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/)

# The Base Plugin

Contents

[Usage](#4d34og8)

[Task](#2s8eyo1)

[Dependency management](#17dp8vu)

[Conventions](#3rdcrjn)

The Base Plugin provides some tasks and conventions that are common to most builds and adds a structure to the build that promotes consistency in how they are run. Its most significant contribution is a set of [*lifecycle* tasks](http://docs.google.com/more_about_tasks.html#sec:lifecycle_tasks) that act as an umbrella for the more specific tasks provided by other plugins and build authors.

[Usage](#4d34og8)

[Example: Applying the Base Plugin](#26in1rg)

**build.gradle**

plugins {  
 id 'base'  
}

[Task](#2s8eyo1)

clean — Delete

Deletes the build directory and everything in it, i.e. the path specified by the [Project.getBuildDir()](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildDir) project property.

check — *lifecycle task*

Plugins and build authors should attach their verification tasks, such as ones that run tests, to this lifecycle task using check.dependsOn(*task*).

assemble — *lifecycle task*

Plugins and build authors should attach tasks that produce distributions and other consumable artifacts to this lifecycle task. For example, jar produces the consumable artifact for Java libraries. Attach tasks to this lifecycle task using assemble.dependsOn(*task*).

build — *lifecycle task*

*Depends on*: check, assemble

Intended to build everything, including running all tests, producing the production artifacts and generating documentation. You will probably rarely attach concrete tasks directly to build as assemble and check are typically more appropriate.

build*Configuration* — task rule

Assembles those artifacts attached to the named configuration. For example, buildArchives will execute any task that is required to create any artifact attached to the archives configuration.

upload*Configuration* — task rule

Does the same as build*Configuration*, but also uploads all the artifacts attached to the given configuration.

clean*Task* — task rule

Removes the [defined outputs](http://docs.google.com/more_about_tasks.html#sec:task_inputs_outputs) of a task, e.g. cleanJar will delete the JAR file produced by the jar task of the Java Plugin.

[Dependency management](#17dp8vu)

The Base Plugin adds no [configurations for dependencies](http://docs.google.com/managing_dependency_configurations.html#managing_dependency_configurations), but it does add the following configurations for [artifacts](http://docs.google.com/artifact_management.html#sec:artifacts_and_configurations):

default

A fallback configuration used by consumer projects. Let’s say you have project B with a [project dependency](http://docs.google.com/dependency_types.html#sub:project_dependencies) on project A. Gradle uses some internal logic to determine which of project A’s artifacts and dependencies are added to the specified configuration of project B. If no other factors apply — you don’t need to worry what these are — then Gradle falls back to using everything in project A’s default configuration.

New builds and plugins should not be using the default configuration! It remains for the reason of backwards compatibility.

archives

A standard configuration for the production artifacts of a project. This results in an uploadArchives task for publishing artifacts attached to the archives configuration.

Note that the assemble task generates all artifacts that are attached to the archives configuration.

[Conventions](#3rdcrjn)

The Base Plugin only adds conventions related to the creation of archives, such as ZIPs, TARs and JARs. Specifically, it provides the following project properties that you can set:

archivesBaseName — default: $project.name

Provides the default [AbstractArchiveTask.getBaseName()](http://docs.google.com/dsl/org.gradle.api.tasks.bundling.AbstractArchiveTask.html#org.gradle.api.tasks.bundling.AbstractArchiveTask:baseName) for archive tasks.

distsDirName — default: *distributions*

Default name of the directory in which distribution archives, i.e. non-JARs, are created.

libsDirName — default: *libs*

Default name of the directory in which library archives, i.e. JARs, are created.

The plugin also provides default values for the following properties on any task that extends [AbstractArchiveTask](http://docs.google.com/dsl/org.gradle.api.tasks.bundling.AbstractArchiveTask.html):

destinationDir

Defaults to *$buildDir/$distsDirName* for non-JAR archives and *$buildDir/$libsDirName* for JARs and derivatives of JAR, such as WARs.

version

Defaults to $project.version or 'unspecified' if the project has no version.

baseName

Defaults to $archivesBaseName.

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/)