Upgrading Artifactory

Overview

The procedure to upgrade Artifactory depends on your installation type. We strongly recommend reading through this page before proceeding with your upgrade. Detailed upgrade instructions are provided in dedicated pages for the following installation types:

- ZIP file
- RPM
- Debian
- Docker

In addition, within each installation type, there may be slight variation in instructions if you are upgrading from older versions of Artifactory.

Page Contents

Learn more

- Upgrading an Enterprise HA Cluster

Before You Proceed

Before you upgrade

We strongly recommend that you take the following actions to ensure you can roll back your system in case you encounter any issues during the upgrade process:

1. Do a complete System Export before commencing your upgrade procedure. If at any time you decide to roll back to your current version, you can use the export to reproduce your current system in its entirety.
2. Back up your database.
3. If you are using a relational database, make sure the database user defined in your *db.properties* file has permissions to modify the database schema

Before proceeding, there are a few points you need to address:

1. **JDK Version**
   From version 4.0, Artifactory requires JDK 8. If your current version is v3.x, before you upgrade to Artifactory 5.x or higher, please make sure you install JDK 8 and update your `JAVA_HOME` environment variable to point to your JDK 8 installation. For more details, please refer to System Requirements.

2. **Repositories with Multiple Package Types**
   From version 4.0, Artifactory will only index, and work with corresponding clients for single package type repositories. If your current version is 3.x and the installation includes repositories that support multiple package types, you need to migrate them to single package type repositories. You may do so before upgrading or after. For more details please refer to Single Package Type Repositories.

3. **'Slash' character encoding for NPM builds**
   Handling of 'slash' character encoding for NPM has been moved from the `artifactory.system.properties` file to the `catalina.properties` file of your Tomcat. For details, please refer to Npm Scope Packages.

Upgrading Artifactory Enterprise / HA

There are different instructions for upgrading Artifactory HA

When upgrading an HA cluster, the procedure for upgrading each node is similar to upgrading a single instance (Non-HA) installation, however, there are additional actions required for the nodes to operate as a high availability cluster.

If you are upgrading an Artifactory HA cluster, please refer to Upgrading an Enterprise HA Cluster.

Upgrading to the Latest Version

⚠ **server.xml file default changes from Artifactory 6.10**

From version 6.10, new Artifactory installations will not include the AJP connector in Artifactory's Tomcat server.xml file (i.e. under $ARTIFACTORY_HOME/tomcat/conf/server.xml).

To use AJP connector, the *<Connector port="8019" protocol="AJP/1.3" sendReasonPhrase="true"/>* should be added manually to the server.xml file under *<Service name="Catalina">*.

Upgraded Artifactory environments will be unchanged provided the *server.xml* file is copied over.

Notice: AJP connector support will be removed in Artifactory's next major version (7.0).

Upgrading from version 4.x or 5.x to the latest version is a simple procedure. Please refer to the sections below with specific instructions for your installation type.

Upgrading from any version prior to 6.14 to any version above 6.14 requires root privileges while running the upgrade process. Upgrading post version 6.14 does not require this.

ZIP Installation

1. Unzip the Artifactory distribution archive.
2. If you have not yet done so, perform a graceful shutdown of your currently running installation.
3. If the *$ARTIFACTORY_HOME/tomcat/conf/server.xml* has been modified keep it in a temporary location.
4. Backup files to a temporary location according to the conditions described below:
   a. In all cases, backup *$ARTIFACTORY_HOME/bin/artifactory.default*
   b. If Artifactory is configured to work with a database that is not Derby, backup the *$ARTIFACTORY_HOME/tomcat/lib/<JDBC>* driver.
5. Remove the following files and folders from your *$ARTIFACTORY_HOME* folder:
   - *webapps/artifactory.war*
   - *webapps/access.war (this will only be present if your current version is 5.4.0 and above due to addition of the Access Service)*
   - *tomcat*
   - *bin*
   - *misc*
6. Replace the removed files and folders with the corresponding ones from the new unzipped version.
7. Any files that were stored in temporary locations should now be returned to their original location under the new installation.

⚠ **Running Artifactory as a ROOT application in Tomcat**

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase=" webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading to version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
  <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

8. **If you installed Artifactory as a service, you now need to run the service**
   a. For a Linux service, browse to *`$ARTIFACTORY_HOME/bin`* and execute the following command **as root**: *`$ARTIFACTORY_HOME/bin/installService.sh [USER [GROUP]]`*
   b. For Windows service, browse to %ARTIFACTORY_HOME%\bin. First uninstall the service by running `uninstallService.bat`, then reinstall the service by running `installService.bat`. You can now run the service through the Windows UI or as described in Running Artifactory.

