Docs

[User Manual](http://docs.google.com/userguide/userguide.html)

[Guides and Tutorials](https://guides.gradle.org)

[DSL Reference](http://docs.google.com/dsl/)

[Javadoc](http://docs.google.com/javadoc/)

[Release Notes](http://docs.google.com/release-notes.html)

[Forums](https://discuss.gradle.org/)

[Training](https://gradle.org/training/)

[Try Gradle Enterprise](https://gradle.com/enterprise)

[PDF](http://docs.google.com/userguide/userguide.pdf)

* [User Manual Home](http://docs.google.com/userguide/userguide.html)
* [Release Notes](http://docs.google.com/release-notes.html)
* [Installing Gradle](http://docs.google.com/userguide/installation.html)
* [Tutorials](https://guides.gradle.org/)

### Reference

* [Groovy DSL Reference](http://docs.google.com/dsl/)
* [Gradle API Javadoc](http://docs.google.com/javadoc/)
* [Core Plugins](http://docs.google.com/userguide/plugin_reference.html)
* [Gradle & Third-party Tools](http://docs.google.com/userguide/third_party_integration.html)

### Getting Started

* [Creating New Gradle Builds](https://guides.gradle.org/creating-new-gradle-builds/)
* [Creating Build Scans](https://guides.gradle.org/creating-build-scans/)
* [Migrating From Maven](https://guides.gradle.org/migrating-from-maven/)

### Running Gradle Builds

* [Command-Line Interface](http://docs.google.com/userguide/command_line_interface.html)
* [Customizing Execution](#gjdgxs)
  + [Configuring the Build Environment](http://docs.google.com/userguide/build_environment.html)
  + [Configuring the Gradle Daemon](http://docs.google.com/userguide/gradle_daemon.html)
  + [Initialization Scripts](http://docs.google.com/userguide/init_scripts.html)
* [Directory Layout](http://docs.google.com/userguide/directory_layout.html)
* [Executing Multi-Project Builds](http://docs.google.com/userguide/intro_multi_project_builds.html)
* [Gradle Wrapper](http://docs.google.com/userguide/gradle_wrapper.html)
* [Troubleshooting](http://docs.google.com/userguide/troubleshooting.html)
* [Using Build Scans](https://docs.gradle.com/build-scan-plugin)
* [Enabling and Configuring the Build Cache](http://docs.google.com/userguide/build_cache.html)
* [Integrating Separate Gradle Builds (Composite Builds)](http://docs.google.com/userguide/composite_builds.html)

### Authoring Gradle Builds

* [Fundamentals](#30j0zll)
  + [Introducing the Basics of Build Scripts](http://docs.google.com/userguide/tutorial_using_tasks.html)
  + [Working with Tasks](http://docs.google.com/userguide/more_about_tasks.html)
  + [Learning More About Build Scripts](http://docs.google.com/userguide/writing_build_scripts.html)
  + [Working with Files](http://docs.google.com/userguide/working_with_files.html)
  + [Creating Custom Task Types](http://docs.google.com/userguide/custom_tasks.html)
  + [Using Gradle Plugins](http://docs.google.com/userguide/plugins.html)
  + [The Standard Gradle Plugins](http://docs.google.com/userguide/standard_plugins.html)
  + [Understanding the Build Lifecycle](http://docs.google.com/userguide/build_lifecycle.html)
  + [Working with Logging](http://docs.google.com/userguide/logging.html)
  + [Configuring Multi-Project Builds](http://docs.google.com/userguide/multi_project_builds.html)
* [Best Practices](#1fob9te)
  + [Authoring Maintainable Build Scripts](http://docs.google.com/userguide/authoring_maintainable_build_scripts.html)
  + [Organizing Gradle Projects](http://docs.google.com/userguide/organizing_gradle_projects.html)
  + [Optimizing Build Performance](https://guides.gradle.org/performance/)
  + [Using the Build Cache](https://guides.gradle.org/using-build-cache/)
* [Dependency Management](#3znysh7)
  + [Introduction to Dependency Management](http://docs.google.com/userguide/introduction_dependency_management.html)
  + [Dependency Management Terminology](http://docs.google.com/userguide/dependency_management_terminology.html)
  + [Dependency Types](http://docs.google.com/userguide/dependency_types.html)
  + [Repository Types](http://docs.google.com/userguide/repository_types.html)
  + [Declaring Dependencies](http://docs.google.com/userguide/declaring_dependencies.html)
  + [Declaring Repositories](http://docs.google.com/userguide/declaring_repositories.html)
  + [Inspecting Dependencies](http://docs.google.com/userguide/inspecting_dependencies.html)
  + [Managing Dependency Configurations](http://docs.google.com/userguide/managing_dependency_configurations.html)
  + [Managing Transitive Dependencies](http://docs.google.com/userguide/managing_transitive_dependencies.html)
  + [Dependency Locking](http://docs.google.com/userguide/dependency_locking.html)
  + [Troubleshooting Dependency Resolution](http://docs.google.com/userguide/troubleshooting_dependency_resolution.html)
  + [Customizing Dependency Resolution Behavior](http://docs.google.com/userguide/customizing_dependency_resolution_behavior.html)
  + [Dependency Cache Internals](http://docs.google.com/userguide/dependency_cache.html)
  + [Working with Dependencies](http://docs.google.com/userguide/working_with_dependencies.html)
* [Publishing Artifacts](http://docs.google.com/userguide/artifact_management.html)
* [C++ Projects](#2et92p0)
  + [Building Native Software](http://docs.google.com/userguide/native_software.html)
  + [Software Model Concepts](http://docs.google.com/userguide/software_model_concepts.html)
  + [Rule-based Model Configuration](http://docs.google.com/userguide/software_model.html)
  + [Implementing Model Rules in a Plugin](http://docs.google.com/userguide/rule_source.html)
  + [Extending the Software Model](http://docs.google.com/userguide/software_model_extend.html)
* [Java Projects](#tyjcwt)
  + [Building Java & JVM projects](http://docs.google.com/userguide/building_java_projects.html)
  + [Testing Java & JVM projects](http://docs.google.com/userguide/java_testing.html)
* [Advanced Techniques](#3dy6vkm)
  + [Configuring Tasks Lazily](http://docs.google.com/userguide/lazy_configuration.html)
  + [Developing Parallel Tasks](https://guides.gradle.org/using-the-worker-api/)
  + [Testing Your Build with TestKit](http://docs.google.com/userguide/test_kit.html)
  + [Using Ant from Gradle](http://docs.google.com/userguide/ant.html)
* [Sample Gradle builds](#1t3h5sf)
  + [Groovy DSL Samples](https://github.com/gradle/gradle/tree/master/subprojects/docs/src/samples)
  + [Kotlin DSL Samples](https://github.com/gradle/kotlin-dsl/tree/master/samples)

