

# Generic Builds

## Overview

The Jenkins Artifactory Plugin supports Generic build jobs, allowing your build jobs deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them.

### Page Contents

- [Overview](#)
- [Configuring Generic \(Freestyle\) Builds](#)
  - [File Specs](#)
  - [Legacy Patterns \(deprecated\)](#)

### Integration Benefits

[JFrog Artifactory and Jenkins CI](#)

## Enter an item name

» Required field



### Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



### Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



### External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.



### Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



### MultiJob Project

MultiJob Project, suitable for running other jobs



### Ivy project

Build an Ivy project. Hudson takes advantage of your Ivy module descriptor files to provide additional functionality.

OK

Add to current view

Folder

## Configuring Generic (Freestyle) Builds

Generic build integration provides Build Info support for any build type. This allows custom builds, such as non-Java builds to:

1. Upload any artifacts to Artifactory, together with custom properties metadata, and keep published artifacts associated with the Jenkins build.
2. Download artifacts from Artifactory that are required by your build.

You can define the artifacts to upload and download by either using "File Specs" or "Legacy Patterns".

## File Specs

File Spec are specified in JSON format.

You can use File Specs in one of the following ways:

1. Manage them in your SCM, and then during the build, have them pulled to the workspace with the other sources. If you choose this option, you should select the "File" option in the "Upload spec source" or "Download spec source" field and specify the relative path to the File Spec in your workspace.
2. Save the File Spec JSON as part of the job configuration. If you choose this option, you should select the "Job configuration" option in the "Upload spec source" or "Download spec source" field and specify the File Spec JSON content in your workspace in the "File path" field.

You can read the File Spec schema [here](#).

## Legacy Patterns (deprecated)

Legacy patterns are deprecated since version 1.8.0 and will be removed in future releases.

1. The 'Published Artifacts' section lets you specify which artifact files produced by the build will be published to Artifactory. At the end of the build run, artifacts in the build's workspace directory will be selected according to the specified patterns and will be published to Artifactory.

The 'Resolved Artifacts' section lets you specify which artifact files you wish to resolve from Artifactory. Before the build starts, Jenkins will download these artifacts from Artifactory so the build can use them as dependencies.

Optionally, you can also map the target path of deployed/resolved artifacts.

Generic-Artifactory Integration

### Artifactory Configuration

Artifactory server

Target Repository

Override default deployer credentials

Edit Published Artifacts

Edit Resolved Artifacts

Deployment properties

Capture and publish build info

Include all environment variables

Discard old builds from Artifactory (requires Artifactory Pro)

Discard build artifacts

Keep archived artifacts from remote agent