

Jenkins Artifactory Plug-in

Overview

The popular Jenkins Artifactory Plug-in brings [Artifactory's Build Integration](#) support to Jenkins.

This integration allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them. The plug-in includes a vast collection of features, including a rich pipeline API library and [release management for Maven and Gradle builds](#) with Staging and Promotion.

The plug-in currently requires version **2.159** or above of Jenkins.

The Jenkins Artifactory Plug-in supports the following Jenkins Build Jobs:

- [Pipeline](#)
- [Freestyle](#)
 - [Maven](#)
 - [Gradle](#)
 - [Ivy-Ant](#)
 - [Generic](#)
- [Maven](#)
- [Ivy-Ant](#)

Get started with [configuring the Jenkins Artifactory Plug-in](#).



[Learn More](#)

[Jenkins Cheat Sheet for Jenkins Pipelines and Continuous Integration](#)

Integration with JFrog Pipelines

General

JFrog Pipelines integration with Jenkins is supported since version 1.6 of JFrog Pipelines and version 3.7.0 of the Jenkins Artifactory Plug-in. This integration allows triggering a Jenkins job from JFrog Pipelines. The Jenkins job is triggered using JFrog Pipeline's native Jenkins step. When the Jenkins job finishes, it reports the status back to JFrog Pipelines.

The integration supports:

1. Passing build parameters to the Jenkins job.
2. Sending data from Jenkins back to JFrog Pipelines.

Setting Up the Integration

- Open the JFrog Pipelines UI.
- Under **Integrations**, click on the **Add an Integration** button.
- Choose **Jenkins Server** integration type and fill out all of the required fields.
- Click on the **Generate** button to generate a token. This token is used by Jenkins to authenticate with JFrog Pipelines.

Page Contents

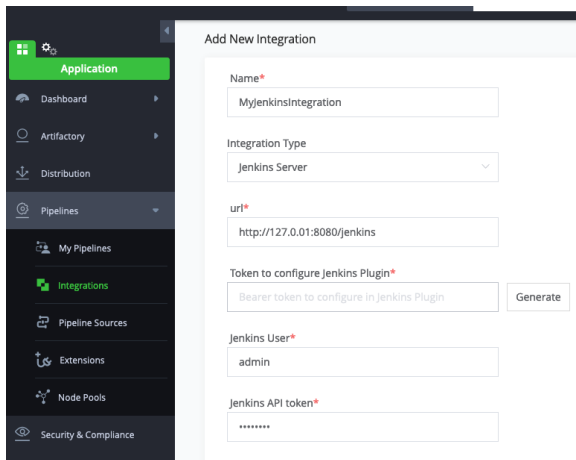
- [Overview](#)
- [Integration with JFrog Pipelines](#)
- [Multi-Configuration \(Freestyle\) Projects](#)
- [Triggering Builds](#)
- [JIRA Integration](#)
- [Build Isolation](#)
- [Excluded artifacts and the BuildInfo](#)
- [Discarding Old Builds](#)
- [Configuring repositories with variables](#)
- [Dynamically Disabling Deployment of Artifacts and Build-info](#)
- [Using the Jenkins Job DSL Plugin](#)
- [Watch the Screencast](#)
- [Release Notes](#)

Read more

- [Configuring Jenkins Artifactory Plug-in](#)
- [Working With Pipeline Jobs in Jenkins](#)
- [Maven Builds](#)
- [Gradle Builds](#)
- [Ivy-Ant Builds](#)
- [Generic Builds](#)
- [Jenkins Artifactory Plug-in - Release Management](#)
- [Setting Up Docker Build-info](#)

Integration Benefits

[JFrog Artifactory and Jenkins CI](#)



- Copy the Callback URL and Token, and save them in **Jenkins | Manage | Configure System | JFrog Pipelines server**

Triggering a Jenkins Job from JFrog Pipelines

To trigger a Jenkins job from JFrog Pipelines, add the Jenkins step in your pipelines yml as shown here.

```
- Name: MyJenkinsStep
  Type: Jenkins
  configuration:
    jenkinsJobName: <jenkins-job-name>
    Integrations:
      - name: MyJenkinsIntegration
```

Once the Jenkins job finishes, it will report back the status to the Jenkins step.

More Options

For pipelines jobs in Jenkins, you also have the option of sending info from Jenkins back to JFrog Pipelines. This info will be received by JFrog Pipelines as output resources of the Jenkins step in JFrog Pipelines. Here's how you add those resources to the Jenkins pipeline script:

```
jfPipelines (
  outputResources: """[
    {
      "name": "pipelinesBuildInfo",
      "content": {
        "buildName": "${env.JOB_NAME}",
        "buildNumber": "${env.BUILD_NUMBER}"
      }
    }
  ]"""
)
```

By default, the Jenkins job status and the optional output resources are sent by Jenkins to JFrog Pipelines when the job finishes. You also however have the option of sending the status and output resources before the Jenkins job ends. Here's how you do it:

```
jfPipelines (
  reportStatus: "SUCCESS"
)
```

After Jenkins reports the status back to JFrog Pipelines, it does not wait for any response from JFrog Pipelines, and the job continues to the next step immediately.

You also have the option of putting the reportStatus and outputResources together as follows:

```

jfPipelines (
  reportStatus: "SUCCESS",
  outputResources: " "[
    {
      "name": "pipelinesBuildInfo",
      "content": {
        "buildName": "${env.JOB_NAME}",
        "buildNumber": "${env.BUILD_NUMBER}"
      }
    }
  ]" "
)

```

The supported statuses are SUCCESS, UNSTABLE, FAILURE, NOT_BUILT or ABORTED.

The Jenkins job will report the status to JFrog Pipelines only once. If you don't report the status as part of the pipelines script as shown above, Jenkins will report it when the job finishes.

Multi-Configuration (Freestyle) Projects

A [multi-configuration project](#) can be used to avoid duplicating many similar steps that would otherwise be made by different builds.

The plugin is used in the same way as the other Freestyle builds, but under "Deploy artifacts to Artifactory" you will find a mandatory "Combination Matches" field where you can enter the specific matrix combinations to which the plugin will deploy the artifacts.

