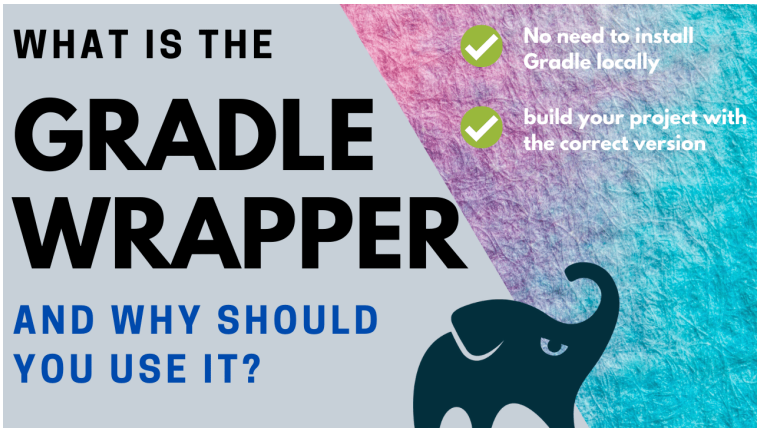
What’s the Gradle Wrapper and Why Use it?



By Tom Gregory Posted on March 14, 2020

Last Updated on April 3, 2022

The **Gradle wrapper** is a script you add to your Gradle project and use to execute your build. The advantages are:

- you don’t need to have Gradle installed on your machine to build the project
- the wrapper guarantees you’ll be using the version of Gradle required by the project

- you can easily update the project to a newer version of Gradle, and push those changes to version control so other team members use the newer version

Now you've got a flavour of what the Gradle wrapper is all about, let's run through some common use cases.

Contents

How do I add the Gradle wrapper to a new project?

How do I add the Gradle wrapper to an existing project?

How do I execute a Gradle build using the wrapper?

How do I see what version of Gradle the wrapper is using in a project?

How do I update the version of Gradle using the wrapper?

Where does the Gradle wrapper store Gradle?

Final info on the Gradle wrapper

Resources

UPDATED in July 2021 to reflect recent Gradle versions

How do I add the Gradle wrapper to a new project?

To initially setup the wrapper, you will need to have Gradle installed on your machine first. Download it from the Gradle website, not forgetting to add the `bin` directory to your `PATH` environment variable.

In an empty directory run `gradle init` to start the Gradle project setup wizard.

```
$ gradle init
Starting a Gradle Daemon (subsequent bui

Select type of project to generate:
  1: basic
  2: application
  3: library
  4: Gradle plugin
Enter selection (default: basic) [1..4]
```

Whatever options you choose, the wrapper will get automatically created. If we inspect the directory, there are some new files.

```
.
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
```

```
│       └── gradle-wrapper.properties
├── gradlew
└── gradlew.bat
```

This includes:

- ***gradle-wrapper.jar*** code required for downloading the correct Gradle version when you run the build
- ***gradle-wrapper.properties*** file to configure the wrapper's properties such as the Gradle version
- ***gradlew*** a shell script for executing the build on Linux
- ***gradlew.bat*** a script for executing the build on Windows

These files should all be added into version control. This way, anyone checking out your project can immediately run a build.

How do I add the Gradle wrapper to an existing project?

This is useful if you have a project which doesn't have a wrapper. Navigate to the project directory and run `gradle wrapper`.

```
$ gradle wrapper
```

```
Starting a Gradle Daemon (subsequent bui

Deprecated Gradle features were used in
Use '--warning-mode all' to show the ind
See https://docs.gradle.org/6.9/userguid

BUILD SUCCESSFUL in 9s
1 actionable task: 1 executed
```

The four Gradle wrapper files described above are automatically created. Remember to check them into version control.

Gradle wrapper version

By default the Gradle wrapper created by running `gradle init` or `gradle wrapper` will use the version of Gradle installed on your machine. Later on we discuss how to update this version.

How do I execute a Gradle build using the wrapper?

That's precisely what ***gradlew*** and ***gradlew.bat*** are for. When you run these scripts a Gradle build will start using the configured version of Gradle.

On **Linux** based operating systems run `./gradlew`.

```
$ ./gradlew

> Task :help

Welcome to Gradle 6.9.

To run a build, run gradlew <task> ...

To see a list of available tasks, run gr

To see a list of command-line options, r

To see more detail about a task, run gra

For troubleshooting, visit https://help.

Deprecated Gradle features were used in
Use '--warning-mode all' to show the ind
See https://docs.gradle.org/6.9/userguid

BUILD SUCCESSFUL in 548ms
1 actionable task: 1 executed
```

And on **Windows** run `gradlew.bat`.

```
c:\workspace\wrapper-test>gradlew.bat

> Task :help

Welcome to Gradle 6.9.

To run a build, run gradlew <task> ...

To see a list of available tasks, run gr

To see a list of command-line options, r
```

```
To see more detail about a task, run gra

For troubleshooting, visit https://help.

Deprecated Gradle features were used in
Use '--warning-mode all' to show the ind
See https://docs.gradle.org/6.9/userguid

BUILD SUCCESSFUL in 1s
1 actionable task: 1 executed
```

We'll be using the Linux version of the wrapper for the examples in the rest of the article, but you can use either depending on your setup.

