

Manipulating Artifacts

Overview

Artifactory supports move, copy and deletion of artifacts to keep your repositories consistent and coherent. When an artifact is moved, copied or deleted, Artifactory immediately and automatically updates the corresponding metadata descriptors (such as *maven-metadata.xml*, RubyGems, Npm and more) to reflect the change and keep your repositories consistent with the package clients.

In addition, as a convenience feature, Artifactory provides a simple way to do a complete [version cleanup](#).

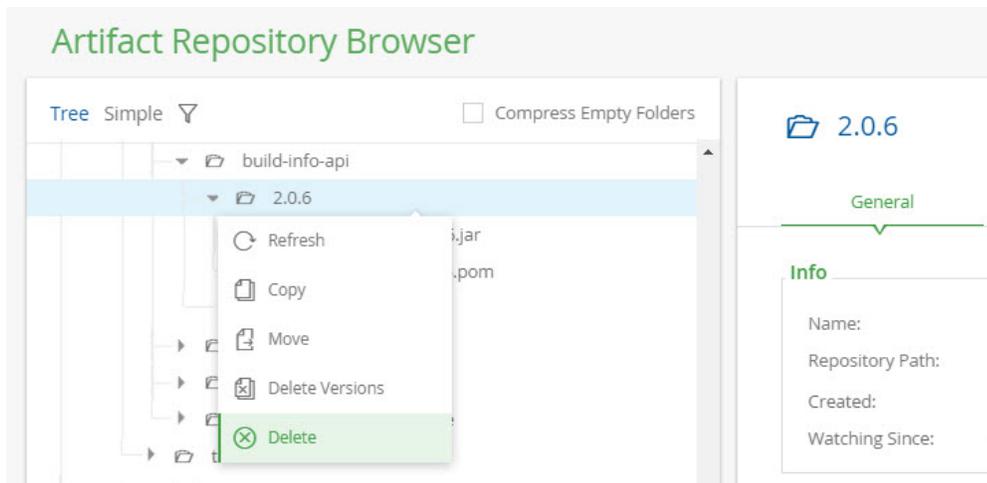
Page Contents

- [Overview](#)
- [Deleting a Single Item](#)
- [Deleting a Version](#)
- [Moving and Copying Artifacts](#)
 - [Simulating a Move or Copy](#)
- [Downloading a Folder](#)
 - [Configuring Folder Download](#)
 - [Blocking Folder Download](#)

Deleting a Single Item

To delete a single artifact or directory, select the item in the Tree Browser, and click **Delete** from the **Actions** menu or the right-click menu.

Once the item is deleted, the corresponding metadata file is updated to reflect the change so the item will not be found in a search.