Deploy artifacts to Artifactory

Combination Matches

 **Mandatory field - You don't have any deploy matches**
Groovy expression

Combination Matches field

Here you can specify build combinations that you want to **deploy** through a Groovy expression that returns true or false. When you specify a Groovy expression here, only the build combinations that result in **true** will be deployed to Artifactory. In evaluating the expression, multi-configuration axes are exposed as variables (with their values set to the current combination evaluated). The Groovy expression uses the same syntax used in **Combination Filter** under **Configuration Matrix**. For example, if you are building on different agents for different jdk's you would specify the following:

Deploy "if both linux and jdk7, it's invalid "	!(label=="linux" && jdk=="jdk7")
Deploy "if on master, just do jdk7 "	(label=="master").implies(jdk=="jdk7")



Important Note

Deployment of the same Maven artifacts by more than one matrix job is not supported!

Triggering Builds

The **Artifactory Trigger** allows a Jenkins job to be automatically triggered when files are added or modified in a specific Artifactory path. The trigger periodically polls Artifactory to check if the job should be triggered.

To enable the Artifactory trigger, follow these steps:

1. In the Jenkins job UI, go to **Build Triggers**, and check the **Enable Artifactory trigger** checkbox.
2. Select an **Artifactory server**.
3. Define a cron expression in the **Schedule** field. For example, to pull Artifactory every ten minutes, set ***/*10 * * * ***
4. Set a **Path to watch**. For example, when setting **generic-libs-local/builds/starship**, Jenkins polls the **/builds/starship** folder under the **generic-libs-local** repository in Artifactory for new or modified files.

Build Triggers

- Build after other projects are built ?
 - Build periodically ?
 - Enable Artifactory trigger
- Artifactory server:
- Schedule: ?
- Path to watch: ?
- GitHub hook trigger for GITScm polling ?
 - Poll SCM ?
 - Disable this project ?
 - Quiet period ?

JIRA Integration

i JIRA Integration is supported only in Free-Style and Maven jobs.

Pipeline jobs support a more generic integration, which allows integrating with any issue tracking system. See the *Collecting Build Issues* section in [Declarative](#) and [Scripted](#) Pipeline APIs documentation pages.

The Jenkins plugin may be used in conjunction with the [Jenkins JIRA plugin](#) to record the build's affected issues, and include those issues in the Build Info descriptor inside Artifactory and as searchable properties on deployed artifacts.

The SCM commit messages must include the JIRA issue ID. For example: *HAP-007 - Shaken, not stirred*

To activate the JIRA integration, make sure that Jenkins is set with a valid JIRA site configuration and select the **Enable JIRA Integration** in the job configuration page:

Enable JIRA Integration

Aggregating Issues from Previous Builds

It is possible to collect under a build deployed to Artifactory all JIRA issues affected by this build as well as previous builds. This allows you, for example, to see all issues between the previous release to the current build, and if the build is a new release build - to see all issues addresses in the new release.

To accumulate JIRA issues across builds, check the "Aggregate issues from previous builds" option and configure the last build status the aggregation should begin from. The default last status is "Released" (case insensitive), which means aggregation will begin from the first build after the last "Released" one.

Aggregate issues from previous builds

Aggregate from status:

Define the latest build status to stop the aggregation at

Build Isolation

i Build Isolation is currently supported only in Free-Style and Maven jobs.

When executing the same chain of integration (snapshot) builds in parallel, a situation may arise in which downstream builds resolve snapshot dependencies which are not the original dependencies existing when the build was triggered. This can happen when a root upstream build has run and triggered downstream builds that depend on its produced artifacts. Then the upstream has run again before the running downstream builds has finished, so these builds may resolve newly created upstream artifacts that are not meant for them, leading to conflicts.

Enable isolated resolution for downstream builds

Solution

The Jenkins plugin offers a new checkbox for its Maven/Gradle builds 'Enable isolated resolution for downstream builds' which plants a new 'build.root' property that is added to the resolution URL. This property will then be read by the **direct** children of this build and implanting them in their resolution URLs respectively, thus guaranteeing that the parent artifact resolved is the one that was built prior to the build being run.

Maven

In order for Maven to use the above feature, the checkbox needs to be checked for the root build only, and make sure that all artifacts are being resolved from Artifactory by using the 'Resolve artifacts from Artifactory' feature. This will enforce Maven to use the resolution URL with Maven builds, alongside with the 'build.root' property as a matrix param in the resolution URL.

Gradle

Once the 'Enable isolated resolution for downstream builds' has been checked, the build.root property will be added to all existing resolvers.

Excluded artifacts and the BuildInfo


Include Patterns 

Exclude Patterns 

Filter excluded artifacts from build Info 

By default when providing exclude patterns for artifacts, they will not get deployed into Artifactory but they will get included in the final BuildInfo JSON. By marking the "Filter excluded artifacts from build Info" the excluded artifacts will appear in a different section inside the BuildInfo and by this providing a clear understanding of the entire Build. This is also crucial for the promotion procedure, since it scans your BuildInfo JSON and trying to promote all the artifacts there, it will fail when you excluded artifacts unless you mark this option.

Discarding Old Builds

 To use this functionality in pipeline jobs, please refer to one of the *Triggering Build Retention* sections in the [Artifactory Pipeline APIs documentation page](#).

The Jenkins project configuration lets you specify a policy for handling old builds.

Discard Old Builds

Strategy

Log Rotation

Days to keep builds

if not empty, build records are only kept up to this number of days

Max # of builds to keep

20

if not empty, only up to this number of build records are kept

You can delete old builds based on age or number as follows:

Days to keep builds	The number of days that a build should be kept before it is deleted
Max # of builds to keep	The maximum number of builds that should be kept. When a new build is created, the oldest one will be deleted

Once these parameters are defined, in the **Post-build Actions** section, you can specify that Artifactory should also discard old builds according to these settings as follows:

Discard old builds from Artifactory (requires Artifactory Pro)

Discard build artifacts

Discard old builds from Artifactory	Configures Artifactory to discard old builds according to the Jenkins project settings above
Discard build artifacts	Configures Artifactory to also discard the artifacts within the build

Configuring repositories with variables

This section is relevant for Free-Style and Maven jobs only,.

Artifactory server

Resolution repository

Publishing repository

You can select text mode in which you can type out your target repository.

In your target repository name, you can use variables that will be dynamically replaced with a value at build time.

The variables should be specified with a dollar-sign prefix and be enclosed by curly brackets.

For example: `${deployRepository}`, `${resolveSnapshotRepository}`, `${repoPrefix}-${repoName}` etc.