### Extending Gradle

* [Writing Custom Plugins](http://docs.google.com/userguide/custom_plugins.html)
* [Plugin Development Guides](https://gradle.org/guides/?q=Plugin+Development)

[Edit this page](https://github.com/gradle/gradle/edit/master/subprojects/docs/src/docs/userguide/)

# Legacy publishing

Contents

[Introduction](#4d34og8)

[Artifacts and configurations](#2s8eyo1)

[Declaring artifacts](#17dp8vu)

[Publishing artifacts](#3rdcrjn)

[More about project libraries](#26in1rg)

| **✨** | This chapter describes the *original* publishing mechanism available in Gradle 1.0, which has since been superseded by [an alternative model](http://docs.google.com/publishing_overview.html#publishing_overview). The approach detailed in this chapter — based on [Upload](http://docs.google.com/dsl/org.gradle.api.tasks.Upload.html) tasks — should not be used in new builds. We cover it in order to help users work with and update existing builds that use it. |
| --- | --- |

[Introduction](#4d34og8)

This chapter is about how you declare the outgoing artifacts of your project, and how to work with them (e.g. upload them). We define the artifacts of the projects as the files the project provides to the outside world. This might be a library or a ZIP distribution or any other file. A project can publish as many artifacts as it wants.

[Artifacts and configurations](#2s8eyo1)

Like dependencies, artifacts are grouped by configurations. In fact, a configuration can contain both artifacts and dependencies at the same time.

For each configuration in your project, Gradle provides the tasks upload*ConfigurationName* and build*ConfigurationName* when the [base plugin](http://docs.google.com/base_plugin.html#base_plugin) is applied. Execution of these tasks will build or upload the artifacts belonging to the respective configuration.

[This listing](http://docs.google.com/java_plugin.html#tab:configurations) shows the configurations added by the Java plugin. Two of the configurations are relevant for the usage with artifacts. The archives configuration is the standard configuration to assign your artifacts to. The Java plugin automatically assigns the default jar to this configuration. We will talk more about the runtime configuration [further on](#26in1rg). As with dependencies, you can declare as many custom configurations as you like and assign artifacts to them.

[Declaring artifacts](#17dp8vu)

[Archive task artifacts](#lnxbz9)

You can use an archive task to define an artifact:

[Example: Defining an artifact using an archive task](#35nkun2)

**build.gradle**

task myJar(type: Jar)  
  
artifacts {  
 archives myJar  
}

It is important to note that the custom archives you are creating as part of your build are not automatically assigned to any configuration. You have to explicitly do this assignment.

