

Repository Replication

Overview

Through the Replication Add-on in Artifactory Pro, Artifactory allows replication of repositories between two Artifactory instances to support development by different teams distributed over distant geographical sites. The benefits of replication are:

- Ensuring developers all work with the same version of remote artifacts
- Ensuring build artifacts are shared efficiently between the different development teams
- Overcome connectivity issues such as network latency and stability when accessing remote artifacts
- Accessing specific versions of remote artifacts



Artifactory versions for replication

We strongly recommend that replication is only performed between servers running the same version of Artifactory Pro.

Two main methods of replication are supported:

- [Push replication](#)
- [Pull replication](#)

Push Replication

Push replication is used to synchronize [Local Repositories](#), and is implemented by the Artifactory server on the near end invoking a synchronization of artifacts to the far end.

There are two ways to invoke push replication:

- **Scheduled push:** Pushes are scheduled asynchronously at regular intervals
- **Event-based push:** Pushes occur in nearly in real-time since each create, copy, move or delete of an artifact is immediately propagated to the far end.

Advantages

- It is fast because it is asynchronous.
- It minimizes the time that repositories are not synchronized.
- It reduces traffic on the master node in case of a replication chain ("Server A" replicates to "Server B", "Server B" then replicates to "Server C" etc.).



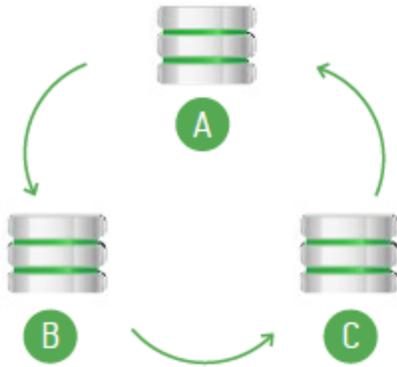
Avoid Replication Loops ("Cyclic Replication")

A replication loop occurs ("Cyclic" or "Bi-directional" replication) occurs when two instances of Artifactory running on different servers are replicating content from one to the other concurrently.

For example, "Server A" is configured to replicate its repositories to "Server B", while at the same time, "Server B" is configured to replicate its repositories to "Server A".

Or "Server A" replicates to "Server B" which replicates to "Server C" which replicates back to "Server A".

We strongly recommend avoiding cyclic replication since this can have disastrous effects on your system causing loss of data, or conversely, exponential growth of disk-space usage.



Replication loop to be strictly avoided



When to Use Push Replication

Event-based push replication is recommended when it is important for the repository at the far end to be updated in near-real-time for any change (create, copy, move or delete of an artifact) in the repository at the near end.

In addition, we recommend running regular scheduled replication on top of event-based replication to guarantee full copy consistency even in cases of server downtime and network partitions.

Multi-push Replication



Requires an **Enterprise license**

With an Enterprise license, Artifactory supports multi-push replication allowing you to replicate a local repository from a single source to multiple enterprise target sites simultaneously.

Pull Replication

Pull replication is invoked by a remote repository, and runs according to a defined schedule to synchronize repositories (local, remote or virtual) at regular intervals.

This provides a convenient way to proactively populate a remote cache, and is very useful when waiting for new artifacts to arrive on demand (when first requested) is not desirable due to network latency.

Pull replication runs as a scheduled task.

Advantages

- Many target servers can pull from the same source server efficiently implementing a one-to-many replication.
- It is safer since each package only has one "hop".
- It reduces traffic on target servers since they do not have to pass on artifacts in a replication chain.



When and when not to Use Pull Replication

Pull replication is recommended in the following cases:

- When you need to replicate a repository to many targets.
- When your source repository is located behind a proxy that prevents push replication (e.g. replicating a repository hosted on Artifactory SaaS to a local repository at your site)

Pull replication cannot be used to replicate a remote resource that is not an Artifactory repository. Artifacts from third party repositories can only be cached on-demand in the normal cache and proxy behavior of a [remote repository](#).

