Use –x command-line to exclude tasks

Exp: gradle dist -x test //this will not running the test task

Use –b command-line to select which build to execute

Exp: gradle -q -b subdir/myproject.gradle hello //this will run the task in the specifity gradle file

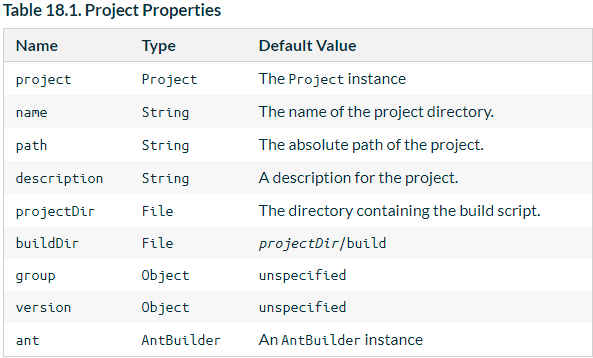
Alternatively, you can use the -p option to specify the project directory to use. For multi-project builds you should use -p option instead of -b option.

Exp: > gradle -q -p subdir hello

use the --rerun-tasks option to force all required tasks to execute

* Any method you call in your build script which is not defined in the build script, is delegated to the Project object.
* Any property you access in your build script, which is not defined in the build script, is delegated to the Project object.

//project 常用的属性



[Section 18.4.2, “Extra properties”](https://docs.gradle.org/current/userguide/writing_build_scripts.html#sec:extra_properties).

[Chapter 43, *Organizing Build Logic*](https://docs.gradle.org/current/userguide/organizing_build_logic.html).

[Section 19.5, “Adding dependencies to a task”](https://docs.gradle.org/current/userguide/more_about_tasks.html#sec:adding_dependencies_to_tasks).

Dependiences

1.//project(String name,Closure c),,用来定位

project('projectA') {

task taskX(dependsOn: ':projectB:taskY') {

doLast {

println 'taskX'

}

}

}

project('projectB') {

task taskY {

doLast {

println 'taskY'

}

}

}

2.

task taskX {

doLast {

println 'taskX'

}

}

task taskY {

doLast {

println 'taskY'

}

}

taskX.dependsOn taskY

复写一个Task

task copy(type: Copy)

task copy(overwrite: true) {

doLast {

println('I am the new one.')

}

}

There are five elements you need to remember: allprojects, subprojects, evaluationDependsOn, evaluationDependsOnChildren and project lib dependencies

# Custom plugin

apply **plugin**:MyPlugin  
myPluginTest{  
 name\_=**'treasure'** age\_=23  
}  
*//myPluginTest.name\_='treasure'  
//myPluginTest.age\_=25***class** MyPlugin **implements** Plugin<Project> {  
  
 @Override  
 **void** apply(Project target) {  
 target.extensions.create(**'myPluginTest'**,MyClassType)  
 target.task(**'helloMyPlugin'**){  
 doLast {  
 println **"\*\*\* Mname:** $target.myPluginTest.name\_ **+Mage:** $target.myPluginTest.age\_**"** println **"\*\*\* Mname:** $target.name**"** }  
 }  
 }  
}  
  
**class** MyClassType{  
 **def** String **name\_**;  
 **def int age\_**;  
}