1.Introduction

## 2.Creation of our own JRE only with required modules

# 3.Compressing the size of JRE with compress plugin

# 4. Providing our own name to the application with launcher plugin

JLINK (Java Linker)

1.Introduction:Until 1.8 version to run a small Java program (like Hello World program) also, we should use a bigger JRE which contains all java's inbuilt 4300+ classes. It increases the size of Java Runtime environment(JRE) and Java applications. The Entire JRE to be loaded in to main memory , even application requires only 3 or 4 .classes from JRE. Therefore more space is required in memory. By this, application performance will decrease. Due to this Java is not suitable for IOT devices and Micro Services. (No one invite a bigger Elephant into their small house).

In Java 9, they introduced a permanent solution to reduce the size of Java Runtime Environment, which is nothing but JLINK.

JLINK is Java's new command line tool (which is available in JDK\_HOME\bin) which allows us to create a our own and small JRE with required predefined or/and customized modules.

Jlink also has a list of plugins (like compress,launcher) that will help optimize our solutions.

How to use JLINK: Demo Program

src

|-demoModule

|-module-info.java

|-packA

|-Test.java

module-info.java:

1) module demoModule 3) }

2) {

Test.java:

|  |  |  |
| --- | --- | --- |
| package packA; | | |
| public class Test | | |
|  | { |  |
| public static void main(String[] args) | | |
|  |  | { |
| System.out.println("JLINK Demo To create our own customized & small JRE"); | | |
|  |  | } |
|  | } |  |

## Compilation:

C:\Users\Durga\Desktop>javac --module-source-path src -d out -m demoModule

## Run with default JRE:

C:\Users\Durga\Desktop>java --module-path out -m demoModule/packA.Test

o/p: JLINK Demo To create our own customized & small JRE

## 2.Creation of our own JRE only with required modules:

demoModule requires java.base module. Hence add java.base module to out directory (copy java.base.jmod from jdk-9\jmods to out folder)