The variables are replaced by values from one of the following job environments:

1. Predefined Jenkins environment variables.
2. Jenkins properties (Read more in [Jenkins Wiki](#))
3. Parameters configured in the Jenkins configuration under the "This build is parameterized" section - these parameters could be replaced by a value from the UI or using the Jenkins REST API.
4. Injected variables via one of the Jenkins plugins ("EnvInject" for example).

Dynamically Disabling Deployment of Artifacts and Build-info

Maven, Gradle and Ivy builds can be configured to deploy artifacts and/or build information to Artifactory. For example, in the case of Gradle builds, you would set the **Publishing repository** field and check **Capture and publish build-info**. Maven and Ivy have similar (although slightly different) configuration parameters. By setting these parameters, you can configure each build tool respectively to deploy build artifacts and/or build information to Artifactory. However, there may be specific cases in which you want to override this setting and disable deploying artifacts and build information. In these cases, you can pass the following two system properties to any of the build tools:

- `artifactory.publish.artifacts`
- `artifactory.publish.buildInfo`

For example, for a Maven build, you can add these system properties to the **Goals and options** field as follows:

```
clean install -Dartifactory.publish.artifacts=false -Dartifactory.publish.buildInfo=false
```

To control these properties dynamically, you can replace the values with Jenkins variables or Environment variables that you define as follows

```
clean install -Dartifactory.publish.artifacts=${PUBLISH_ARTIFACTS} -Dartifactory.publish.buildInfo=${PUBLISH_BUILDINFO}
```

Using the Jenkins Job DSL Plugin

The Jenkins Job DSL plugin allows the programmatic creation of jobs using a DSL. Using the Jenkins Job DSL plugin, you can create Jenkins jobs to run Artifactory operations. To learn about the Jenkins Job DSL, see the [Job DSL Tutorial](#).

To view Seed job examples and instructions for each type of Jenkins jobs, see [jenkins-job-dsl-examples](#).

Watch the Screencast

To see the Jenkins Artifactory Integration in action you can watch the following screencast.

Release Notes

3.13.1 (2 September 2021)

1. Bug fix - NPE in Gradle jobs when deployer is empty - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/554>
2. Bug fix - NPE in Gradle when build info was not generated - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/552>
3. Bug fix - NPE in extractVcs when running on the master - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/549>

3.13.0 (17 August 2021)

1. Show deployed Gradle artifacts - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/538>
2. Improved build scan table - <https://github.com/jfrog/build-info/pull/545>
3. Bug fix - Autofill JFrog servers get triggered for the wrong server-id - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/534>
4. Bug fix - Go jobs don't run on Docker images when they should - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/544>
5. Bug fix - Go jobs don't collect environment variables when they should - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/544>

3.12.5 (20 July 2021)

1. Bug fix - Exception raised when workflow - multibranch plugin is not installed - <https://github.com/jfrog/build-info/pull/530>
2. Bug fix - Fail to deserialize deployable artifacts - <https://github.com/jfrog/build-info/pull/537>
3. Gradle - Skip uploading JAR if no jar produced in build - <https://github.com/jfrog/build-info/pull/538>
4. Make download headers comparisons case insensitive - <https://github.com/jfrog/build-info/pull/539>

3.12.4 (7 July 2021)

1. Bug fix - Save JFrog instances for the first time throws NPE - <https://github.com/jfrog/build-info/pull/527>

3.12.3 (5 July 2021)

1. Bug fix - ReportUsage throws an exception for Artifactory version 6.9.0 - <https://github.com/jfrog/build-info/pull/525>
2. Bug fix - BuildInfo link is broken if contains special characters - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/525>

3.12.1 (30 Jun 2021)

1. Add new JFrog Distribution commands - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/518>
2. Support collecting build-info for Docker images created by Kaniko and JIB - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/503>
3. Support Maven wrapper - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/482>
4. Add project option to scan build - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/517>
5. Support multibranch Artifactory trigger - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/507>
6. Bug fix - Legacy patterns doesn't display defined instances - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/510>
7. Bug fix - Link to BuildInfo page gets encoding twice - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/516>
8. Bug fix - In some cases, NoSuchMethodError raised using IOUtils.toString - <https://github.com/jfrog/build-info/pull/516>

3.11.4 (31 May 2021)

1. Bug fix - Error when trying to download an empty (zero bytes size) file - <https://github.com/jfrog/build-info/pull/507>
2. Bug fix - Deploy file doesn't print full URL in the log output - <https://github.com/jfrog/build-info/pull/509>

3.11.3 (29 May 2021)

1. Bug fix - Artifactory servers are missing in Jenkins dropdown list - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/489>
2. Bug fix - Ignore missing fields while deserialize HTTP response - <https://github.com/jfrog/build-info/pull/502>
3. Bug fix - Build retention service sends an empty body - <https://github.com/jfrog/build-info/pull/504>
4. Bug fix - Artifactory Trigger cannot deserialize instance ItemLastModified - <https://github.com/jfrog/build-info/pull/503>

3.11.2 (26 May 2021)

1. Bug fix - Set bypassProxy in the configuration page fails <https://github.com/jfrog/jenkins-artifactory-plugin/pull/486>

3.11.1 (25 May 2021)

1. Gradle - Improve Gradle transitive dependency collection - <https://github.com/jfrog/build-info/pull/498>
2. Bug fix - Compatibility with JCasC fails - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/478>

3.11.0 (19 May 2021)

1. Update commons-codec and commons-io - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/462>
2. Add JFrog platform URL to Jenkins configuration page - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/455>
3. Add support for Artifactory Projects - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/449>
4. Add support for npm 7.7 - <https://github.com/jfrog/build-info/pull/484>
5. Add support for NuGet V3 protocol - <https://github.com/jfrog/build-info/pull/479> & <https://github.com/jfrog/build-info/pull/494>
6. Bug fix - Gradle init script with space in path causes failure - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/469>
7. Bug fix - env doesn't collected in npm, Go, Pip, and NuGet builds - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/468>

3.10.6 (23 Mar 2021)