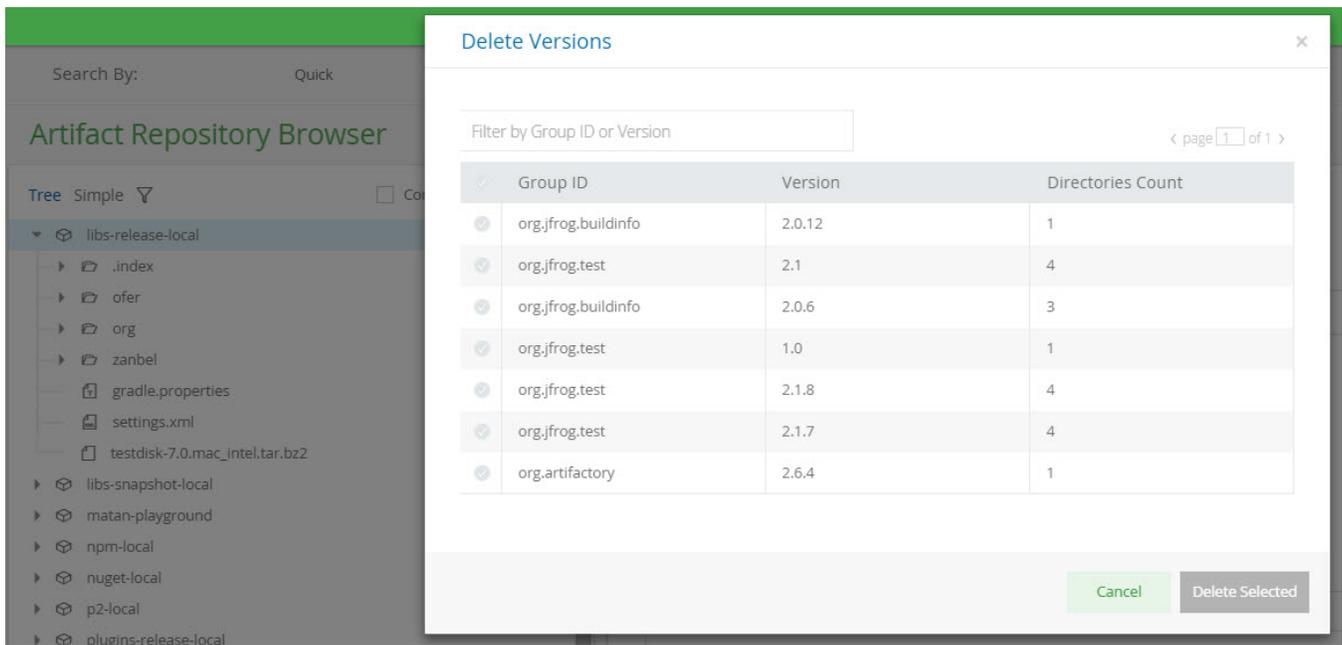


Deleting a Version

It is common for a repository to accumulate many different artifacts deployed under the same group (or path prefix) and the same version. This is especially true for snapshot deployments of multi-module projects, where all deployed artifacts use the same version. To delete a version by individually deleting its constituent artifacts can be tedious, and would normally be managed by writing a dedicated script. Artifactory lets you select one of the artifacts in a version and then delete all artifacts with the same version tag in a single click while keeping the corresponding metadata descriptors up to date.

To delete a version, right-click a folder in the [Tree Browser](#) and select **Delete Versions...**

Artifactory drills down into the selected folder and returns with a list of all the groups and versions that can be deleted.