out

|-java.base.jmod

|-demoModule

|-module-info.class

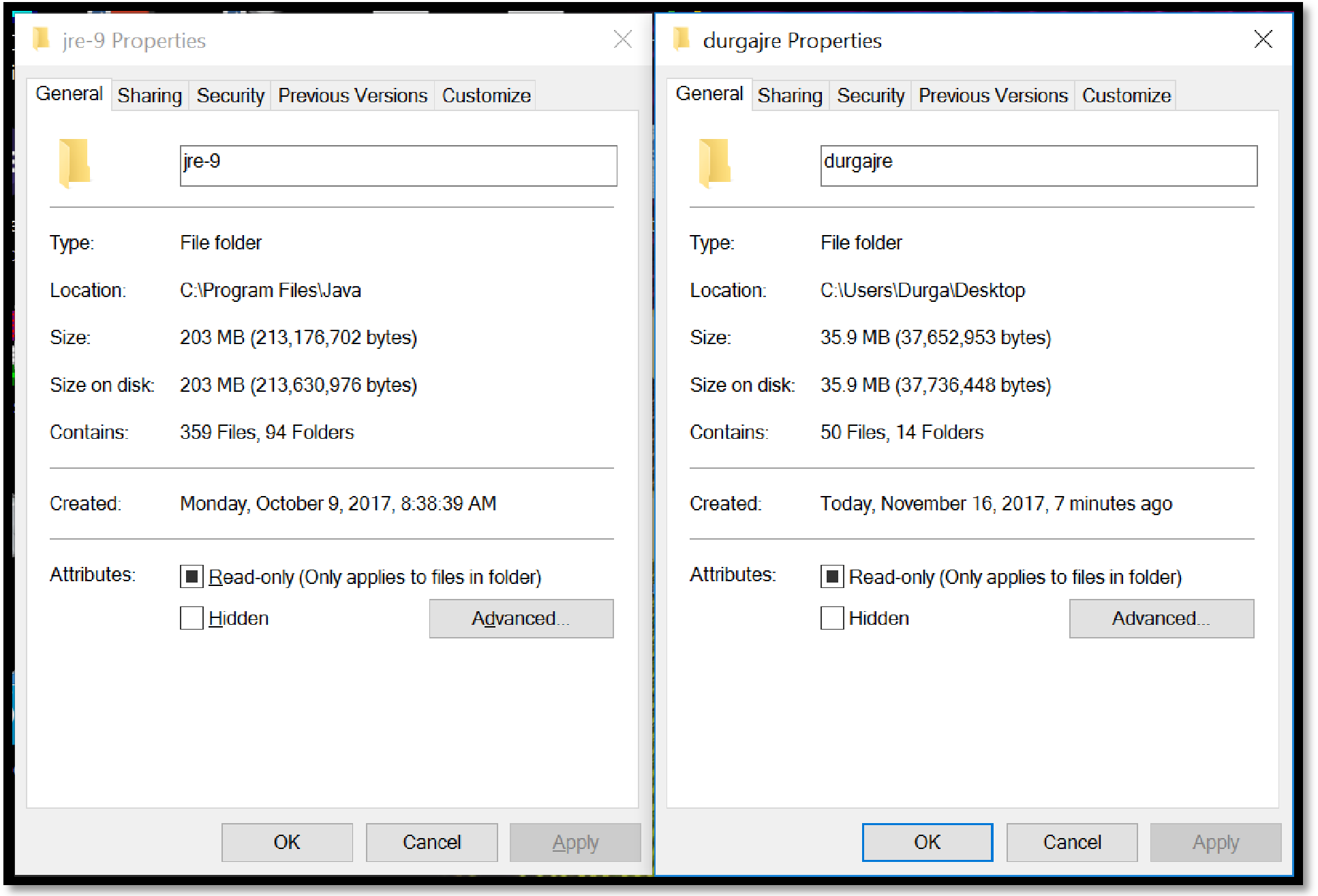
|-packA

|-Test.class

Now we can create our own JRE with JLINK command

C:\Users\Durga\Desktop>jlink --module-path out --add-modules demoModule,java.base –output durgajre

Now observe the size of durgajre is just 35.9MB which is very small when compared with default JRE size 203MB.



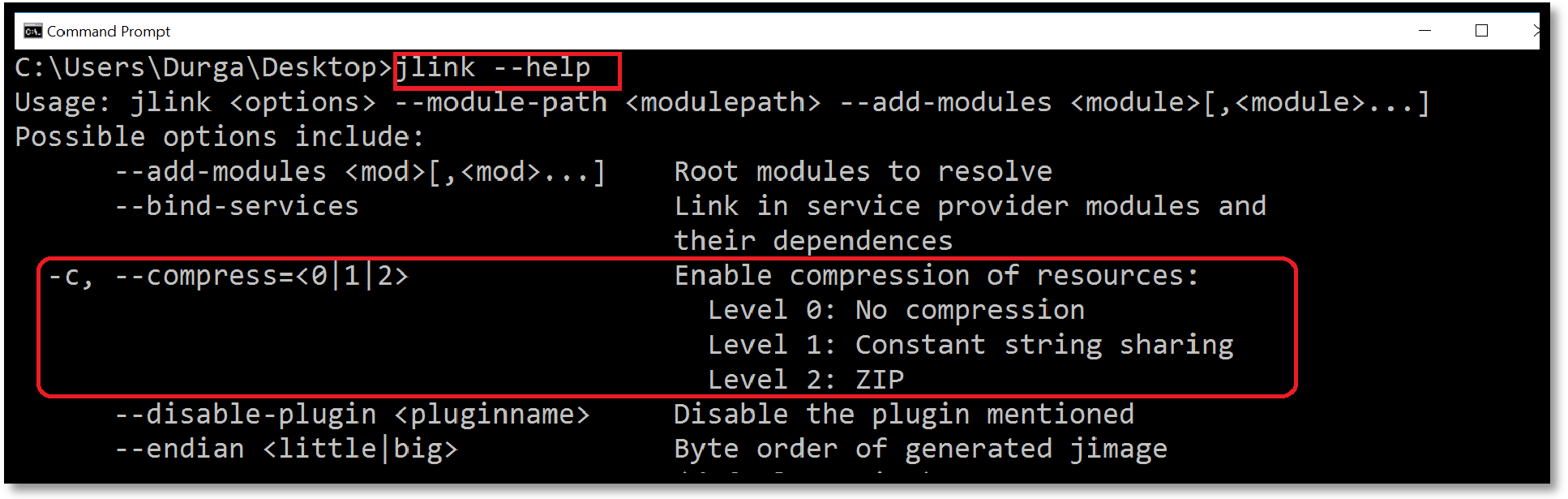
We can run our application with our own custom jre (durgajre) as follows