By default, if you don't pass a task name to the Gradle wrapper script, the *help* task is executed. You can pass a task name using the format `./gradlew <task-name>`.

Running a build

Our next step might be to run a build itself in an existing project with `./gradlew build`:

```
$ ./gradlew build

BUILD SUCCESSFUL in 610ms
```

How do I see what version of Gradle the

wrapper is using in a project?

That's easy, just run `./gradlew --version`:

```
$ ./gradlew --version

------------------------------------------------------------
Gradle 6.9
------------------------------------------------------------

Build time:   2021-05-07 07:28:53 UTC
Revision:     afe2e24ababc7b0213ccffff44

Kotlin:       1.4.20
Groovy:       2.5.12
Ant:          Apache Ant(TM) version 1.1
JVM:          11.0.10 (Ubuntu 11.0.10+9-
OS:           Linux 4.19.128-microsoft-s
```

Or you can also inspect the contents of the ***gradle/wrapper/gradle-wrapper.properties*** file mentioned earlier:

```
$ cat gradle/wrapper/gradle-wrapper.prop
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
distributionUrl=https\://services.gradle
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
```

Right now this project is on version `6.9`. We can get the latest version from the Gradle releases page, which at the time of writing is `7.1.1`. We

better sort that out as we always want to be on the latest tech, right? ✅

How do I update the version of Gradle using the wrapper?

Just run this command:

```
./gradlew wrapper --gradle-version <version-number>
```

So if we wanted to update to version `7.1.1`, we'd run:

```
$ ./gradlew wrapper --gradle-version=7.1

Deprecated Gradle features were used in
Use '--warning-mode all' to show the ind
See https://docs.gradle.org/6.9/userguid

BUILD SUCCESSFUL in 575ms
1 actionable task: 1 executed
```

And just as described in the previous section, we can verify this using `./gradlew --version`:

```
$ ./gradlew --version
Downloading https://services.gradle.org/
.........10%.........20%.........30

------------------------------------------
Gradle 7.1.1
```

```
------------------------------------------

Build time:   2021-07-02 12:16:43 UTC
Revision:     774525a055494e0ece39f522ac

Kotlin:       1.4.31
Groovy:       3.0.7
Ant:          Apache Ant(TM) version 1.1
JVM:          11.0.10 (Ubuntu 11.0.10+9-
OS:           Linux 4.19.128-microsoft-s
```

Awesome, we're on the correct version! 🐘

If you're observant you might notice that at the beginning of the above execution the Gradle wrapper is downloading the newer version of Gradle, which brings us to the next question.

Where does the Gradle wrapper store Gradle?

As discussed, the wrapper ensures you're executing tasks with the correct version of Gradle, without having to have Gradle installed on your machine.

If the wrapper were to download Gradle every time you ran a Gradle task though, that would get *very* annoying very quickly. Consequently, the wrapper caches Gradle versions in the

.gradle/wrapper/dists directory in your user home directory:

```
$ ls -l ~/.gradle/wrapper/dists/
total 20
drwxr-xr-x 3 tom tom 4096 Mar 29 10:59 g
drwxr-xr-x 3 tom tom 4096 Mar 29 08:58 g
drwxr-xr-x 3 tom tom 4096 Jun 26 14:49 g
drwxr-xr-x 3 tom tom 4096 Apr 11 22:25 g
drwxr-xr-x 3 tom tom 4096 Jul  6 15:02 g
```

Here you'll also find any other versions of Gradle you've used before.

Final info on the Gradle wrapper

One final tip. When you're not sure about what tasks you can run in a given context execute `./gradlew tasks` to find out:

```
$ ./gradlew tasks

> Task :tasks

------------------------------------------
Tasks runnable from root project 'wrappe
------------------------------------------

Build Setup tasks
-----------------
init - Initializes a new Gradle build.
wrapper - Generates Gradle wrapper files

Help tasks
----------
```

```
buildEnvironment - Displays all buildscr
dependencies - Displays all dependencies
dependencyInsight - Displays the insight
help - Displays a help message.
javaToolchains - Displays the detected j
outgoingVariants - Displays the outgoing
projects - Displays the sub-projects of
properties - Displays the properties of
tasks - Displays the tasks runnable from

To see all tasks and more detail, run gr

To see more detail about a task, run gra

BUILD SUCCESSFUL in 464ms
1 actionable task: 1 executed
```

To learn more about configuring the Gradle wrapper, see these Gradle docs.