1. Bug fix - root workspace detection from flow - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/432>
2. Bug fix - tests for upload files with props omit multiple slashes - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/436>
3. Bug fix - Legacy patterns in existing Freestyle jobs are messed-up - <https://github.com/jfrog/jenkins-artifactory-plugin/issues/434>
4. Bug fix - upload files with props omit multiple slashes - <https://github.com/jfrog/build-info/issues/460>
5. Bug fix - Git-collect-issues should ignore errors when the revision of the previous build info doesn't exist - <https://github.com/jfrog/build-info/issues/457>
6. Add usage report for pipeline APIs - <https://github.com/jfrog/jenkins-artifactory-plugin/pull/441>
7. Populate requestedBy field in NPM build-info - <https://github.com/jfrog/build-info/pull/446>
8. Populate requestedBy field in Gradle build-info - <https://github.com/jfrog/build-info/pull/454>

3.10.5 (25 Feb 2021)

1. Bug fixes ([HAP-1405](#), [HAP-1442](#), [HAP-1450](#))

3.10.4 (25 Jan 2021)

1. Legacy patterns option removed from newly created Freestyle jobs
2. Support a new ALL_PUBLICATIONS option bt gradle pipeline deployer API
3. Bug fix - java.lang.UnsatisfiedLinkError
4. Bug fix - Conan - incorrect quotation marks
5. Bug fix - Allow rtBuildInfo to configure existing server configuration
6. Bug fix - Freestyle jobs UI crashes following recent changes in Jenkins

3.10.3 (3 Jan 2021)

1. Bug fix ([HAP-1439](#))

3.10.2 (3 Jan 2021)

1. Bug fix ([HAP-1438](#))

3.10.1 (29 December 2020)

1. Bug fixes and improvements ([HAP-1389](#), [HAP-1417](#), [HAP-1419](#), [HAP-1420](#), [HAP-1421](#), [HAP-1422](#), [HAP-1423](#), [HAP-1424](#), [HAP-1425](#), [HAP-1429](#), [HAP-1430](#), [HAP-1431](#), [HAP-1432](#), [HAP-1433](#), [HAP-1434](#), [HAP-1435](#))

3.10.0 (1 December 2020)

1. New "npm ci" pipelines API ([HAP-1338](#))
2. Support for build-info aggregation across different agents ([HAP-1412](#))
3. New build scan summary table added to the build log ([HAP-1413](#))
4. Bug fixes ([HAP-1415](#), [HAP-1416](#))

3.9.1 (25 November 2020)

1. File-Specs: Support wildcard in repositories, Add new 'exclusions' field which allows excluding repositories. ([HAP-1409](#))
2. Bug fixes ([HAP-1401](#), [HAP-1408](#), [HAP-1410](#), [HAP-1411](#))

3.9.0 (27 October 2020)

1. Support Docker pull ([HAP-1397](#))
2. Fix and improve interactive promotions UI ([HAP-1394](#))
3. Get Artifactory trigger path which caused the build ([HAP-1246](#))
4. Breaking change - BuildInfo.getArtifacts() Api returns the artifact's remote path without repository ([HAP-1426](#)).
5. Bug fixes ([HAP-1398](#), [HAP-1384](#), [HAP-1399](#), [HAP-1385](#), [HAP-1382](#), [HAP-1396](#), [HAP-1392](#), [HAP-1391](#), [HAP-1390](#), [HAP-1379](#), [HAP-1395](#))

3.8.1 (28 August 2020)

1. Bug fix ([HAP-1378](#))

3.8.0 (17 August 2020)

1. Breaking change - Push / pull docker image steps use as a separate Java process ([HAP-1375](#))
2. Define Artifactory Trigger using pipeline script ([HAP-1373](#))
3. Support for NuGet & .NET Core CLI in scripted and declarative pipelines ([HAP-1370](#))
4. Support for Python in scripted and declarative pipelines ([HAP-1369](#))
5. Allow defining Gradle Publications in scripted and declarative pipelines ([HAP-1367](#))
6. Display the list of deployed artifacts on the job summary page for maven builds ([HAP-1346](#))
7. Support for JCasC (Jenkins Configuration as Code) ([HAP-1092](#))
8. Bug fixes ([HAP-1368](#), [HAP-1372](#))

3.7.2 (7 July 2020)

1. Bug fixes ([HAP-1307](#), [HAP-1364](#), [HAP-1363](#), [HAP-1092](#))

3.7.0 (25 June 2020)

1. Integration with JFrog Pipelines ([HAP-1348](#))
2. Declarative pipeline support for Conan ([HAP-1360](#))
3. Avoid assigning builds a FAILURE result ([HAP-1357](#))

3.6.2 (4 May 2020)

1. Bug fixes and improvements ([HAP-1329](#), [HAP-1339](#), [HAP-1341](#), [HAP-1342](#))

3.6.1 (19 Mar 2020)

1. Bug fixes ([HAP-1303](#), [HAP-1322](#))

3.6.0 (9 Mar 2020)

1. Parallel maven and gradle deployments ([HAP-1308](#))
2. Build-info module name customisation for generic, npm and go builds ([HAP-1259](#))
3. Docker foreign layers support ([HAP-1314](#))
4. Jenkins core API bump ([HAP-1312](#))
5. Support the conan client's verify SSL option ([HAP-1309](#))
6. Fixes and improvements ([HAP-1313](#), [HAP-1311](#), [HAP-1305](#), [HAP-1304](#), [HAP-1303](#), [HAP-1299](#), [HAP-1297](#), [HAP-1295](#), [HAP-1290](#), [HAP-1286](#), [HAP-1283](#), [HAP-1286](#), [HAP-1286](#), [HAP-1395](#))

3.5.0 (30 Dec 2019)

1. New Go pipeline APIs ([HAP-1272](#))
2. Support access token auth with Artifactory ([HAP-1271](#))
3. File Specs - support for sorting. ([HAP-1215](#))
4. Gradle pipeline - support defining snapshot and release repositories. ([HAP-1174](#))
5. Replace the usage of gradle's maven plugin with the maven-publish plugin. ([HAP-1096](#))
6. NPM pipeline - allow running inside a container. ([HAP-1261](#))
7. New Xray Scan Report icon. ([HAP-1274](#))
8. Bug fixes ([HAP-1265](#), [HAP-1264](#), [HAP-1250](#), [HAP-1240](#), [HAP-1243](#), [HAP-954](#), [HAP-1268](#))

3.4.1 (23 Sep 2019)

1. Bug fixes ([HAP-1235](#), [HAP-1236](#), [HAP-1238](#))

3.4.0 (5 Sep 2019)