[File artifacts](#1ksv4uv)

You can also use a file to define an artifact:

[Example: Defining an artifact using a file](#44sinio)

**build.gradle**

def someFile = file('build/somefile.txt')  
  
artifacts {  
 archives someFile  
}

Gradle will figure out the properties of the artifact based on the name of the file. You can customize these properties:

[Example: Customizing an artifact](#2jxsxqh)

**build.gradle**

task myTask(type: MyTaskType) {  
 destFile = file('build/somefile.txt')  
}  
  
artifacts {  
 archives(myTask.destFile) {  
 name 'my-artifact'  
 type 'text'  
 builtBy myTask  
 }  
}

There is a map-based syntax for defining an artifact using a file. The map must include a file entry that defines the file. The map may include other artifact properties:

[Example: Map syntax for defining an artifact using a file](#z337ya)

**build.gradle**

task generate(type: MyTaskType) {  
 destFile = file('build/somefile.txt')  
}  
  
artifacts {  
 archives file: generate.destFile, name: 'my-artifact', type: 'text', builtBy: generate  
}

[Publishing artifacts](#3rdcrjn)

We have said that there is a specific upload task for each configuration. Before you can do an upload, you have to configure the upload task and define where to publish the artifacts to. The repositories you have defined (as described in [Declaring Repositories](http://docs.google.com/declaring_repositories.html#declaring_repositories)) are not automatically used for uploading. In fact, some of those repositories only allow downloading artifacts, not uploading. Here is an example of how you can configure the upload task of a configuration:

[Example: Configuration of the upload task](#3j2qqm3)

**build.gradle**

repositories {  
 flatDir {  
 name "fileRepo"  
 dirs "repo"  
 }  
}  
  
uploadArchives {  
 repositories {  
 add project.repositories.fileRepo  
 ivy {  
 credentials {  
 username "username"  
 password "pw"  
 }  
 url "http://repo.mycompany.com"  
 }  
 }  
}

As you can see, you can either use a reference to an existing repository or create a new repository.

If an upload repository is defined with multiple patterns, Gradle must choose a pattern to use for uploading each file. By default, Gradle will upload to the pattern defined by the url parameter, combined with the optional layout parameter. If no url parameter is supplied, then Gradle will use the first defined artifactPattern for uploading, or the first defined ivyPattern for uploading Ivy files, if this is set.

Uploading to a Maven repository is described in [this section](http://docs.google.com/maven_plugin.html#uploading_to_maven_repositories).

[More about project libraries](#26in1rg)

If your project is supposed to be used as a library, you need to define what are the artifacts of this library and what are the dependencies of these artifacts. The Java plugin adds a runtime configuration for this purpose, with the implicit assumption that the runtime dependencies are the dependencies of the artifact you want to publish. Of course this is fully customizable. You can add your own custom configuration or let the existing configurations extend from other configurations. You might have a different group of artifacts which have a different set of dependencies. This mechanism is very powerful and flexible.

If someone wants to use your project as a library, she simply needs to declare which configuration of the dependency to depend on. A Gradle dependency offers the configuration property to declare this. If this is not specified, the default configuration is used (see [Managing Dependency Configurations](http://docs.google.com/managing_dependency_configurations.html#managing_dependency_configurations)). Using your project as a library can either happen from within a multi-project build or by retrieving your project from a repository. In the latter case, an ivy.xml descriptor in the repository is supposed to contain all the necessary information. If you work with Maven repositories you don’t have the flexibility as described above. For how to publish to a Maven repository, see the section [Uploading to Maven repositories](http://docs.google.com/maven_plugin.html#uploading_to_maven_repositories).

Docs

* [User Manual](http://docs.google.com/userguide/userguide.html)
* [DSL Reference](http://docs.google.com/dsl/)
* [Release Notes](http://docs.google.com/release-notes.html)
* [Javadoc](http://docs.google.com/javadoc/)

News

* [Blog](https://blog.gradle.org/)
* [Newsletter](https://newsletter.gradle.com/)
* [Twitter](https://twitter.com/gradle)

Products

* [Build Scans](https://gradle.com/build-scans)
* [Build Cache](https://gradle.com/build-cache)
* [Enterprise Docs](https://gradle.com/enterprise/resources)

Get Help

* [Forums](https://discuss.gradle.org/c/help-discuss)
* [GitHub](https://github.com/gradle/)
* [Training](https://gradle.org/training/)
* [Services](https://gradle.org/services/)

Subscribe for important Gradle updates and news

Subscribe

By entering your email, you agree to our [Terms](https://gradle.org/terms/) and [Privacy Policy](https://gradle.org/privacy/), including receipt of emails. You can unsubscribe at any time.

© [Gradle Inc.](https://gradle.com) 2018 All rights reserved.

[Careers](https://gradle.com/careers) | [Privacy](https://gradle.org/privacy) | [Terms of Service](https://gradle.org/terms) | [Contact](https://gradle.org/contact/)