Select the versions you want to clean up and click **Delete Selected**

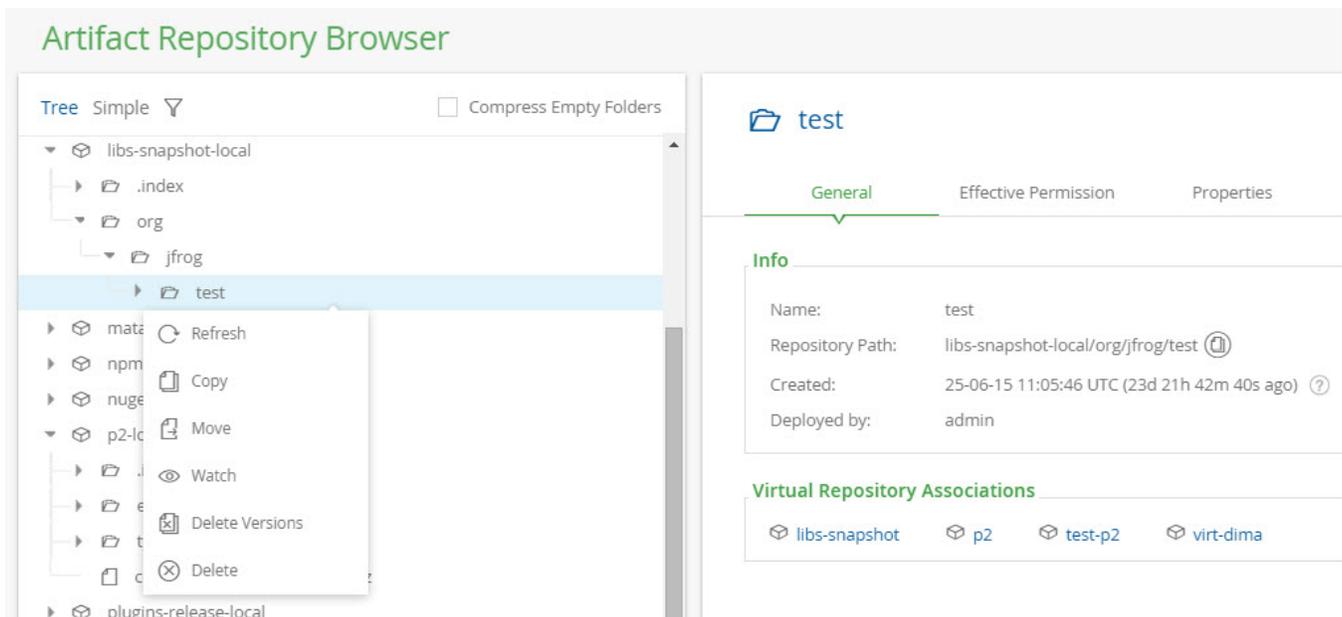


Limit to number of versions displayed

To avoid an excessively long search, Artifactory only displays the different version numbers assigned to the first 1000 artifacts found in the selected level of the repository hierarchy. If you do not see the version number you wish to delete, filter the artifacts displayed in the **Delete Versions** dialog by Group ID or Version number.

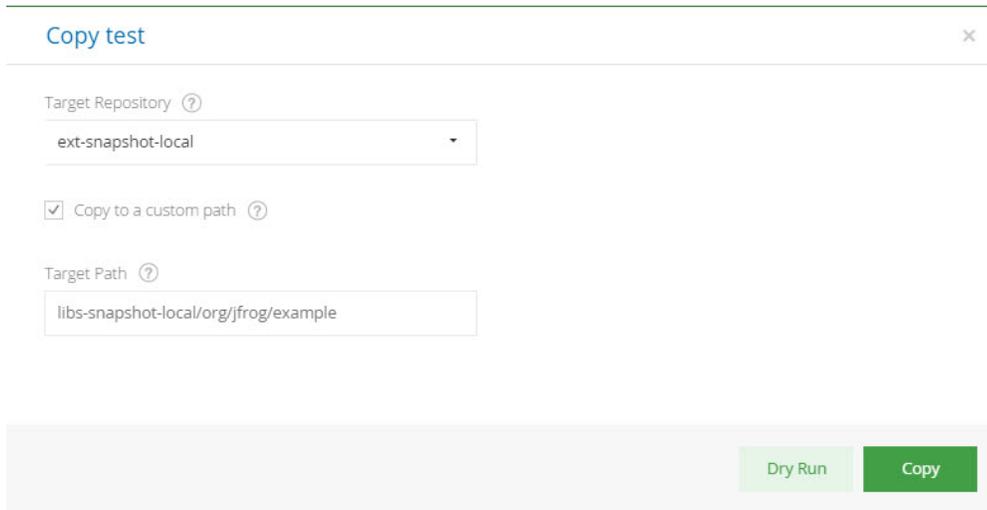
Moving and Copying Artifacts

To move or copy an artifact or folder, select it in the Tree Browser and then click **Move...** or **Copy...** from the **Actions** menu or from the right-click menu.



Artifactory will display a list of repositories from which you need to select your **Target Repository** for the operation.

The list of target repositories will contain all the local repositories, or virtual repositories with a "Default Deployment Repository" configured that are configured with the same package type as the source repository, or configured as generic repositories. This means, for example, you can only move an artifact from a debian repository to a generic repository or to a local debian repository.



Copy test

Target Repository [?]
ext-snapshot-local

Copy to a custom path [?]

Target Path [?]
libs-snapshot-local/org/jfrog/example

Dry Run Copy

After selecting your **Target Repository**, you can specify a custom **Target Path** if you want your source artifacts moved to a different location the **Target Repository**.

 **Copy operations are executed using Unix conventions**

Copy operations are executed using Unix conventions. For example, copying *org/jfrog/1* from a source repository to *org/jfrog/1* in a target repository will result in the contents of the source being copied to *org/jfrog/1/1*. To achieve the same path in the target repository, copy the source into one folder up in the hierarchy. In the above example, that would mean copying source *org/jfrog/1* into target *org/jfrog*. If you leave the Target Path empty, the source will be copied into the target repository's root folder.

