

Gradle学习系列之四——增量式构建

在[本系列的上篇文章](#)中，我们讲到了如何读懂Gradle的语法，在本篇文章中，我们将讲到增量式地构建项目。

请通过以下方式下载本系列文章的Github示例代码：

```
git clone https://github.com/davenkin/gradle-learning.git
```

如果我们将Gradle的Task看作一个黑盒子，那么我们便可以抽象出输入和输出的概念，一个Task对输入进行操作，然后产生输出。比如，在使用java插件编译源代码时，输入即为Java源文件，输出则为class文件。如果多次执行一个Task时的输入和输出是一样的，那么我们便可以认为这样的Task是没有必要重复执行的。此时，反复执行相同的Task是冗余的，并且是耗时的。

为了解决这样的问题，Gradle引入了增量式构建的概念。在增量式构建中，我们为每个Task定义输入（inputs）和输出（outputs），如果在执行一个Task时，如果它的输入和输出与上一次执行时没有发生变化，那么Gradle便会认为该Task是最新的（UP-TO-DATE），因此Gradle将不予执行。一个Task的inputs和outputs可以是一个或多个文件，可以是文件夹，还可以是Project的某个Property，甚至可以是某个闭包所定义的条件。

每个Task都拥有inputs和outputs属性，他们的类型分别为TaskInputs和TaskOutputs。在下面的例子中，我们展示了这么一种场景：名为combineFileContent的Task从sourceDir目录中读取所有的文件，然后将每个文件的内容合并到destination.txt文件中。让我们先来看看没有定义Task输入和输出的情况：



我翻译的书：



我读过的书：



```
task combineFileContentNonIncremental {
    def sources = fileTree('sourceDir')

    def destination = file('destination.txt')

    doLast {
        destination.withPrintWriter { writer ->
            sources.each {source ->
                writer.println source.text
            }
        }
    }
}
```



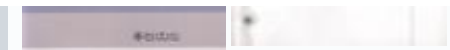
多次执行“gradle combineFileContentNonIncremental”时，整个Task都会反复执行，即便在第一次执行后我们已经得到了所需的结果。如果该combineFileContentNonIncremental是一个繁重的Task，那么多次重复执行势必造成没必要的时间耗费。

这时，我们可以将sources声明为该Task的inputs，而将destination声明为outputs，重新创建一个Task如下：

```
task combineFileContentIncremental {
    def sources = fileTree('sourceDir')
    def destination = file('destination.txt')

    inputs.dir sources
    outputs.file destination

    doLast {
        destination.withPrintWriter { writer ->
            sources.each {source ->
                writer.println source.text
            }
        }
    }
}
```



昵称：无知者云
园龄：4年8个月
粉丝：88
关注：3
[+加关注](#)

我的标签

社会(17)

诗歌(14)

java(13)

gradle(10)

j2ee(9)

java ee(9)

事务(8)

transaction(5)

Ruby(3)

数据库(3)

[更多](#)

随笔档案(65)

2014年7月 (1)

2014年3月 (1)

2014年2月 (1)

2013年11月 (10)

2013年6月 (2)

2013年4月 (1)



相比之下，后一个Task只比前一个Task多了两行代码：

```
inputs.dir sources
outputs.file destination
```

当首次执行combineFileContentIncremental时，Gradle会完整地执行该Task。但是紧接着再执行一次，命令行显示：

```
:combineFileContentIncremental UP-TO-DATE

BUILD SUCCESSFUL

Total time: 2.104 secs
```

我们发现，combineFileContentIncremental被标记为UP-TO-DATE，表示该Task是最新的，Gradle将不予执行。在实际应用中，你将遇到很多这样的情况，因为Gradle的很多插件都引入了增量式构建机制。

如果我们修改了inputs（即sourceDir文件夹）中的任何一个文件或删除掉了destination.txt，当调用“gradle combineFileContentIncremental”时，Gradle又会重新执行，因为此时的Task已经不再是最新了。对于outputs，我们还可以使用upToDateWhen()方法来决定一个Task的outputs是否为最新的，该方法接受一个闭包作为检查条件，感兴趣的读者可以自行了解。

在[下一篇文章](#)中，我们将讲到如何自定义Project的Property。

标签: [gradle](#)

好文要顶

关注我

收藏该文



2013年3月 (1)

2013年2月 (12)

2013年1月 (1)

2012年12月 (2)

2012年11月 (1)

2012年10月 (1)