1. Allow adding projects issues to the build-info in pipeline jobs ([HAP-1231](#))
2. Attach vcs.url and vcs.revision to files uploaded from a pipeline job ([HAP-1233](#))
3. Conan remote.add pipeline API - support "force" and "remoteName" ([HAP-1232](#))
4. Bug fixes ([HAP-1230](#), [HAP-1229](#), [HAP-1225](#), [HAP-1224](#), [HAP-1219](#), [HAP-1218](#), [HAP-1214](#), [HAP-1210](#), [HAP-1209](#), [HAP-1190](#))

3.3.2 (2 July 2019)

1. Change Xray connection timeout to 300 seconds ([HAP-1213](#))
2. Add *psw* to the exclude environment variables list of pipeline jobs only. ([HAP-1212](#))
3. Bug fixes ([HAP-1204](#), [HAP-1205](#))

3.3.0 (17 June 2019)

1. Declarative pipeline API for Xray build scan ([HAP-1175](#))
2. Declarative pipeline API for docker push ([HAP-1201](#))
3. Bug fix ([HAP-1200](#))

3.2.4 (5 June 2019)

1. Bug fixes ([HAP-1195](#), [HAP-1167](#), [HAP-1197](#))

3.2.3 (4 June 2019)

1. Bug fixes & Improvements ([HAP-1191](#), [HAP-1183](#), [HAP-1182](#), [HAP-1176](#), [HAP-1172](#))

3.2.2 (31 Mar 2019)

1. Bug fixes & Improvements ([HAP-1150](#), [HAP-1151](#), [HAP-1157](#), [HAP-1158](#), [HAP-1160](#), [HAP-1164](#), [HAP-1168](#), [HAP-1169](#))

3.2.1 (20 Feb 2019)

1. Bug fix ([HAP-1156](#))

3.2.0 (18 Feb 2019)

1. New Set/Delete Properties step in generic pipeline ([HAP-1153](#))
2. Add an option to get a list of all downloaded artifacts ([HAP-1114](#))

3.1.2 (11 Feb 2019)

1. Bug fix ([HAP-1146](#))

3.1.1 (10 Feb 2019)

1. Add the ability to control the number of parallel uploads using File Specs ([HAP-1085](#))
2. Bug fixes & Improvements ([HAP-1116](#), [HAP-1136](#), [HAP-1137](#), [HAP-1140](#), [HAP-1143](#), [HAP-1144](#), [HAP-1145](#))

3.1.0 (16 Jan 2019)

1. Support for NodeJS plugin in npm builds ([HAP-1127](#))
2. Support for Declarative syntax in npm builds ([HAP-1128](#))
3. Bug fixes & Improvements ([HAP-1130](#), [HAP-1132](#), [HAP-1133](#), [HAP-1134](#))

3.0.0 (31 Dec 2018)

1. Upgrade to Java 8 - Maven, Gradle and Ivy builds no longer support JDK 7
2. Support for Declarative syntax in Generic Maven and Gradle builds ([HAP-1093](#))
3. Support for npm in scripted pipeline ([HAP-1044](#))
4. Support for Artifactory trigger ([HAP-1012](#))
5. New Fail No Op flag in generic pipeline ([HAP-1123](#))
6. Breaking changes: Removal of deprecated features ([HAP-1119](#))
7. Bug fixes & Improvements ([HAP-1120](#), [HAP-1121](#), [HAP-1122](#), [HAP-1113](#), [HAP-1112](#), [HAP-1110](#), [HAP-1102](#), [HAP-1098](#), [HAP-1090](#), [HAP-1124](#))

2.16.2 (9 Jul 2018)

1. Docker build-info performance improvement ([HAP-1075](#))
2. Bug fixes ([HAP-1057](#), [HAP-1076](#), [HAP-1083](#), [HAP-1086](#), [HAP-1087](#))

2.16.1 (3 May 2018)

1. Support "proxyless" configuration for docker build-info ([HAP-1061](#))
2. Bug fixes ([HAP-1043](#), [HAP-1068](#), [HAP-1070](#))

2.16.0 (1 May 2018)

1. File Specs - Deployment of Artifacts is now parallel ([HAP-1066](#))
2. Conan build with Docker improvement ([HAP-1055](#))
3. Bug fixes & improvements ([HAP-1000](#), [HAP-1049](#), [HAP-1058](#), [HAP-1062](#), [HAP-1065](#))

2.15.1 (1 Apr 2018)

1. Bug fixes ([HAP-1046](#), [HAP-1047](#), [HAP-1048](#), [HAP-1051](#), [HAP-1052](#))

2.15.0 (14 Mar 2018)

1. Support Jenkins [JEP-200](#) changes ([HAP-1032](#))
2. File Specs - File larger than 5mb are now downloaded concurrently ([HAP-1041](#))
3. Add support for Jenkins job-DSL ([HAP-1028](#))
4. Validate git credentials early in Artifactory Release Staging ([HAP-1027](#))
5. Bug fixes & improvements ([HAP-1042](#), [HAP-1039](#), [HAP-1031](#), [HAP-1030](#), [HAP-1029](#), [HAP-1026](#), [HAP-1025](#), [HAP-1024](#), [HAP-1021](#), [HAP-1019](#), [HAP-999](#), [HAP-970](#))

2.14.0 (28 Dec 2017)

1. Docker build info without the Build Info Proxy ([HAP-1016](#))
2. Bug fixes ([HAP-1001](#), [HAP-1003](#), [HAP-1007](#), [HAP-1008](#), [HAP-1009](#), [HAP-1010](#), [HAP-1013](#))

2.13.1 (26 Oct 2017)

1. Bug fixes ([HAP-1002](#), [HAP-1004](#), [HAP-1006](#))

2.13.0 (27 Sep 2017)

1. Support JFrog DSL from within Docker containers ([HAP-937](#))
2. Release Staging API - Send release and queue metadata back as JSON in the response body ([HAP-971](#))
3. Allow adding properties in a pipeline docker push ([HAP-974](#), [HAP-977](#))
4. Support pattern exclusion in File Specs ([HAP-985](#))
5. File specs AQL optimizations ([HAP-990](#))
6. Bug fixes ([HAP-961](#), [HAP-962](#), [HAP-964](#), [HAP-969](#), [HAP-972](#), [HAP-978](#), [HAP-980](#), [HAP-981](#), [HAP-983](#), [HAP-988](#), [HAP-991](#))