 **Custom target path suppresses cross-layout translation**

If you are copying or moving your source artifacts to a repository with a different layout, specifying a **Custom Target Path** suppresses cross-layout translation. This means that your client may NOT be able to resolve the artifacts, even in cases of a same-layout operation.

Once you have selected your **Target Repository** (and **Custom Target Path** if needed), click **Move...** or **Copy...** to complete the operation.

All metadata is updated to reflect the operation once completed.

Simulating a Move or Copy

Note that an operation may fail or generate warnings for several reasons. For example, the target repository may not accept all the items due to its own specific policies, or you may not have the required permissions to perform the operation.

Before actually doing an operation, we recommend that you check if it will succeed without errors or warnings by clicking **Dry Run**.

Artifactory will run a simulation and display any errors and warnings that would appear when doing the actual operation.

Copy test



Target Repository [?](#)

ext-snapshot-local

Copy to a custom path [?](#)

Target Path [?](#)

libs-snapshot-local/org/jfrog/example

Dry Run

Copy

Failed to validate target path of pom: libs-snapshot-local/org/jfrog/example/multi/2.18-SNAPSHOT/multi-2.18-20150625.110536-1.pom
Failed to validate target path of pom: libs-snapshot-local/org/jfrog/example/multi/2.18-SNAPSHOT/multi-2.18-

Permissions required

To successfully complete a move, you need to have DELETE permission on your source repository, and DEPLOY permission on your target repository

Downloading a Folder

Artifactory allows the download of a complete folder that is selected in the [Tree Browser](#) or [Simple Browser](#).

This ability is configurable by an Artifactory administrator, and if allowed, when a folder is selected the **Download** function is available in the **Actions** menu.

Actions

 Refresh

 Copy

 Move

 Delete Versions

 Delete

 Download

Download Folder
✕

Folder Name: org/jfrog
Files Count: 1
Total Size: 0,44 MB

Archive Type

Include Checksum files

Download

Archive Type	The Archive Type. Currently, zip, tar, tar.gz and tgz are supported.
Include Checksum Files	<p>Include SHA1, SHA256 and MD5 files - In Artifactory, checksum files (.sha1, .sha256 and .md5 files) are displayed and are downloadable in the HTML browsing endpoint (for example, http://<ARTIFACTORY>/artifactory/<REPOSITORY_KEY>), depending on one of the below pre-conditions:</p> <ol style="list-style-type: none"> 1.The artifact was originally uploaded with its checksum value (i.e the deploying client provided a checksum header such as the "X-Checksum-Sha1" header on the request). 2.The repository Checksum Policy is set to "Trust Server Generated Checksums". <p>If the latter applies, there is no need to provide the artifact checksums during the upload in order for its checksum files to be visible.</p> <p>The Download Folder functionality mimics this mechanism, and will write checksum files to the output archive based on the same conditions.</p> <p>*With remote repository caches, there is no distinction for the checksum policy of the repository. Simply checking this checkbox will always result in checksum files being added.</p>

You can also download a folder using the [Rest API](#).

Configuring Folder Download

An Artifactory administrator can enable complete folder download in the **Admin** module under **Admin | General Configuration**. This configuration will apply to all Artifactory users.

Folder Download Settings

Enable Folder Download

Enable Folder Download For Anonymous Access

Max Size [?](#)

1024

Max Number of Files [?](#)

5000

Max Parallel Folder Downloads [?](#)

10

Max Size	The maximum size (MB) of artifacts that can be downloaded in a folder download.
Max Number of Files	The maximum number of artifacts that may be downloaded from the selected folder and all its sub-folders.
Max Parallel Folder Downloads	The maximum number of concurrent folder downloads allowed.

Blocking Folder Download

From version 5.10, when trying to download a folder, if any of the artifacts in that folder are blocked for download, then downloading the folder will fail with an HTTP FORBIDDEN (403) error.