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 Ear Plugin

Contents

[Usage](#4d34og8)

[Tasks](#2s8eyo1)

[Project layout](#17dp8vu)

[Dependency management](#3rdcrjn)

[Convention properties](#26in1rg)

[Ear](#lnxbz9)

[Customizing](#35nkun2)

[Using custom descriptor file](#1ksv4uv)

The Ear plugin adds support for assembling web application EAR files. It adds a default EAR archive task. It doesn’t require the Java plugin, but for projects that also use the Java plugin it disables the default JAR archive generation.

[Usage](#4d34og8)

To use the Ear plugin, include the following in your build script:

[Example: Using the Ear plugin](#44sinio)

**build.gradle**

apply plugin: 'ear'

[Tasks](#2s8eyo1)

The Ear plugin adds the following tasks to the project.

ear — [Ear](http://docs.google.com/dsl/org.gradle.plugins.ear.Ear.html)

*Depends on*: compile (only if the Java plugin is also applied)

Assembles the application EAR file.

[Dependencies added to other tasks](#2jxsxqh)

The Ear plugin adds the following dependencies to tasks added by the [Base Plugin](http://docs.google.com/base_plugin.html#base_plugin).

assemble

*Depends on*: ear.

[Project layout](#17dp8vu)

.  
└── src  
 └── main  
 └── application (1)

1. Ear resources, such as a META-INF directory

[Dependency management](#3rdcrjn)

The Ear plugin adds two dependency configurations: deploy and earlib. All dependencies in the deploy configuration are placed in the root of the EAR archive, and are *not* transitive. All dependencies in the earlib configuration are placed in the 'lib' directory in the EAR archive and *are* transitive.

[Convention properties](#26in1rg)

appDirName — String

The name of the application source directory, relative to the project directory. *Default value: `src/main/application`*.

libDirName — String

The name of the lib directory inside the generated EAR. *Default value: `lib`*.

deploymentDescriptor — [DeploymentDescriptor](http://docs.google.com/javadoc/org/gradle/plugins/ear/descriptor/DeploymentDescriptor.html)

Metadata to generate a deployment descriptor file, e.g. application.xml. *Default value: A deployment descriptor with sensible defaults named application.xml`*. If this file already exists in the `appDirName/META-INF then the existing file contents will be used and the explicit configuration in the ear.deploymentDescriptor will be ignored.

These properties are provided by a [EarPluginConvention](http://docs.google.com/dsl/org.gradle.plugins.ear.EarPluginConvention.html) convention object.

[Ear](#lnxbz9)

The default behavior of the Ear task is to copy the content of src/main/application to the root of the archive. If your application directory doesn’t contain a META-INF/application.xml deployment descriptor then one will be generated for you.

The [Ear](http://docs.google.com/dsl/org.gradle.plugins.ear.Ear.html) class in the API documentation has additional useful information.

[Customizing](#35nkun2)

Here is an example with the most important customization options:

[Example: Customization of ear plugin](#z337ya)

**build.gradle**

apply plugin: 'ear'  
apply plugin: 'java'  
  
repositories { mavenCentral() }  
  
dependencies {  
 // The following dependencies will be the ear modules and  
 // will be placed in the ear root  
 deploy project(path: ':war', configuration: 'archives')  
  
 // The following dependencies will become ear libs and will  
 // be placed in a dir configured via the libDirName property  
 earlib group: 'log4j', name: 'log4j', version: '1.2.15', ext: 'jar'  
}  
  
ear {  
 appDirName 'src/main/app' // use application metadata found in this folder  
 // put dependent libraries into APP-INF/lib inside the generated EAR  
 libDirName 'APP-INF/lib'  
 deploymentDescriptor { // custom entries for application.xml:  
// fileName = "application.xml" // same as the default value  
// version = "6" // same as the default value  
 applicationName = "customear"  
 initializeInOrder = true  
 displayName = "Custom Ear" // defaults to project.name  
 // defaults to project.description if not set  
 description = "My customized EAR for the Gradle documentation"  
// libraryDirectory = "APP-INF/lib" // not needed, above libDirName setting does this  
// module("my.jar", "java") // won't deploy as my.jar isn't deploy dependency  
// webModule("my.war", "/") // won't deploy as my.war isn't deploy dependency  
 securityRole "admin"  
 securityRole "superadmin"  
 withXml { provider -> // add a custom node to the XML  
 provider.asNode().appendNode("data-source", "my/data/source")  
 }  
 }  
}

You can also use customization options that the [Ear](http://docs.google.com/dsl/org.gradle.plugins.ear.Ear.html) task provides, such as from and metaInf.

[Using custom descriptor file](#1ksv4uv)

You may already have appropriate settings in a application.xml file and want to use that instead of configuring the ear.deploymentDescriptor section of the build script. To accommodate that goal, place the META-INF/application.xml in the right place inside your source folders (see the appDirName property). The file contents will be used and the explicit configuration in the ear.deploymentDescriptor will be ignored.

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/)