2.12.2 (27 Jul 2017)

1. Bug fixes ([HAP-944](#), [HAP-945](#), [HAP-946](#), [HAP-951](#), [HAP-952](#), [HAP-953](#), [HAP-958](#), [HAP-959](#), [HAP-961](#))

2.12.1 (11 Jul 2017)

1. Bug fixes ([HAP-949](#), [HAP-950](#))

2.12.0 (29 Jun 2017)

1. Pipepine - Allow separation of build and deployment for maven and gradle ([HAP-919](#))
2. Git commit information as part of Build Info ([HAP-920](#))
3. Support asynchronous build retention ([HAP-934](#))
4. Support archive extraction using file specs ([HAP-942](#))
5. Bug fixes ([HAP-933](#), [HAP-929](#), [HAP-912](#), [HAP-941](#), [HAP-940](#), [HAP-935](#), [HAP-938](#))

2.11.0 (17 May 2017)

1. Support distribution as part of the Pipeline DSL ([HAP-908](#))
2. Compatibility with JIRA Plugin 2.3 ([HAP-928](#))
3. Bug fixes ([HAP-915](#), [HAP-917](#), [HAP-927](#))

2.10.4 (27 Apr 2017)

1. Bug fixes ([HAP-900](#), [HAP-909](#), [HAP-913](#), [HAP-918](#))

2.10.3 (5 Apr 2017)

1. Support interactive promotion in Pipeline jobs ([HAP-891](#))
2. Conan commands support for Pipeline jobs ([HAP-899](#))
3. Improve build-info links for multi build Pipeline jobs ([HAP-901](#))
4. Bug fixes ([HAP-556](#), [HAP-855](#), [HAP-887](#), [HAP-897](#))

2.9.2 (5 Mar 2017)

1. Bug fixes ([HAP-892](#), [HAP-889](#))

2.9.1 (1 Feb 2017)

1. Bug fixes ([HAP-877](#), [HAP-883](#), [HAP-884](#), [HAP-879](#))

2.9.0 (7 Jan 2017)

1. Add to file spec the ability to download artifact by build name and number ([HAP-865](#))
2. Support Release Management as part of the Pipeline DSL ([HAP-797](#))
3. Capture docker build-info from all agents proxies ([HAP-868](#))
4. Support for Xray scan build feature ([HAP-866](#))
5. Support customised build name ([HAP-869](#))
6. Change file Specs pattern ([HAP-876](#))
7. Bug fixes ([HAP-872](#), [HAP-873](#), [HAP-870](#), [HAP-856](#), [HAP-862](#))

2.8.2 (6 Dec 2016)

1. Support promotion fail fast ([HAP-803](#))
2. Enable setting the JDK for Maven/Gradle Pipeline API ([HAP-848](#))
3. Support environment variables within Pipeline specs ([HAP-849](#))

4. Improve release staging with git ([HAP-842](#))
5. Bug fixes ([HAP-854](#), [HAP-852](#), [HAP-846](#), [HAP-843](#), [HAP-791](#), [HAP-716](#), [HAP-488](#))

2.8.1 (10 Nov 2016)

1. Bug fix ([HAP-841](#))

2.8.0 (9 Nov 2016)

1. Upload and download specs support to Jenkins generic jobs ([HAP-823](#))
2. Pipeline docker enhancements ([HAP-834](#))
3. Support matrix params as part of the Maven and Gradle Pipeline DSL ([HAP-835](#))
4. Support credentials ID as part of the Docker Pipeline DSL ([HAP-838](#))
5. Support reading download and upload specs from FS ([HAP-838](#))
6. Bug fix ([HAP-836](#), [HAP-830](#), [HAP-829](#), [HAP-822](#), [HAP-824](#), [HAP-816](#), [HAP-828](#), [HAP-826](#))

2.7.2 (25 Sep 2016)

1. Build-info support for docker images within pipeline jobs ([HAP-818](#))
2. Support for Maven builds within pipeline jobs ([HAP-759](#))
3. Support for Gradle builds within pipeline jobs ([HAP-760](#))
4. Support for Credentials plugin ([HAP-810](#))
5. Bug fix ([HAP-723](#), [HAP-802](#), [HAP-804](#), [HAP-815](#), [HAP-816](#))

2.6.0 (7 Aug 2016)

1. Pipeline support for build retention (discard old builds) ([HAP-796](#))
2. Pipeline support for build promotion, environment variables and system properties filtered collection ([HAP-787](#))
3. Expose release version and next development release property as environment variable ([HAP-798](#))
4. Bug fix ([HAP-772](#), [HAP-762](#), [HAP-795](#), [HAP-796](#), [HAP-799](#))

2.5.1 (30 Jun 2016)

1. Bug fix ([HAP-771](#))

2.5.0 (9 Jun 2016)

1. Pipeline support for Artifactory ([HAP-625](#), [HAP-722](#))
2. Support for credentials using inside a Folder with the Credentials Plugin ([HAP-742](#))
3. Configure from the job a default release repository from configuration page ([HAP-688](#))
4. Allowing override Git credentials at artifactory release staging page ([HAP-626](#))
5. Bug fix ([HAP-754](#), [HAP-752](#), [HAP-747](#), [HAP-736](#), [HAP-726](#), [HAP-722](#), [HAP-715](#), [HAP-712](#), [HAP-704](#), [HAP-695](#), [HAP-688](#), [HAP-671](#), [HAP-642](#), [HAP-639](#))

2.4.7 (12 Jan 2016)

1. Bug fix ([HAP-678](#))

2.4.6 (13 Dec 2015)

1. Bug fix ([HAP-674](#))

2.4.5 (6 Dec 2015)

1. Bug fixes ([HAP-665](#), [HAP-668](#))

2.4.4 (17 Nov 2015)

1. Bug fixes ([HAP-661](#), [HAP-663](#), [HAP-664](#))

2.4.1 (6 Nov 2015)

1. Bug fix ([HAP-660](#))

2.4.0 (2 Nov 2015)

1. Use the Credentials Plugin;([HAP-491](#))
2. FreeStyle Jobs - Support different Artifactory server for resolution and deployment;([HAP-616](#))
3. Jenkins should write the Artifactory Plugin version to the build log and build-info json;([HAP-620](#))
4. Bug fixes ([HAP-396](#), [HAP-534](#), [HAP-583](#), [HAP-616](#), [HAP-617](#), [HAP-618](#), [HAP-621](#), [HAP-622](#), [HAP-641](#), [HAP-646](#))

