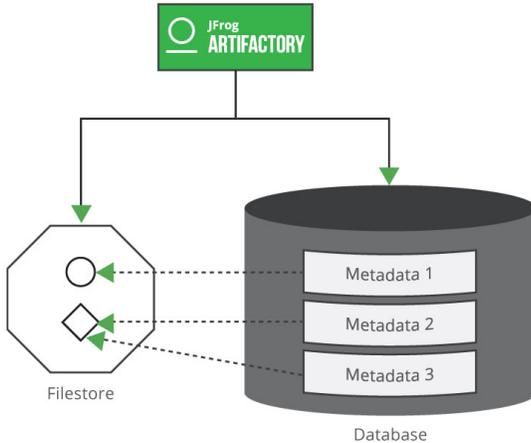


Checksum-Based Storage

Checksum-Based Storage Overview

Artifactory uniquely stores artifacts using checksum-based storage.

A file that is uploaded to Artifactory, first has its SHA1 checksum calculated, and is then renamed to its checksum. It is then hosted in the configured filestore in a directory structure made up of the first two characters of the checksum. For example, a file whose checksum is "ac3f5e56..." would be stored in directory "ac"; a file whose checksum is "dfe12a4b..." would be stored in directory "df" and so forth. The example below shows the "d4" directory that contains two files whose checksum begins with "d4".



In parallel, Artifactory creates a database entry mapping the file's checksum to the path it was uploaded to in a repository. This way of storing binaries optimizes many operations in Artifactory since they are implemented through simple database transactions rather than actually manipulating files.

Page Contents

- [Checksum-Based Storage Overview](#)
- [Checksum-Based Storage Implementation](#)
 - [Deduplication](#)
 - [Copying and Moving Files](#)
 - [Deleting Files](#)
 - [Upload, Download, and Replication](#)
 - [Filesystem Performance](#)
 - [Checksum Search](#)
 - [Flexible Layout](#)

Checksum-Based Storage Implementation

The following sections provide more information on how checksum-based storage features are implemented in Artifactory.

Deduplication

Artifactory stores any binary file only once. This is what we call "once and once only storage". First time a file is uploaded, Artifactory runs the required checksum calculations when storing the file, however, if the file is uploaded again (to a different location, for example), the upload is implemented as a simple database transaction that creates another record mapping the file's checksum to its new location. There is no need to actually store the file again in storage. No matter how many times a file is uploaded, the filestore only hosts a single copy of the file.

Copying and Moving Files

Copying and moving a file is implemented by simply adding and removing database references and, correspondingly, performance of these actions is that of a database transaction.

Deleting Files

Deleting a file is also a simple database transaction in which the corresponding database record is deleted. The file itself is not directly deleted, even if the last database entry pointing to it is removed. So-called "orphaned" files are removed in the background by Artifactory's garbage collection processes.

Upload, Download, and Replication

Before moving files from one location to another, Artifactory sends checksum headers. If the files already exist in the destination, they are not transferred even if they exist under a different path.

Filesystem Performance

Filesystem performance is greatly improved because actions on the filestore are implemented as database transactions, so there is never any need to do a write-lock on the filesystem.

Checksum Search

Searching for a file by its checksum is extremely fast since Artifactory is actually searching through the database for the specified checksum.

Flexible Layout

Since the database is a layer of indirection between the filestore and the displayed layout, any layout can be supported, whether for one of the standard packaging formats such as Maven1, Maven2, npm, NuGet etc. or for any custom layout.