

# Importing and Exporting

## Overview

Artifactory supports import and export of data at two levels:

- [System level](#)
- [Repository level](#)

At **system level**, Artifactory can export and import the whole Artifactory server: configuration, security information, stored data and metadata. The format used is identical to the [System Backup](#) format. This is useful when manually running backups and for migrating and restoring a complete Artifactory instance (as an alternative to using database level backup and restore).

At **repository level**, Artifactory can export and import data and metadata stored in a repository. This is useful when moving store data, including its metadata between repositories and for batch population of a repository.

### Page Contents

- [Overview](#)
- [System Import and Export](#)
  - [System Import and Export for an HA Cluster](#)
- [Repositories Import and Export](#)
  - [Export](#)
  - [Import](#)
    - [Import Layout](#)

---

## System Import and Export



### Excluding builds from version 6.6 and above

From Artifactory version 6.6, you can exclude builds only using the [System Export REST API](#). This is no longer available from the UI.



Note: The System Import and Export feature is not available for Artifactory Cloud users.

To access import and export of your entire system, in the **Admin** module, select **Import & Export | System**

## System Import & Export

### Export System

Export Path on Server \*

Browse

- Exclude Content
- Exclude Metadata
- Create .m2 Compatible Export [?](#)
- Create a Zip Archive (Slow and CPU Intensive!)
- Output Verbose Log [?](#)

Export

### Import System

**!** This action will wipe all Artifactory content - make sure to back up before completing this action!

Import Zip or Path on Server \*

Browse

- Exclude Content
- Exclude Metadata
- Output Verbose Log [?](#)



Import



#### Maintaining security entities on system import

From version 5.10, Artifactory maintains all security entities (users, groups, permissions and access tokens) when doing a system import. Previously, these entities would be removed along with existing configurations, repositories and artifacts.

Target Export Dir	The target directory for the exported files. You may browse your file system to select the directory
Exclude Content	<b>Export:</b> When set, repository binaries are excluded from the export. <b>Import:</b> When set, binaries and metadata are excluded from the import. Only builds and configuration files are imported.

Exclude Metadata	<p>When set, repository metadata are excluded from the import/export. (Maven 2 metadata is unaffected by this setting)</p> <div data-bbox="448 205 1463 338" style="border: 1px solid #f96; padding: 10px;"> <p> <b>Docker repositories must have metadata</b></p> <p>For Docker repositories to work they must have their metadata intact. Therefore, if you have Docker repositories, make sure that <b>Exclude Metadata</b> is not checked when doing a system export or import.</p> </div>
Exclude Builds	When set, all builds are excluded from the export
Create .m2 Compatible Export	When set, includes Maven 2 repository metadata and checksum files as part of the export
Create a Zip Archive (Slow and CPU Intensive!)	When set, creates and exports to a Zip archive
Output Verbose Log	<p>When set, lowers the log level to "debug" and redirects the output from the standard log to the import-export log.</p> <div data-bbox="448 657 1463 762" style="border: 1px solid #ccc; padding: 10px;"> <p> <b>Monitoring the log</b></p> <p>You can monitor the log in the <a href="#">System Logs</a> page.</p> </div>

The source/target of the import/export operations are folders (Zip archives are not recommended) on the Artifactory server itself.

You can use the built-in server-side browsing inside Artifactory to select server-side source/target folders:

## Server File System Browser ✕

Mount Point: Path to export

📁

Directory To Export ?

Cancel
Select

Importing or exporting a large amount of data may be time consuming. During the import/export operation you can browse away from the page and sample the [System Logs](#) to monitor progress.

## System Import and Export for an HA Cluster

When performing a system export and subsequent import for an HA cluster, you need to follow the procedure below to ensure that the cluster is able to correctly synchronize its nodes.

- Perform a normal system export from the source cluster as described [above](#)
- In the target cluster, keep the primary node running, and perform a graceful shutdown of all secondary nodes
- Perform normal system import to the target cluster (which now has only the primary node running) as described [above](#)
- Perform a graceful shutdown of the primary node and then restart it
- For each secondary node:
  - Delete the following folders
    - `$ARTIFACTORY_HOME/access`
    - `$ARTIFACTORY_HOME/etc/security`
    - `$ARTIFACTORY_HOME/etc/ui`
    - `$ARTIFACTORY_HOME/etc/plugins`
  - Copy the `$ARTIFACTORY_HOME/etc/security/master.key` from the primary node to the secondary node.



### Master Key

Copying the `$ARTIFACTORY_HOME/etc/security/master.key` is a critical step in the import process.

