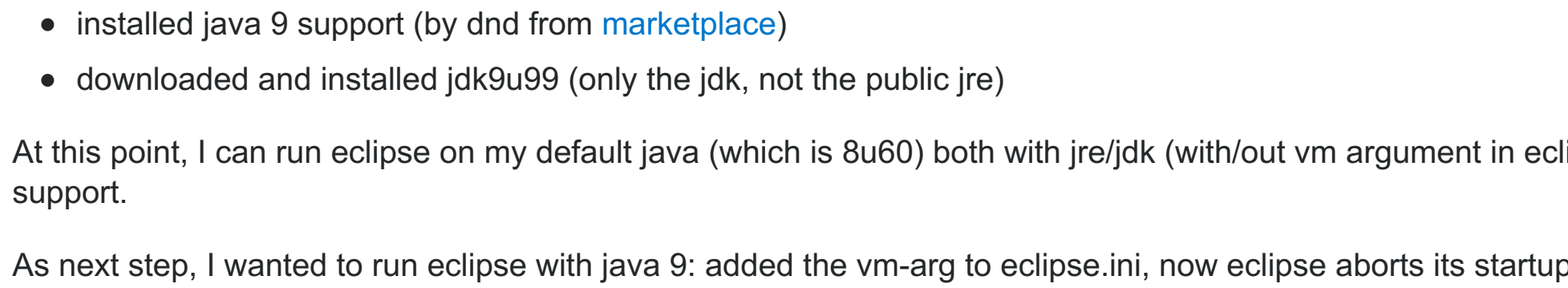


Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers.

Join the world's largest developer community.

Neon: how to run on jdk9?



Just started playing with jdk9 - and kind-of stuck right at the beginning:

- downloaded and extracted eclipse-java-neon-m4a-win32
- installed java 9 support (by dnd from marketplace)
- downloaded and installed jdk9u99 (only the jdk, not the public jre)

At this point, I can run eclipse on my default java (which is 8u60) both with jre/jdk (with/out vm argument in eclipse.ini) and use the java 9 support.

As next step, I wanted to run eclipse with java 9: added the vm-arg to eclipse.ini, now eclipse aborts its startup with a lengthy error log (both ini and log are shown below).

Where's the problem?

My ini:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.200.v20151021-1308.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_1.1.300.v20151013-1129
-product
org.eclipse.epp.package.java.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vm
d:\java\jdk\190_ea\bin\javaw.exe
-vmargs
-Dosgi.requiredJavaVersion=1.8
-Xms256m
-Xmx1024m
```

The top of the error log (it's > 900k)

```
!SESSION 2016-01-22 14:31:55.974 -----
eclipse.buildId=4.6.0.I20151209-2300
java.version=9-ea
java.vendor=Oracle Corporation
BootLoader constants: OS=win32, ARCH=x86, WS=win32, NL=de_DE
Framework arguments: -product org.eclipse.epp.package.java.product
Command-line arguments: -os win32 -ws win32 -arch x86 -product
org.eclipse.epp.package.java.product

!ENTRY org.eclipse.equinox.common 4 0 2016-01-22 14:31:57.113
!MESSAGE FrameworkEvent ERROR
!STACK 0
org.osgi.framework.BundleException: Could not resolve module:
org.eclipse.equinox.common [143]
Unresolved requirement: Require-Capability: osgi.ee; filter:="(&(osgi.ee=JavaSE)
(version=1.7))"
    at org.eclipse.osgi.container.Module.start(Module.java:429)
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.incStartLevel(ModuleC
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.incStartLevel(ModuleC
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.doContainerStartLevel
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.dispatchEvent(ModuleC
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.dispatchEvent(ModuleC
    at
org.eclipse.osgi.framework.eventmgr.EventManager.dispatchEvent(EventManager.java:230)
    at
org.eclipse.osgi.framework.eventmgr.EventManager$EventThread.run(EventManager.java:3

