

JAVA基础：JAR文件包及jar命令详解 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_JAVA\\_E5\\_9F\\_BA\\_E7\\_A1\\_80\\_c104\\_145263.htm](https://www.100test.com/kao_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145263.htm)

常常在网上看到有人询问：如何把 java 程序编译成 .exe 文件。通常回答只有两种，一种是制作一个可执行的 JAR 文件包，然后就可以像.chm 文档一样双击运行了；而另一种是使用 JET 来进行编译。但是 JET 是要用钱买的，而且据说 JET 也不是能把所有的 Java 程序都编译成执行文件，性能也要打些折扣。所以，使用制作可执行 JAR 文件包的方法就是最佳选择了，何况它还能保持 Java 的跨平台特性。下面就来看看什么是 JAR 文件包吧：1. JAR 文件包 JAR 文件就是 Java Archive File，顾名思义，它的应用是与 Java 息息相关的，是 Java 的一种文档格式。JAR 文件非常类似 ZIP 文件准确的说，它就是 ZIP 文件，所以叫它文件包。JAR 文件与 ZIP 文件唯一的区别就是在 JAR 文件的内容中，包含了一个 META-INF/MANIFEST.MF 文件，这个文件是在生成 JAR 文件的时候自动创建的。举个例子，如果我们具有如下目录结构的一些文件：`test -- test -- Test.class` 把它压缩成 ZIP 文件 `test.zip`，则这个 ZIP 文件的内部目录结构为：`test.zip -- test -- Test.class` 如果我们使用 JDK 的 jar 命令把它打成 JAR 文件包 `test.jar`，则这个 JAR 文件的内部目录结构为：`test.jar |-- META-INF | `-- MANIFEST.MF `-- test `--Test.class` 2. 创建可执行的 JAR 文件包 制作一个可执行的 JAR 文件包来发布你的程序是 JAR 文件包最典型的用法。Java 程序是由若干个 .class 文件组成的。这些 .class 文件必须根据它们所属的包不同而分级分目录存放；运行前需要把所有用到的包的根目录指

定给 CLASSPATH 环境变量或者 java 命令的 -cp 参数；运行时还要到控制台下去使用 java 命令来运行，如果需要直接双击运行必须写 Windows 的批处理文件 (.bat) 或者 Linux 的 Shell 程序。因此，许多人说，Java 是一种方便开发者苦了用户的程序设计语言。其实不然，如果开发者能够制作一个可执行的 JAR 文件包交给用户，那么用户使用起来就方便了。在 Windows 下安装 JRE (Java Runtime Environment) 的时候，安装文件会将 .jar 文件映射给 javaw.exe 打开。那么，对于一个可执行的 JAR 文件包，用户只需要双击它就可以运行程序了，和阅读 .chm 文档一样方便 (.chm 文档默认是由 hh.exe 打开的)。那么，现在的关键，就是如何来创建这个可执行的 JAR 文件包。创建可执行的 JAR 文件包，需要使用带 cvfm 参数的 jar 命令，同样以上述 test 目录为例，命令如下：jar cvfm test.jar manifest.mf test 这里 test.jar 和 manifest.mf 两个文件，分别是对应的参数 f 和 m，其重头戏在 manifest.mf。因为要创建可执行的 JAR 文件包，光靠指定一个 manifest.mf 文件是不够的，因为 MANIFEST 是 JAR 文件包的特征，可执行的 JAR 文件包和不可执行的 JAR 文件包都包含 MANIFEST。关键在于可执行 JAR 文件包的 MANIFEST，其内容包含了 Main-Class 一项。这在 MANIFEST 中书写格式如下：Main-Class: 可执行主类全名(包含包名) 例如，假设上例中的 Test.class 是属于 test 包的，而且是可执行的类 (定义了 public static void main(String[]) 方法)，那么这个 manifest.mf 可以编辑如下：Main-Class: test.Test 这个 manifest.mf 可以放在任何位置，也可以是其它的文件名，只需要有 Main-Class: test.Test 一行，且该行以一个回车符结束即可。创建了 manifest.mf 文件之后，我

们的目录结构变为：`== |-- test | `-- Test.class `-- manifest.mf` 这时候，需要到 test 目录的上级目录中去使用 jar 命令来创建 JAR 文件包。也就是在目录树中使用“`==`”表示的那个目录中，使用如下命令：`jar cvfm test.jar manifest.mf test` 之后在“`==`”目录中创建了 test.jar，这个 test.jar 就是执行的 JAR 文件包。运行时只需要使用 `java -jar test.jar` 命令即可。需要注意的是，创建的 JAR 文件包中需要包含完整的、与 Java 程序的包结构对应的目录结构，就像上例一样。而 Main-Class 指定的类，也必须是完整的、包含包路径的类名，如上例的 test.Test；而且在没有打成 JAR 文件包之前可以使用 java 来运行这个类，即在上例中 `java test.Test` 是可以正确运行的（当然要在 CLASSPATH 正确的情况下）。

### 3. jar 命令详解

jar 是随 JDK 安装的，在 JDK 安装目录下的 bin 目录中，Windows 下文件名为 jar.exe，Linux 下文件名为 jar。它的运行需要用到 JDK 安装目录下 lib 目录中的 tools.jar 文件。不过我们除了安装 JDK 什么也不需要，因为 SUN 已经帮我们做好了。我们甚至不需要将 tools.jar 放到 CLASSPATH 中。使用不带任何的 jar 命令我们可以看到 jar 命令的用法如下：`jar {ctxu}[vfm0M] [jar-文件] [manifest-文件] [-C 目录] 文件名 ...` 其中 {ctxu} 是 jar 命令的子命令，每次 jar 命令只能包含 ctxu 中的一个，它们分别表示：`-c` 创建新的 JAR 文件包 `-t` 列出 JAR 文件包的内容列表 `-x` 展开 JAR 文件包的指定文件或者所有文件 `-u` 更新已存在的 JAR 文件包（添加文件到 JAR 文件包中）[vfm0M] 中的选项可以任选，也可以不选，它们是 jar 命令的选项参数 `-v` 生成详细报告并打印到标准输出 `-f` 指定 JAR 文件名，通常这个参数是必须的 `-m` 指定需要包含的 MANIFEST 清单文件 `-0` 只存储

，不压缩，这样产生的 JAR 文件包会比不用该参数产生的体积大，但速度更快 -M 不产生所有项的清单（MANIFEST）文件，此参数会忽略 -m 参数[jar-文件] 即需要生成、查看、更新或者解开的 JAR 文件包，它是 -f 参数的附属参数[manifest-文件] 即 MANIFEST 清单文件，它是 -m 参数的附属参数[-C 目录] 表示转到指定目录下去执行这个 jar 命令的操作。它相当于先使用 cd 命令转该目录下再执行不带 -C 参数的 jar 命令，它只能在创建和更新 JAR 文件包的时候可用。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)