C:\Users\Durga\Desktop\durgajre\bin>java -m demoModule/packA.Test

o/p: JLINK Demo To create our own customized & small JRE

# 3.Compressing the size of JRE with compress plugin:

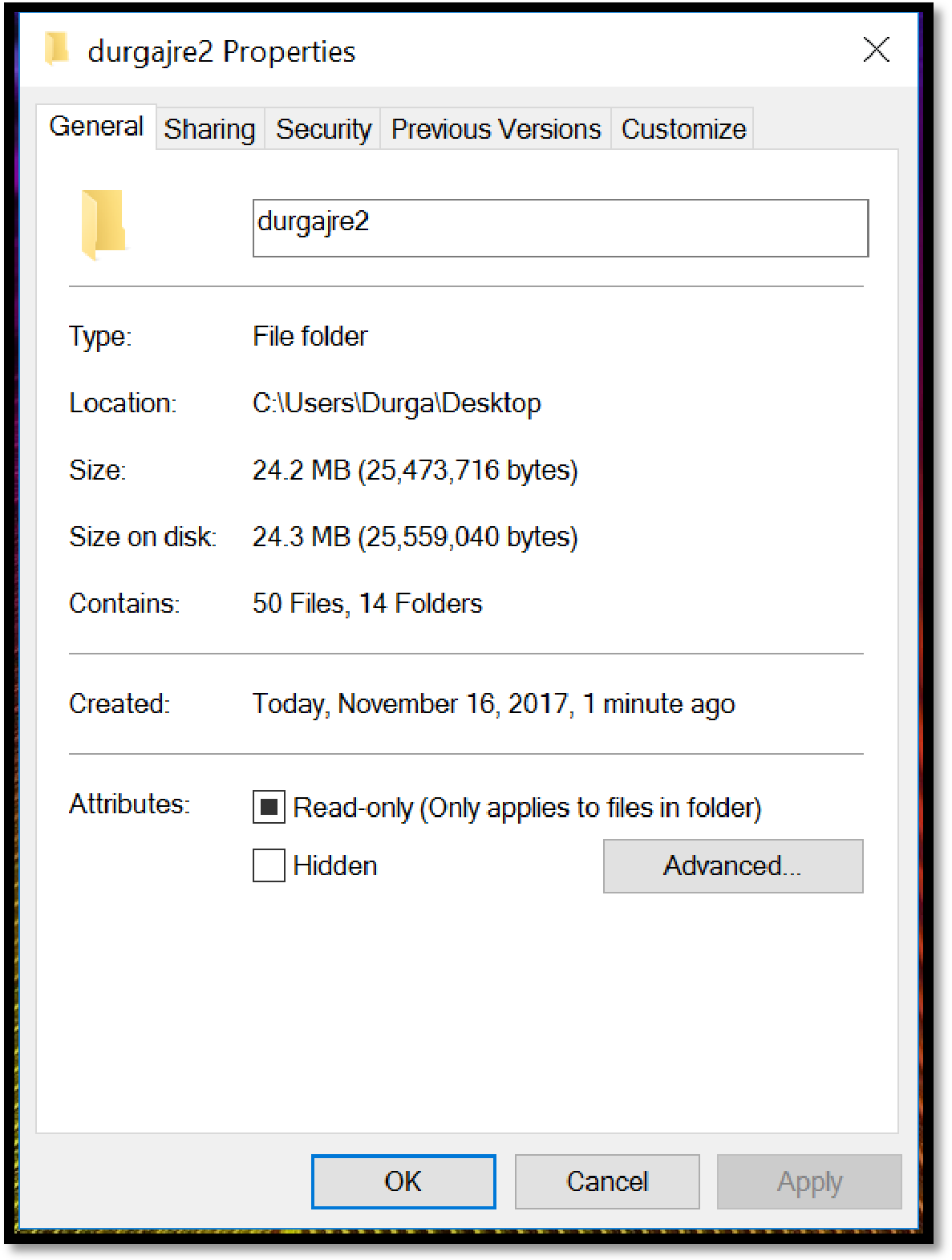
Still we can compress the size of JRE with compress plugin.



C:\Users\Durga\Desktop>jlink --module-path out --add-modules demoModule,java.base -- compress 2 --output durgajre2

C:\Users\Durga\Desktop\durgajre2\bin>java -m demoModule/packA.Test

o/p: JLINK Demo To create our own customized & small JRE



## 4.Providing our own name to the application with launcher plugin:

C:\Users\Durga\Desktop>jlink --module-path out --add-modules demoModule,java.base -- launcher demoapp=demoModule/packA.Test --compress 2 --output durgajre3

Now we can run our application only with the name demoapp C:\Users\Durga\Desktop\durgajre3\bin>demoapp

JLINK Demo To create our own customized & small JRE

If we set the path PATH = C:\Users\Durga\Desktop\durgajre3\bin Then we can run our application from anywhere.

D:\>demoapp E:\>demoapp