Page Contents

- [Overview](#)
 - [Push Replication](#)
 - [Advantages](#)
 - [Multi-push Replication](#)
 - [Pull Replication](#)
 - [Advantages](#)
- [Scheduling and Configuring Replication](#)
 - [Using the UI](#)
 - [Configuring Push Replication](#)
 - [Adding a push replication target](#)
 - [Configuring Pull Replication](#)
 - [Replicating with REST API](#)
 - [Replication Properties](#)
- [Watch the Screencast](#)

Scheduling and Configuring Replication

Using the UI

Replication is configured via the user interface as a scheduled task. Local repositories can be configured for push replication, and remote repositories can be configured for pull replication.

All replication messages are logged in the main [Artifactory log file](#) (*artifactory.log*).

The **Replications** column in your list of local repositories indicates if replication is configured for each repository in the list. If replication is indeed configured for a repository, you can click the icon in the list to invoke it.



Repository Key	Type	Recalculate Index	Replications
ext-release-local	Maven	⌂	▶
ext-snapshot-local	Maven	⌂	▶
libs-release-local	Maven	⌂	▶
libs-snapshot-local	Maven	⌂	▶
plugins-release-local	Maven	⌂	▶
plugins-snapshot-local	Maven	⌂	▶

Configuring Push Replication

A push replication task for a Local Repository is configured in the **Replication** tab of the [Edit Local Repository](#) dialog.

First, in the **Cron Expression** field define the replication task schedule using a valid [cron](#) expression.

The **Next Replication Time** will indicate update accordingly.



Cron Expression VS Event Base Replication

Replication of this repository to all of its targets occurs simultaneously according to the **Cron Expression** you define.

The event base replication will attempt to replicate **only** the artifacts affected by the event while the Cron Expression will trigger a sync of all artifacts in repository. This difference is important since in case one of the event sync has failed the next time the Cron Expression will trigger a sync all changed will be synced.

Once you have configured the replication properties for each of your replication targets, the **Replication** tab for your repository displays them.

Edit libs-release-local

Basic > Advanced > **Replications**

Cron Expression * Next Replication Time Wed Jul 15 14:00:00 UTC 2015

Enable Event Replication

Replication

New < page 1 of 1 >

Url	Sync Deletes	Sync Properties	Enabled
http://10.10.10.10:8080/artifactory/lib-release-local	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
http://10.10.10.10:8080/artifactory/lib-release-local	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Cancel < Back Next > **Save & Finish**

Field Name	Description
Push to	The replication targets you have defined
Enabled	When set, enables replication of this repository to the target specified in Push to
Enable Event Replication	When set, each event will trigger replication of the artifacts changed in this event. This can be any type of event on artifact, e. g. add, deleted or property change.



Number of replication targets

If you do not have an Enterprise license, you may only define **one** replication target. With an Enterprise license, Artifactory supports multi-push replication and you may define as many targets as you need.

Adding a push replication target

To add a target site for this replication, click **Add** to display the **Replication Properties** dialog, and fill in the details as follows.

New Replication ✕

Enabled

URL *

Username * Password

Network Proxy Reference Socket Timeout *

Sync Deleted Artifacts Sync Artifact Properties

Sync Artifact Statistics

Path Prefix ?

Field Name	Description
Enable Active Replication of this Repository	When set, this replication will be enabled when saved
URL	The URL of the target local repository on a remote Artifactory server. For some package types, you need to prefix the repository key in the URL with api /<pkg>. For a list of package types where this is required, see the note below under Configuring Pull Replication .
Username	The HTTP authentication username.

Password	The HTTP authentication password.
Proxy	A proxy configuration to use when communicating with the remote instance.
Socket Timeout	The network timeout in milliseconds to use for remote operations.
Sync Deleted Artifacts	When set, items that were deleted locally should also be deleted remotely (also applies to properties metadata).
Sync Artifact Properties	When set, the task also synchronizes the properties of replicated artifacts.
Sync Artifact Statistics	When set, the task also synchronizes artifact download statistics. Set to avoid inadvertent cleanup at the target instance when setting up replication for disaster recovery.
Path Prefix (optional)	Only artifacts that located in path that matches the subpath within the repository will be replicated.