Debian Installation

1. Log in as root (or use `sudo su -`).
2. If you have not yet done so, perform a graceful shutdown of your currently running installation
3. Execute the following command:

```
dpkg -i $jfrog-artifactory-<oss|pro>-6.y.z.deb
```

Managing Configuration Files

When upgrading a **Debian** installation the upgrade process overwrites the following set of configuration files:

- *`system.properties`*
- *`config.xml`*
- *`default`*
- *`logback.xml`*
- *`mimetypes.xml`*
- All files under *`opt/jfrog/artifactory/misc`*
- All files under *`opt/jfrog/artifactory/webapps`*

If any of these files were modified, a backed up file will be created automatically with a notification in the upgrade log. If you need to restore the configuration changes you can restore them from the backup created.

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase="webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading to version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upg

```
  <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

Upgrading to version 6.14.0 and above

From version 6.14.0, the systemd runs successfully. Prior to upgrading to this version perform the following steps:

1. Stop Artifactory by running the following command.
2. Ignore any errors that may arise, and allow it to run to disable the auto start by systemd.

   ```
   systemctl stop artifactory.service
   ```

3. Kill any running Artifactory Java process that may still be running, using the following command:

   ```
   ps -ef | grep java | grep artifactory | grep tomcat
   ```

4. Rename the existing /etc/opt/jfrog/artifactory/default file to the new artifactory.pid file location:

   ```
   export ARTIFACTORY_PID=/var/run/artifactory.pid
   ```

5. Run the new Debian installation
6. Start Artifactory by running the following command:

   ```
   systemctl start artifactory.service
   ```

To ensure that all of the data remains in Artifactory following the upgrade to Pro (assuming you are using the default DB), please follow these steps:

1. Perform a graceful shutdown of your currently running installation.
2. Perform a backup of $ARTIFACTORY_HOME/data folder.

3. Delete the OSS installation.

   ```
   apt-get remove jfrog-artifactory-oss)
   ```

4. Download and install the Pro package.

   ```
   dpkg -i $jfrog-artifactory-pro-<version>.deb)
   ```

5. Import the ‘data’ backup folder to the new installation location.

   ```
   $ARTIFACTORY_HOME/data
   ```

6. Start Artifactory.

   ```
   $ARTIFACTORY_HOME/data
   ```

Docker Installation

In order to keep your data and configuration between versions, when upgrading the Artifactory Docker image, you need to use an external mounted volume as described under Managing Data Persistence.

To upgrade the Artifactory Docker image, follow these steps:

1. Stop the current container
2. Start a container with the new version using same data and configuration
3. Remove the old container

The example below shows this process for upgrading Artifactory from v5.0.0 to v5.1.0.

```
$ # Stop the currently running container
$ docker stop artifactory-5.0.0

$ # Start a new container from the new version image
$ docker run -d --name artifactory-5.1.0 --volumes-from=artifactory-5.0.0 -p 8081:8081 docker.bintray.io
/jfrog/artifactory-pro:5.1.0

$ # Remove old container
$ docker rm artifactory-5.0.0
```

Once these commands have completed successfully, you would have the new version (5.1.0 in the above example) running with the data and configuration from the old version that was removed.

⚠ **Running Artifactory as ROOT**

If you have configured Artifactory to run as the ROOT application in Tomcat, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase="webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading from version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
 <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

RPM Installation

Make sure you are upgrading from v3.6 or above

When running as an RPM installation, you can only upgrade to v5.x if your current version is 3.6 or above. If necessary, first upgrade your current version to 3.6, and then upgrade to v5.x .

If you try to upgrade a version below 3.6 using `rpm --force` you may end up deleting all of your data.

1. Download the **Artifactory Pro RPM Installer.** The latest version can be downloaded from the JFrog Artifactory Pro Download Page. Previous versions can be downloaded from JFrog Bintray.
2. Log in as root (or use `sudo su -`).
3. If you have not yet done so, perform a graceful shutdown of your currently running installation
4. Execute the following command:

```
rpm -U jfrog-artifactory-pro-6.y.z.rpm
```

Switching from Artifactory OSS to Pro

If you are just switching from Artifactory with the same version number, you need to append the command with --force --nodeps as follows:

```
rpm -U jfrog-artifactory-pro-6.y.z.rpm --force --nodeps
```

During an upgrade of an RPM installation different files may get backed up, where the backup file is appended with either a *.rpmorig* or a *.rpmnew* extension.

A *.rpmorig* extension means that the original file in your installation, the one that was there before performing the upgrade, was backed up before being replaced in the upgrade process.

A *.rpmnew* extension means that the original file in your installation, was **not** replaced in the upgrade, and instead, the new file with the same filename was backed up.

In either case, Artifactory will display a message such as:

```
warning: /etc/opt/jfrog/artifactory/default saved as /etc/opt/jfrog/artifactory/default.rpmorig
```

In these cases we recommend comparing the file installed once the upgrade has been completed with the backed-up file to see which best fits your needs, and using that one in the final setup.

If you make any changes, you may need to restart Artifactory for the change to be applied.

Upgrading Using YUM

An easy way to upgrade Artifactory from version 3.x or 4.x to the latest version is to use YUM with the Bintray Artifactory repository. The code snippets below show how to do this depending on whether your current version is below 3.6, or 3.6 and above.

Upgrading an Artifactory HA cluster?

If you are upgrading an Artifactory HA cluster, and you are running with a version that is older than version **5.4.6**, you should review the instructions on Upgrading an Enterprise HA Cluster prior to upgrading.

```
curl https://bintray.com/jfrog/artifactory-pro-rpms/rpm -o bintray-jfrog-artifactory-pro-rpms.repo &&
sudo mv bintray-jfrog-artifactory-pro-rpms.repo /etc/yum.repos.d
yum install jfrog-artifactory-pro-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-pro
```

```
curl https://bintray.com/jfrog/artifactory-pro-rpms/rpm -o bintray-jfrog-artifactory-pro-rpms.repo &&
sudo mv bintray-jfrog-artifactory-pro-rpms.repo /etc/yum.repos.d
yum upgrade artifactory
yum install jfrog-artifactory-pro-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-pro
```

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase="webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading to version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
 <Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

Upgrading to version 6.14.0 and above

From version 6.14.0, the systemd runs successfully. Prior to upgrading to this version perform the following steps:

1. Stop Artifactory by running the following command.
2. Ignore any errors that may arise, and allow it to run to disable the auto start by systemd.

   ```
   systemctl stop artifactory.service
   ```

3. Kill any running Artifactory Java process that may still be running, using the following command:

   ```
   ps -ef | grep java | grep artifactory | grep tomcat
   ```

4. Rename the existing /etc/opt/jfrog/artifactory/default file to the new artifactory.pid file location:

   ```
   export ARTIFACTORY_PID=/var/run/artifactory.pid
   ```

5. Run the new RPM installation
6. Start Artifactory by running the following command:

   ```
   systemctl start artifactory.service
   ```

RPM OSS Installation

1. Download the **Artifactory OSS RPM Installer.** The latest version can be downloaded from the JFrog Open Source page. Previous versions can be downloaded from JFrog Bintray.
2. Log in as root (or use `sudo su -`).
3. If you have not yet done so, perform a graceful shutdown of your currently running installation
4. Execute the following command:

   ```
   rpm -U jfrog-artifactory-oss-6.y.z.rpm
   ```

During an upgrade of an RPM installation different files may get backed up, where the backup file is appended with either a *.rpmorig* or a *.rpmnew* extension.

A *.rpmorig* extension means that the original file in your installation, the one that was there before performing the upgrade, was backed up before being replaced in the upgrade process.

A *.rpmnew* extension means that the original file in your installation, was **not** replaced in the upgrade, and instead, the new file with the same filename was backed up.

In either case, Artifactory will display a message such as:

```
warning: /etc/opt/jfrog/artifactory/default saved as /etc/opt/jfrog/artifactory/default.rpmorig
```

In these cases we recommend comparing the file installed once the upgrade has been completed with the backed-up file to see which best fits your needs, and using that one in the final setup.

If you make any changes, you may need to restart Artifactory for the change to be applied.

Upgrading Using YUM

An easy way to upgrade Artifactory from version 3.x or 4.x to the latest version is to use YUM with the Bintray Artifactory repository. The code snippets below show how to do this depending on whether your current version is below 3.6, or 3.6 and above.