2.3.1 (13 Jul 2015)

1. Expose Release SCM Branch and Release SCM Tag as build variables;([HAP-606](#))
2. Bug fixes ([HAP-397](#), [HAP-550](#), [HAP-576](#), [HAP-593](#), [HAP-603](#), [HAP-604](#), [HAP-605](#), [HAP-609](#))

2.3.0 (01 Apr 2015)

1. Push build to Bintray directly from Jenkins UI ([HAP-566](#))
2. Multijob (plugin job) type not supported by Artifactory Plugin ([HAP-527](#))
3. Support multi-configuration projects ([HAP-409](#))
4. "Target-Repository" - need a dynamic parameter ([HAP-547](#))
5. Bug fixes ([HAP-585](#), [HAP-573](#), [HAP-574](#), [HAP-554](#), [HAP-567](#))

2.2.7 (27 Jan 2015)

1. Add resolve artifacts from Artifactory to the Free Style Maven 3 integration ([HAP-379](#))
2. Bug fixes ([HAP-411](#), [HAP-553](#), [HAP-555](#))

2.2.5 (18 Dec 2014)

1. Maven jobs - Record Implicit Project Dependencies and Build-Time Dependencies ([HAP-539](#))
2. Possibility to refer to git url when using target remote for Release management ([HAP-525](#))
3. Bug fixes ([HAP-537](#), [HAP-542](#), [HAP-535](#), [HAP-528](#), [HAP-516](#), [HAP-507](#), [HAP-484](#), [HAP-454](#), [HAP-538](#), [HAP-523](#), [HAP-548](#))

2.2.4 (21 Aug 2014)

1. New Artifactory Release Management API ([HAP-503](#))
2. Compatibility with Gradle 2.0 ([GAP-153](#))
3. Job configuration page performance improvements ([HAP-492](#))
4. Bug fixes ([HAP-485](#), [HAP-499](#)), [HAP-502](#), [HAP-508](#), [HAP-509](#), [HAP-301](#))

2.2.3 (10 Jun 2014)

1. Artifactory plugin is back to support Maven 2 ([HAP-459](#))
2. New feature, "Refresh Repositories" button, a convenient way to see your available repositories that exists in the configured Artifactory. This feature also improves the Job page load response, and fixes the following bug ([HAP-483](#))
3. Supporting "Subversion plugin" version 2+, on their compatibility with "Credentials plugin" ([HAP-486](#))
4. Bug fixes ([HAP-489](#), [HAP-480](#))

2.2.2 (21 May 2014)

1. Split Resolution repository to Snapshot and Release ([HAP-442](#))
2. Supporting Git plugin credentials ([HAP-479](#))
3. Upgrading and also minimum supporting version of Git plugin 2.0.1 (recommended 2.0.4 for the credentials feature)
4. Fix bug with Maven release plugin ([HAP-373](#))
5. Adding Version Control Url property to the Artifactory Build Info JSON ([HAP-478](#))
6. Bug fixes ([HAP-432](#), [HAP-470](#))

2.2.1 (11 Nov 2013)

1. Fix for IllegalArgumentException in Deployment when no deployment is defined in Job ([HAP-241](#))

2.2.0 (16 Oct 2013)

1. Fix parent pom resolution issue ([HAP-236](#)) from Jenkins 1.521
2. Add support for maven 3.1.X
3. Option to ignore artifacts that are not deploy because of include/exclude patterns from the build info ([HAP-444](#))
4. Enable credentials configuration for repository listing per project ([HAP-430](#))
5. Bug fixes

2.1.8 (26 Aug 2013)

1. Fix migration to Jenkins 1.528 ([HAP-428](#))

2.1.7 (31 Jul 2013)

1. Maven build failure during deployment ([HAP-420](#))
2. Bug fixes ([HAP-406](#))

2.1.6 (24 Jun 2013)

1. Fix plugin compatibility with Jenkins 1.519 ([HAP-418](#))

2.1.5 (23 Apr 2013)

1. Black duck integration - Automatic Black duck Code-Center integration for open source license governance and vulnerability control ([HAP-394](#))
2. Gradle 1.5 support for maven and ivy publishes - New 'artifactory-publish' plugin with fully supported for Ivy and Maven publications ([GAP-138](#))

3. Bug fixes ([HAP-341](#), [HAP-390](#), [HAP-366](#), [HAP-380](#))

2.1.4 (03 Feb 2013)

1. Generic resolution interpolates environment variables ([HAP-352](#))
2. Broken link issues ([HAP-362](#),[HAP-371](#),[HAP-360](#))
3. [Minor bug fixes](#)

2.1.3 (14 Oct 2012)

1. Support include/exclude patterns of captured environment variables ([BI-143](#))
2. Bug fixes ([HAP-343](#),[HAP-4](#),[GAP-136](#))

2.1.2 (08 Aug 2012)

1. Aggregating Jira issues from previous builds ([HAP-305](#))
2. Bug fixes and improvements in generic deploy ([HAP-319](#), [HAP-329](#))

2.1.1 (31 May 2012)

1. NPE on Maven2 builds ([HAP-316](#))

2.1.0 (24 May 2012)

1. Support for cloudbees 'Folder plugin' ([HAP-312](#),[HAP-313](#))
2. [Minor bug fixes](#)

2.0.9 (15 May 2012)

1. Fix UI integration for Jenkins 1.463+ ([HAP-307](#))
2. [Minor bug fixes](#)

2.0.8 (09 May 2012)

1. [Integration with Jira plugin](#) ([HAP-297](#))
2. Support build promotion for all build types ([HAP-264](#))
3. Ability to leverage custom user plugins for [staging and promotion](#) ([HAP-271](#), [HAP-272](#))

2.0.7 (20 Apr 2012)

1. Generic artifact resolution (based on patterns or other builds output) to freestyle builds ([HAP-292](#))
2. Optimized deploy - when a binary with the same checksum as an uploaded artifact already exists in the Artifactory storage, a new local reference will be created instead of reuploading the same content
3. Bug fixes

2.0.6 (19 Mar 2012)

1. Support Perforce in release management ([HAP-265](#))
2. Generic artifacts deployment ([HAP-153](#))
3. [Bug fixes](#)

2.0.5 (08 Dec 2011)