Configuring Pull Replication

A pull replication task for a Remote Repository is configured in the **Replication** tab of the [Edit Remote Repository](#) dialog.

First, in the **Cron Expression** field define the replication task schedule using a valid [cron](#) expression.

The **Next Replication Time** will indicate update accordingly.

Edit gradle-libs

Basic
Advanced
Replications

Enable Active Replication of this Repository

Cron Expression*

Next Replication Time

Wed Jul 15 14:00:00 UTC 2015

Sync Deletes

Sync Properties

Path Prefix

Cancel
< Back
Next >

Field Name	Description
------------	-------------

Enable Active Replication of this Repository	When set, this replication will be enabled when saved
URL	The URL of the target local repository on a remote Artifactory server. For some package types, you need to prefix the repository key in the URL with api/<pkg> . For a list of package types where this is required, see the note below.
Sync Deleted Artifacts	When set, items that were deleted locally should also be deleted remotely (also applies to properties metadata).
Sync Artifact Properties	When set, the task also synchronizes the properties of replicated artifacts.
Path Prefix (optional)	Only artifacts that located in path that matches the subpath within the remote repository will be replicated.

i **Regarding credentials of the remote repository configuration**

The remote repository's file listing for replication is retrieved using the repository's credentials defined under the repository's [Advanced](#) configuration section.

The remote files retrieved depend on the effective permissions of the configured user on the remote repository (on the other Artifactory instance).

! * **Don't prefix the repository path with api/<pkg>, but watch out for exceptions**

For some packaging formats, when using the corresponding client to access a repository through Artifactory, the repository key in the URL needs to be prefixed with **api/<pkg>** in the path. For example, in the case of [Npm](#) repositories, the repository key should be prefixed with `api/` `/npm`. Nevertheless, there are exceptions to this rule. For example, when replicating Maven repositories, you do **not** need to add a prefix the remote repository path.

The considerations of whether to prefix the repository key with **api/<pkg>** or not are the same as those when configuring smart remote repositories. For a detailed list of package formats that should be prefixed with **api/<pkg>**, please refer to [Configuration](#) under [Smart Remote Repositories](#).

Replicating with REST API

Both Push and Pull Replication are supported by Artifactory's REST API. For details please refer to the following:

- [Get Repository Replication Configuration](#)
- [Set Repository Replication Configuration](#)
- [Update Repository Replication Configuration](#)
- [Delete Repository Replication Configuration](#)
- [Scheduled Replication Status](#)
- [Pull/Push Replication](#)

Replication Properties

Once replication has been invoked, Artifactory annotates the source repository being replicated and annotates it with properties that indicate the status of the replication. These can be viewed, along with other properties that may annotate the repository, in the **Properties** tab of the [Tree Browser](#).

For single push replication operations, the following properties are created/updated:

Key	Value
artifactory.replication.<source_repo_key>.started	Indicates when the replication started

artifactory.replication.<source_repo_key>.status	Indicates the status of the replication operation once complete. It can take the following values: ok: The replication succeeded failure: The replication failed. You should check the log files for errors
artifactory.replication.<source_repo_key>.finished	Indicates when the replication finished

Filter by Property Delete Delete Recursively < page 1 of 1 >

Property	Value(s)
artifactory.replication.libs-snapshot-local.started	1437483921588
artifactory.replication.libs-snapshot-local.finished	1437484260530
artifactory.replication.libs-snapshot-local.result	ok

For multi-push replication operations (available to Enterprise customers only), the following properties are created/updated:

Key	Value
artifactory.replication.<source_repo_key>_<target_repo_URL>.started	Indicates when the replication started
artifactory.replication.<source_repo_key>_<target_repo_URL>.status	Indicates the status of the replication operation once complete. It can take the following values: ok: The replication succeeded failure: The replication failed. You should check the log files for errors
artifactory.replication.<source_repo_key>_<target_repo_URL>.finished	Indicates when the replication finished

Watch the Screencast

To see replication in action, watch the short screencast below.