2012年8月 (1)

2012年7月 (2)

2012年6月 (2)

2012年5月 (3)

2012年4月 (10)

2012年3月 (4)

2012年2月 (7)

2011年12月 (1)

2011年10月 (1)

最新评论

1. Re:毕业两年返校随想

楼主在西电？

--wxyjuly

2. Re:Gradle学习系列之三——读懂Gradle语法

mark

--headchen

3. Re:Gradle学习系列之十一——自定义Plu



关注 - 3

粉丝 - 88

+加关注

« 上一篇 : [Gradle学习系列之三——读懂Gradle语法](#)
» 下一篇 : [Gradle学习系列之五——自定义Property](#)

posted @ 2013-11-12 16:08 无知者云 阅读(5955) 评论(1) 编辑 收藏

3

推荐

0

反对

(请您对文章做出评价)

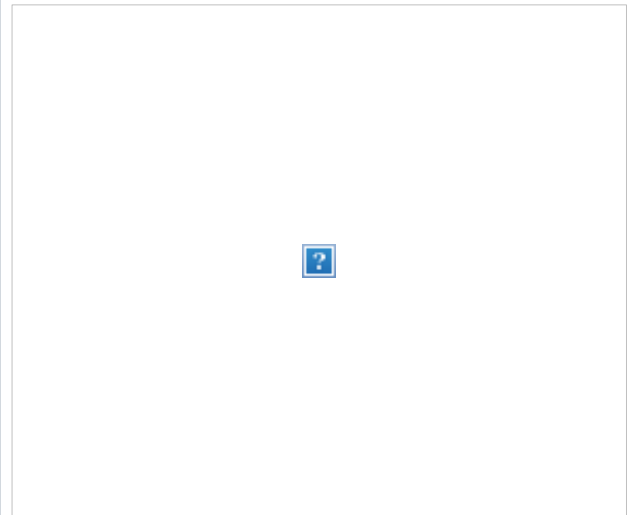
#1楼 2013-11-12 16:32 打一个情

很新潮的样子。。。

支持(0) 反对(0)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
【推荐】融云即时通讯云 - 豆果美食、Faceu等亿级APP都在用



最新IT新闻:
· [Linux Mint 18发布](#)，基于Ubuntu的发行版

gin (本系列完)

DateAndTimePluginExtension 这个里面的timeFormat 是不是没有用，为什么样定义呢？

--灵感仰

4. Re:Gradle学习系列之一——Gradle快速入门

您好，图中这个是否应该是taskA？

--Mr.XuFeng

5. Re:Gradle学习系列之十一——自定义Plugin (本系列完)

博主你真是太棒了！循序渐进通俗易懂。我这两天看gradle看得稀里糊涂的，终于在你这弄明白了。谢谢你！

--无意义

阅读排行榜

1. Gradle学习系列之一——Gradle快速入门(32393)
2. Gradle学习系列之二——创建Task的多种方法(10420)
3. Gradle学习系列之七——依赖管理(9404)
4. 重温大师经典：Martin Fowler的持续集成(9320)
5. Gradle学习系列之三——读懂Gradle语法(9240)

评论排行榜

- 移植.NET Core计划，整合各平台变得更简单了！
 - 太机智了：国外用户利用VR视频逃过盗版审查
 - 微软研发出智能捕蚊器 可预防80多种疾病传播
 - 宇宙竟真的存在多维空间？
- » 更多新闻...



最新知识库文章:

- 编程同写作，写代码只是在码字
 - 遇见程序员男友
 - 设计师的视觉设计五项修炼
 - 我听到过的最精彩的一个软件纠错故事
 - 如何避免软件工程中最昂贵错误的发生
- » 更多知识库文章...

1. 重温大师经典：Martin Fowler的持续集成(12)

2. 5个小时写一个扑克牌游戏——金钩钓鱼(10)

3. Gradle学习系列之二——创建Task的多种方法(6)

4. 用Spring MVC3+Ant+Jenkins+SVN+Tomcat做一个持续集成例子(5)

5. Gradle学习系列之九——自定义Task类型(4)

推荐排行榜

1. Java加载资源文件的两种方法(8)

2. Gradle学习系列之一——Gradle快速入门(7)

3. 重温大师经典：Martin Fowler的持续集成(6)

4. Java事务之一——Java事务的基本问题(5)

5. 用Spring MVC3+Ant+Jenkins+SVN+Tomcat做一个持续集成例子(4)