!ENTRY org.eclipse.ant.core 4 0 2016-01-22 14:31:57.175
!MESSAGE FrameworkEvent ERROR
!STACK 0
org.osgi.framework.BundleException: Could not resolve module: org.eclipse.ant.core
[61]
Unresolved requirement: Require-Bundle: org.eclipse.core.variables; bundle-
version="[3.1.0,4.0.0]"
    -> Bundle-SymbolicName: org.eclipse.core.variables; bundle-
version="3.2.800.v20130819-1716"; singleton="true"
    org.eclipse.core.variables [88]
Unresolved requirement: Require-Bundle: org.eclipse.core.runtime; bundle-
version="[3.3.0,4.0.0]"
    -> Bundle-SymbolicName: org.eclipse.core.runtime; bundle-
version="3.12.0.v20151104-0048"; singleton="true"
    org.eclipse.core.runtime [87]
Unresolved requirement: Require-Bundle: org.eclipse.equinox.common;
bundle-version="[3.8.0,4.0.0]"; visibility:="reexport"
    -> Bundle-SymbolicName: org.eclipse.equinox.common; bundle-
version="3.8.0.v20151023-1114"; singleton="true"
    org.eclipse.equinox.common [143]
Unresolved requirement: Require-Capability: osgi.ee;
filter:="(&(osgi.ee=JavaSE)(version=1.7))"
    at org.eclipse.osgi.container.Module.start(Module.java:429)
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.incStartLevel(ModuleC
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.incStartLevel(ModuleC
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.doContainerStartLevel
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.dispatchEvent(ModuleC
    at
org.eclipse.osgi.container.ModuleContainers$ContainerStartLevel.dispatchEvent(ModuleC
    at
org.eclipse.osgi.framework.eventmgr.EventManager.dispatchEvent(EventManager.java:230)
    at
org.eclipse.osgi.framework.eventmgr.EventManager$EventThread.run(EventManager.java:3
```

Update

still no luck with the latest Neon (M6) and 9-ea-113, the logged error has changed, though

```
!SESSION 2016-04-28 16:27:53.344 -----
eclipse.buildId=4.6.0.I20160317-0200
java.version=9-ea
java.vendor=Oracle Corporation
BootLoader constants: OS=win32, ARCH=x86, WS=win32, NL=de_DE
Framework arguments: -product org.eclipse.epp.package.java.product
Command-line arguments: -os win32 -ws win32 -arch x86 -product
org.eclipse.epp.package.java.product

!ENTRY org.eclipse.osgi 4 0 2016-04-28 16:28:10.145
!MESSAGE Application error
!STACK 1
org.eclipse.e4.core.di.InjectionException: java.lang.NoClassDefFoundError:
javax/annotation/PostConstruct
    at
org.eclipse.e4.core.internal.di.InjectorImpl.internalMake(InjectorImpl.java:386)
    at
org.eclipse.e4.core.internal.di.InjectorImpl.make(InjectorImpl.java:294)
    at
org.eclipse.e4.core.contexts.ContextInjectionFactory.make(ContextInjectionFactory.java:165)
    at
org.eclipse.e4.ui.internal.workbench.swt.E4Application.createDefaultHeadlessContext(I
    at
org.eclipse.e4.ui.internal.workbench.swt.E4Application.createDefaultContext(E4Applic
    at
org.eclipse.e4.ui.internal.workbench.swt.E4Application.createE4Workbench(E4Applicati
    at
org.eclipse.ui.internal.Workbench$5.run(Workbench.java:627)
    at
org.eclipse.core.databinding.observable.Realm.runWithDefault(Realm.java:336)
    at
org.eclipse.ui.internal.Workbench.createAndRunWorkbench(Workbench.java:605)
    at
org.eclipse.ui.PlatformUI.createAndRunWorkbench(PlatformUI.java:148)
    at
org.eclipse.ui.internal.ide.application.IDEApplication.start(IDEApplication.java:138)
    at
org.eclipse.equinox.internal.app.EclipseAppHandle.run(EclipseAppHandle.java:196)
    at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.runApplication(EclipseA
    at
org.eclipse.core.runtime.internal.adaptor.EclipseAppLauncher.start(EclipseAppLaunche
    at
org.eclipse.core.runtime.adaptor.EclipseStarter.run(EclipseStarter.java:388)
    at
org.eclipse.core.runtime.adaptor.EclipseStarter.run(EclipseStarter.java:243)
    at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
    at
java.lang.reflect.Method.invoke(Method.java:520)
    at
org.eclipse.equinox.launcher.Main.invokeFramework(Main.java:670)
    at
org.eclipse.equinox.launcher.Main.basicRun(Main.java:609)
    at
org.eclipse.equinox.launcher.Main.run(Main.java:1516)
    at
org.eclipse.equinox.launcher.Main.main(Main.java:1489)
Caused by: java.lang.NoClassDefFoundError: javax/annotation/PostConstruct
    at
org.eclipse.e4.core.internal.di.InjectorImpl.inject(InjectorImpl.java:151)
    at
org.eclipse.e4.core.internal.di.InjectorImpl.internalMake(InjectorImpl.java:375)
    ... 23 more
```

Which looks similar to the one in a bug report. It's marked as fixed in M7.

eclipse java-9 eclipse-neon

edited Apr 29 '16 at 8:42

asked Jan 22 '16 at 13:36

kleopatra
41.4k • 15 • 66 • 135

Should work with download.eclipse.org/eclipse/downloads/drops4/Y20160324-1000 – wzberger Apr 29 '16 at 9:05

@wzberger thanks, will try soon – kleopatra Apr 29 '16 at 9:06

What do you mean by "only the jdk, not the public jre", did you mix the java 9 jdk and java 7/8 jre? That would sound like something that might cause java.lang.NoClassDefFoundError – jan.supol Jun 29 '16 at 9:21

1 @jan.supol yeah sure I mix, won't have a preview as public standard jre :-)) But they should be isolated, I think (and always were in earlier versions) – kleopatra Jun 29 '16 at 14:19

I do not know how you mix it, but javax/annotation/PostConstruct is in rt.jar, and in jdk 9 there is not such a file. – jan.supol Jun 29 '16 at 14:27

4 Answers

I am running Java(TM) SE Runtime Environment (build 9-ea+158) and eclipse Version: Neon.2 Release (4.6.2) - Build id: 20161208-0600

Somehow the solutions with -Djdk.launcher.addmods= and -addmods using values java.se.ee or java.annotations.common did not work.

Below is the solution that worked for me:

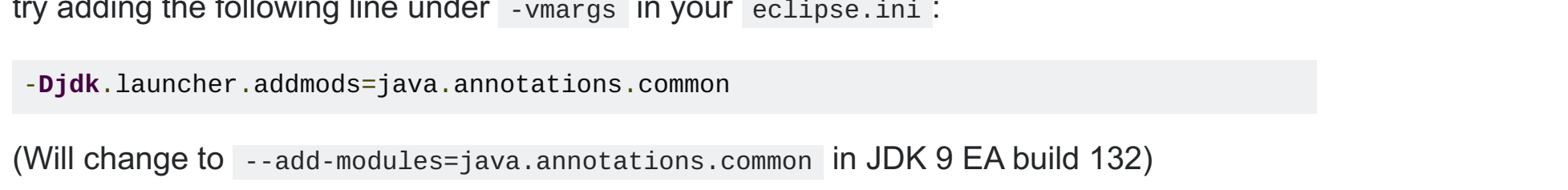
```
--add-modules=java.se.ee
```

Source: <https://issues.jboss.org/browse/JBIDE-22417?focusedCommentId=13310535&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-13310535>

answered Feb 28 at 9:35

Pavan Kumar
1,129 • 5 • 18

Worked for me with build 9-ea+162! thanks – Massoud Afrashteh Apr 15 at 14:56



To get rid of the java.lang.ClassNotFoundException: javax.annotation.PostConstruct issue, try adding the following line under -vmargs in your eclipse.ini:

```
-Djdk.launcher.addmods=java.annotations.common
```

(Will change to --add-modules=java.annotations.common in JDK 9 EA build 132)

See https://bugs.eclipse.org/bugs/show_bug.cgi?id=493761

UPDATE: Since this answer, more things changed, I am currently running Eclipse with JDK 9 b177 and these added vmargs: --add-modules=java.xml.bind,java.xml.ws.annotation

edited Jul 6 at 12:17

answered Aug 16 '16 at 10:10

Klara
1,438 • 10 • 9

This worked for me on build 140 – Daniel Jipa Oct 17 '16 at 17:51

This solution seems not to work on newer builds (build 9+176). However, Pavan Kumar's answer works properly. – Kao Jul 5 at 10:02

Your eclipse.ini is wrong for specifying the JVM. First of all, the -vm option needs to be near the end, right before the --vmargs line. Second, you don't specify the location of the bin folder, rather the java executable (or dll). See the eclipse.ini wiki page for details (and pay careful attention to the details written there).

By the way, where did you get the info to write eclipse.ini the way you did? I'd like to know the source in hopes of correcting that misinformation.

answered Jan 22 '16 at 15:22

E-Riz
23.3k • 5 • 54 • 93

1 thanks - guilty of incorrectly copying from waynebeaton.wordpress.com/2015/10/18/... (that's Linux, so might make a difference to my win7 <g> Now changed the link as suggested - error is unchanged. – kleopatra Jan 22 '16 at 15:41

Bummer, I was hoping (grasping, really) that the other errors were due to not finding the Java 9 VM. Dirk is probably more on track in suspecting the EE resolution is somehow related. – E-Riz Jan 22 '16 at 15:45

I had the same issue. The NoClassDefFoundError shows when you select to build/compile your project with Java 9 JDK. Eclipse does not manage to find a required jar file in root JDK folder, so it fails to start completely. After installing the Java 9 support plugin from the marketplace you may be able to start Eclipse, but still you will see a lot of errors and won't be able to compile a single project. The solution (for now) is to switch to Java 9 JRE in "Installed JREs" under "Project Preferences". This fixed everything for me

Taken from here, you may want to read the full article for more instructions

edited Sep 12 at 21:11

answered Sep 12 at 18:04

filip_j
341 • 5 • 12