```
curl https://bintray.com/jfrog/artifactory-rpms/rpm -o bintray-jfrog-artifactory-rpms.repo && sudo mv
bintray-jfrog-artifactory-rpms.repo /etc/yum.repos.d/
yum install jfrog-artifactory-oss-6.x.y
```

✓ To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-oss
```

```
curl https://bintray.com/jfrog/artifactory-rpms/rpm -o bintray-jfrog-artifactory-rpms.repo && sudo mv
bintray-jfrog-artifactory-rpms.repo /etc/yum.repos.d/
yum upgrade artifactory
yum install jfrog-artifactory-oss-6.y.z
```

To install the latest version of Artifactory, you can run:

```
yum install jfrog-artifactory-oss
```

Running Artifactory as ROOT

If you have configured Artifactory to run as the ROOT application in Tomcat, before you proceed, you need to follow the steps described in **this Knowledge Base article**.

Version-specific Special Instructions

To accommodate internal changes in Artifactory in different version upgrades, you need to perform or ensure certain changes to Artifactory's server.xml according to your target upgrade version. Note that the changes listed below are cumulative as the versions progress.

Upgrading to version 5.4.0 and above

From version 5.4.0, Artifactory's management of Access Tokens was moved to a separate service, JFrog Access, installed as a separate web application under the same Tomcat as Artifactory. This requires your Tomcat's server.xml to be configured to allow running 2 processes. If you are using a server.xml file from a previous installation, when returning it, make sure it is configured to allow **2 start/stop threads** as shown below (see <Host name="localhost" appBase="webapps" startStopThreads="2"/>):

```
...
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="webapps" startStopThreads="2"/>
        </Engine>
...
```

Upgrading to version 5.6.1 and above

In version 5.6.1, the version of the Tomcat embedded in Artifactory was upgraded. To accommodate YUM clients following this upgrade, a new attribute, `sendReasonPhrase` was added for **each** Tomcat connector in the server.xml file as shown in the example below:

```
<Connector port="8081" sendReasonPhrase="true"/>
```

Upgrading to version 5.7.0 and above

From version 5.7.0, Artifactory routes requests to its Access service directly through a port, and this requires the following modification to the server.xml file

```
<Connector port="8040" sendReasonPhrase="true" maxThreads="50"/>
```

Upgrading from version 6.3 and above

From version 6.3, the version of the Tomcat embedded in Artifactory was upgraded. To accomodate special characters following this upgrade, the following new attributes, **relaxedPathChars='[]' relaxedQueryChars='[]'** were added for each Tomcat connector in the server.xml as shown in the example below:

```
<Service name="Catalina">
        <Connector port="8081" sendReasonPhrase="true" relaxedPathChars='[]'
relaxedQueryChars='[]'/>
```

Using SHA256 Checksums

From version 5.5, Artifactory natively supports SHA-256. New artifacts that are uploaded will automatically have their SHA-256 checksum calculated, however, artifacts that were already hosted in Artifactory prior to the upgrade will not have their SHA-256 checksum in the database yet.

To make full use of Artifactory's SHA-256 capabilities after upgrading to version 5.5 and above, you need to run a process that migrates Artifactory's database making sure that the record for each artifact includes its SHA-256 checksum. For full details on Artifactory's SHA-256 support and instructions on how to migrate your database, please refer to **SHA-256 Support**.

Upgrading from OSS to Pro

Even if you're just switching your current version of Artifactory OSS to the same version of Artifactory Pro, please follow the instructions under Upgrading to the Latest Version according to your installation type (ZIP, RPM, Debian or Docker).

⚠ **Add your license**

After upgrading to the latest version, make sure you provide your Artifactory installation with a license. Simply create a file called *`artifactory.lic,`* copy the license you have received from JFrog and paste it into the *`artifactory.lic`* license file, and place it in the *`$ARTIFACTORY_HOME/etc`* folder.

Upgrading to Version 6.6 and Above

This section is only relevant for users using Build Integration and Build Info.

During the 6.6 upgrade, there will be a migration of all existing build info json files from the DB to a default ***artifactory-build-info*** repository in Artifactory.

This default ***artifactory-build-info*** repository will store all build info files uploaded to Artifactory by the different CI server plugins, such as the Artifactory Jenkins Plugin, CLI, and directly through the Build Upload REST API or Artifactory UI.

Additional information about the migration process can be found here.

Upgrading from Version 3.x

Single Package Type Repositories

⚠ **Single Package Type**

To work with version 4.x and above, you need to ensure that your repositories only contain artifacts with the same package type. A script to check this can be found on the JFrog GitHub.