1. Compatible with Gradle 1.0-milestone-6
2. Different Artifactory servers can be used for resolution and deployment ([HAP-203](#))
3. Using the new Jenkins user cause class to retrieve triggering user. **Requires Jenkins 1.428 or above** ([HAP-254](#))
4. Release management with Git work with the latest plugin. Requires Git plugin v1.1.13 or above ([HAP-259](#), [JENKINS-12025](#))
5. Build-info exports an environment variable 'BUILDINFO_PROPFILE' with the location of the generated build info properties file

2.0.4 (15 Aug 2011)

1. Compatible with Jenkins 1.424+ ([HAP-223](#))
2. Resolved Maven 3 deployments intermittently failing on remote nodes ([HAP-220](#))
3. Target repository for staged builds is now respected for Maven 3 builds ([HAP-219](#))
4. Remote builds no longer fail when "always check out a fresh copy" is used ([HAP-224](#))

2.0.3 (26 Jul 2011)

1. Support for Git Plugin v1.1.10+ ([HAP-217](#))
2. Native maven 3 jobs doesn't work if the Jenkins home path contains spaces ([HAP-218](#))
3. Wrong tag URL is used when changing scm element during staged build ([HAP-215](#))

2.0.2 (07 Jul 2011)

1. Support Jenkins version 1.417+ ([HAP-211](#))

2.0.1 (19 May 2011)

1. Maven deployment from remote slaves - artifact deployment for Maven builds will run directly from a remote slave when artifact archiving is turned off, saving valuable bandwidth and time normally consumed by copying artifacts back to master for archiving and publishing (requires Maven 3.0.2 and above)
2. Staging of Maven builds now correctly fails if snapshot dependencies are used in POM files ([HAP-183](#))
3. All staging and promotion commit comments are now customizable ([HAP-181](#))
4. Fix for staged builds failing on remote slaves ([HAP-189](#))

2.0.0 (4 May 2011)

1. [Release management with staging and promotion](#) support
2. Support for forcing artifact resolution in Maven 3 to go through Artifactory ([HAP-144](#))
3. [Isolated resolution](#) for snapshot build chains for Maven and Gradle
4. Ability to attach custom properties to published artifacts ([HAP-138](#))
5. Improved Ant/Ivy integration
6. Improved Gradle integration
7. Support saving pinned builds ([HAP-129](#), [HAP-140](#))
8. Option to delete deployed artifacts when synchronizing build retention ([HAP-161](#))

1.4.3 (7 Apr 2011)

1. Compatible to work with Jenkins 1.405 ([HAP-159](#))

1.4.2 (27 Jan 2011)

1. The plugin now works with Jenkins' new native Maven 3 jobs ([HAP-130](#), [HAP-131](#))

1.4.1 (10 Jan 2011)

1. Synchronize the build retention policy in Artifactory with Jenkins' build retention settings (requires Artifactory Pro) ([HAP-90](#))

1.4.0 (09 Jan 2011)

1. Improved Gradle support
2. Optimized checksum-based publishing with Artifactory 2.3.2+ that saves redeploying the same binaries ([RTFACT-3808](#))
3. Remote agent support for Gradle, Maven 3 and Ivy builds ([HAP-59](#), [HAP-60](#), [HAP-114](#))
4. Configurable ivy/artifact patterns for Ivy builds ([HAP-120](#))

1.3.6 (21 Nov 2010)

1. Allow specifying include/exclude patterns for published artifacts ([HAP-61](#)).
2. Support for custom Ivy/artifact patterns for Gradle published artifacts ([HAP-108](#)).

1.3.5 (7 Nov 2010)

1. Fixed integration with Jenkins maven release plugin. ([HAP-93](#))
2. Global Artifactory credentials ([HAP-53](#))
3. Auto preselect target release and snapshot repositories. ([HAP-98](#))

1.3.4 (28 Oct 2010)

1. Fixed Gradle support

1.3.3 (21 Oct 2010)

1. Update version of the Gradle extractor.

1.3.2 (19 Oct 2010)

1. Support for running license checks on third-party dependencies and sending license violation email notifications ([HAP-91](#))

1.3.1 (19 Sep 2010)

1. Maven 2 and Maven 3 support two target deploy repositories - releases and snapshots ([HAP-29](#))
2. Maven 2 - Allow deployment even if the build is unstable ([HAP-77](#))
3. Link to the build info next to each build that deployed build info ([HAP-80](#))
4. Link to the builds list in the jobs' main page ([HAP-41](#))
5. Allow skipping the creation and deployment of the build info ([HAP-47](#))

1.3.0 (26 Aug 2010)

1. **New support for Maven 3 Beta builds!**

1.2.0 (26 Jul 2010)

1. **New support for Ivy builds!** (many thanks to Timo Bingaman for adding the hooks to the the [Jenkins Ivy Plugin](#))
2. Supporting incremental builds ([HAP-52](#))
3. Testing connection to Artifactory in the main configuration page
4. Update Jenkins dependency to version 1.358
5. Fixed [HAP-51](#) - tar.gz files were deployed as .gz files

1.1.0 (09 Jun 2010)

1. Added support for gradle jobs, see: <http://www.jfrog.org/confluence/x/tYK5>
2. Connection timeout setting changed from milliseconds to seconds.
3. Allow bypassing the http proxy ([JENKINS-5892](#))

1.0.7 (04 Mar 2010)

1. Improved Artifactory client
2. Another fix for duplicate pom deployments
3. Sending parent (upstream) build information
4. Displaying only local repositories when working with Artifactory 2.2.0+

1.0.6 (16 Feb 2010)

1. Fixed a bug in the plugin that in some cases skipped deployment of attached artifacts
2. In some cases, pom were deployed twice
3. MD5 hash is now set on all files
4. Dependency type is passed to the build info

1.0.5 (22 Jan 2010)

1. Using Jackson as JSON generator for BuildInfo (will fix issues with Hudson version 1.340-1.341)

1.0.4 (15 Jan 2010)

1. Accept Artifactory urls with slash at the end
2. Fixed JSON object creation to work with Hudson 1.340

1.0.3 (07 Jan 2010)

1. Using preemptive basic authentication

1.0.2 (22 Dec 2009)

1. Configurable connection timeout

1.0.1 (16 Dec 2009)

1. Fixed Artifactory plugin relative location (for images and help files)

1.0.0 (14 Dec 2009)

1. First stable release