- Start up the secondary node
- Perform a normal system export from the source cluster as described [above](#)
- In the target cluster, keep the primary node running, and perform a graceful shutdown of all secondary nodes
- Perform normal system import to the target cluster (which now has only the primary node running) as described [above](#)

- Perform a graceful shutdown of the primary node and then restart it
- [Create the bootstrap bundle](#) on the primary node
- For each secondary node:
  - Delete the following folders
    - `$ARTIFACTORY_HOME/access`
    - `$ARTIFACTORY_HOME/etc/security`
    - `$ARTIFACTORY_HOME/etc/ui`
    - `$ARTIFACTORY_HOME/etc/plugins`
  - Delete the `$ARTIFACTORY_HOME/etc/db.properties` file
  - Delete the `$ARTIFACTORY_HOME/etc/binarystore.xml` file
  - Copy the bootstrap bundle you created on the primary node, `bootstrap.bundle.tar.gz`, to the `$ARTIFACTORY_HOME/etc` folder on the secondary node.



#### Bootstrap Bundle and db.properties

This is a critical step in the import process. The bootstrap bundle must be installed in each secondary node before you start it up for it to operate correctly in the cluster. Note also, if the `$ARTIFACTORY_HOME/etc` folder in your secondary node already contains a `db.properties` file, it will prevent the bootstrap bundle from being properly extracted when you start up the secondary node causing the import to fail.

- Start up the secondary node

Once you have completed the import, we recommend verifying that your HA cluster is up and running normally as described in [Testing your HA Configuration](#).

## Repositories Import and Export



Note: Importing repositories is supported for all users whereas exporting repositories is only available for non-SaaS users.

To access import and export of repositories, in the **Admin** tab, select **Import & Export | Repositories**

### Export

#### Export Repository to Path

**Target Local Repository\***

All Repositories ▾

Export to Path\*


Exclude Metadata

Create .m2 Compatible Export

Output Verbose Log

When exporting, you need to specify the following parameters:

Source Local Repository	You can specify a single repository to export, or <b>All Repositories</b>
Export to Path	The export target directory on your server
Exclude Metadata	When set, repository metadata are excluded from the export.(Maven 2 metadata is unaffected by this setting)

Create .m2 Compatible Export	When set, includes Maven 2 repository metadata and checksum files as part of the export
Output Verbose Log	<p>When set, lowers the log level to "debug" and redirects the output from the standard log to the import-export log.</p> <div style="border: 1px solid #ccc; padding: 5px;"> <p> <b>Monitoring the log</b></p> <p>You can monitor the log in the <a href="#">System Logs</a> page.</p> </div>

## Import

You can import repositories from a server side folder, or by zipping a repository and uploading it to Artifactory.

### Import Repository from Path

Target Local Repository\*

debian-local

Server Path For Import\*


Browse

Exclude Metadata

Output Verbose Log

Import

When importing, you need to specify the following parameters:

Target Local Repository	You can specify a single repository to import, or <b>All Repositories</b> . The repository layout should be different depending on your selection. Please refer to <a href="#">Import Layout</a>
Server Path for Import	The import source directory on your server
Exclude Metadata	When set, repository metadata are excluded from the import
Output Verbose Log	<p>When set, lowers the log level to "debug" and redirects the output from the standard log to the import-export log.</p> <div style="border: 1px solid #ccc; padding: 5px;"> <p> <b>Monitoring the log</b></p> <p>You can monitor the log in the <a href="#">System Logs</a> page.</p> </div>

#### Don't exclude metadata for Docker

To work with a Docker repository, it must have its metadata intact. Therefore, when importing to/exporting from a Docker repository make sure that **Exclude Metadata** is not checked.



### Importing into a Remote Repository Cache

You can take advantage of remote repositories you have already downloaded to your local environment, and import them directly into a local repository.

For example, you can take your local Maven repository (usually located under `~/.m2`) and upload it into Artifactory so that all the artifacts you have already downloaded are now available on the server.

## Import Layout

An imported repository needs to be formatted using a Maven 2 repository layout.

When importing a single repository, the file structure within the import folder (or zip file) should be as follows:

```
IMPORT_FOLDER/ZIP_FILE
|
|--LIB_DIR_1
```

When importing all repositories, the file structure within the import folder should be as follows:

```
IMPORT_FOLDER/ZIP_FILE
|
|--REPOSITORY_NAME_DIR_1
| |
| |--LIB_DIR_1
```



When importing all repositories, you need to ensure that the names of the directories representing the repositories in the archive match the names of the target repositories in Artifactory.