In version 3.x Artifactory supported repositories with multiple package types. You were able to upload packages with different types to the same repository and Artifactory would calculate the metadata for those packages. Nevertheless, maintaining a single package type per repository was always a best practice that optimized performance and produced a more organized repository structure in your system. From version 4.0, you need to specify a single **Package Type** for a repository when you create it. Artifactory will only calculate metadata for, and be recognized by the corresponding client software for artifacts of the **Package Type** specified for that repository. (Artifactory will not prevent you from uploading packages of a different type, however, it will not calculate metadata for those packages, and the client for the different package types will not recognize the repository).

If you currently have repositories that are configured to support multiple package types, you need to migrate them to single package type repositories, however, you may do so either before or after running the upgrade procedure.

To migrate your repositories before upgrading, please refer to Migrating to Single Package Type Repositories.

If you prefer to migrate your repositories after upgrading, or have already upgraded, please refer to Fixing Multiple Package Type Repositories.

✓ **Generic repositories**

In version 4.x and 5.x, if you need a repository to hold packages of several different types, you may specify its package type to be **Generic.** Artifactory does not calculate metadata for Generic repositories, and effectively, they behave like a simple file system to store packages.

Migrating to Single Package Type Repositories

To migrate a repository with multiple package types to single package type repositories, execute the following steps:

1. Change the configuration of the original repository so it supports only one package type.
2. For each additional packaging type needed, create a new repository with the corresponding package type

3. Use the REST API or the UI to move packages from the original repository to the new one(s) created until all repositories only contain packages of the same type.
   When using the REST API, make sure to include the `suppressLayouts=1` query parameter in order to prevent artifact path transformations.

> **Npm Repositories**
>
> If you move data to an Npm repository, make sure to include the *`.npm`* folder. This will preserve extra information that may have been stored when deploying packages using the npm client.

Fixing Multiple Package Type Repositories

If you upgraded without migrating to single package type repositories, then Artifactory will start normally, however, repositories containing multiple package types will be **randomly** assigned one of the single package types from the original repository and output corresponding messages to the *`$ARTIFACTORY_HOME/logs/artifactory.log`* file.

For example, if libs-release-local contained three different package types: RubyGems, Npm and NuGet, after upgrading, your *$ARTIFACTORY_HOME/logs/artifactory.log* may contain messages similar to the ones below:

```
2015-06-28 10:10:47,656 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:42) Converting repositories to a single package type
2015-06-28 10:10:47,663 [art-init] [ERROR] (o.a.v.c.v.SingleRepoTypeConverter:155) Disabling package 'Gems' for repo 'libs-release-local' since only one packaging type is allowed!
2015-06-28 10:10:47,664 [art-init] [ERROR] (o.a.v.c.v.SingleRepoTypeConverter:155) Disabling package 'Npm' for repo 'libs-release-local' since only one packaging type is allowed!
2015-06-28 10:10:47,664 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'libs-release-local' to type NuGet
2015-06-28 10:10:47,664 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'libs-snapshot-local' to type Maven
2015-06-28 10:10:47,664 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'plugins-release-local' to type Maven
2015-06-28 10:10:47,664 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'plugins-snapshot-local' to type Maven
2015-06-28 10:10:47,665 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'ext-release-local' to type Maven
2015-06-28 10:10:47,665 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'ext-snapshot-local' to type Maven
2015-06-28 10:10:47,666 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'jcenter' to type Maven
2015-06-28 10:10:47,666 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'remote-repo' to type Maven
2015-06-28 10:10:47,668 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'libs-snapshot' to type Maven
2015-06-28 10:10:47,668 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:128) Setting repository 'p2' to type P2
2015-06-28 10:10:47,668 [art-init] [INFO ] (o.a.v.c.v.SingleRepoTypeConverter:56) Finished Converting repositories to a single package type
```

In this example, Artifactory set the **Package Type** to NuGet.

To fix this condition, you can simply follow steps described above in Migrating to Single Package Type Repositories, or after you upgrade, use the packageType utility found on the JFrog Github for 4.x migrations.

Upgrading from Any Version Below v3.0

To upgrade from a version prior to 3.0, you first need to upgrade to version 3.9.x as described in Upgrading Artifactory in the Artifactory 3 documentation.

> **Interim versions**
>
> Depending on your current version, upgrading to version 3.9.x may require you to first upgrade to an interim version.

Downgrading Artifactory

The procedure to downgrade Artifactory may vary depending on the version you are using. For more details, please contact JFrog Support.