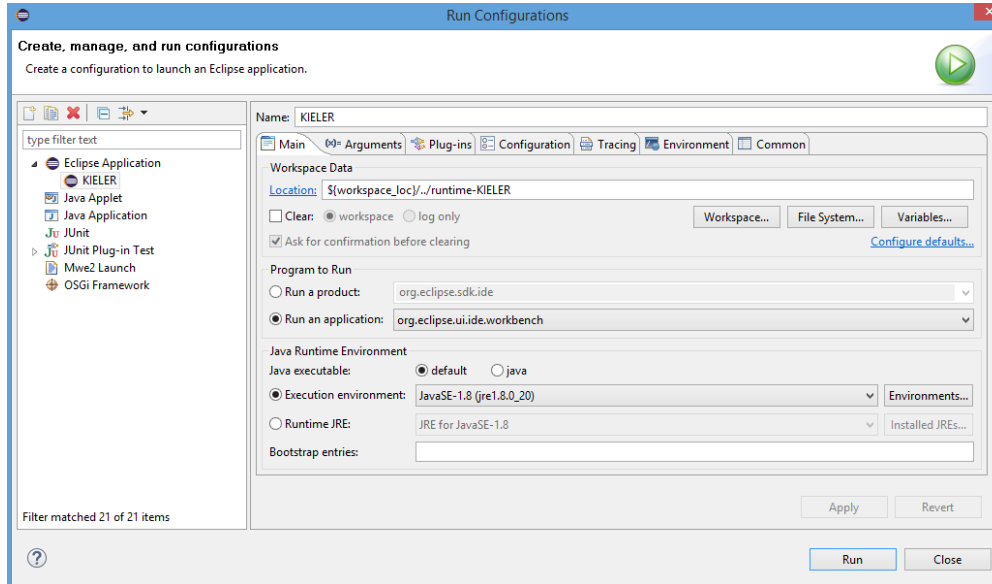


Running KIELER

If you have checked out the necessary KIELER plug-ins, it's time to finally run it for the first time. From the *Run* menu, choose *Run Configurations....* We will use this window to tell Eclipse how to run KIELER.

In the tree view to the left of the *Run Configurations* window, right-click on *Eclipse Application* and select *New*. This will create a new run configuration. Give it a proper name and leave the settings on the main tab unchanged, with one exception: under *Program to Run*, make sure that *Run an application* is selected and choose *org.eclipse.ui.ide.workbench* as the application to run. (This item will only be available if you have checked out the plug-in of the same name.) Your main tab should now look something like this:



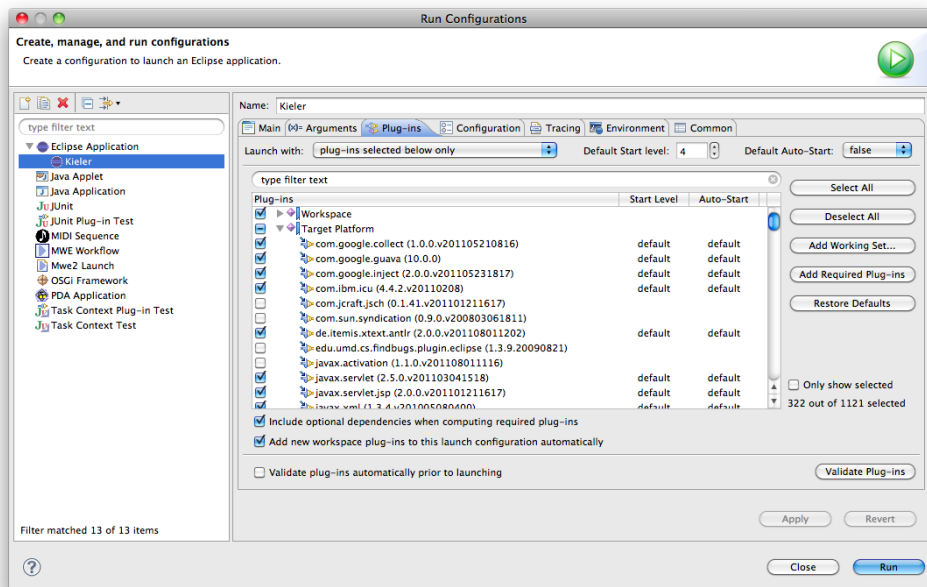
Switch to the *Arguments* tab and append the *VM arguments* with the parameter `-ea`. This will enable assertions in the KIELER code, helping us to fix bugs and stuff.

The most important settings are found in the *Plug-ins* tab. You should select "all workspace and enabled target plug-ins". This will start your the new Eclipse instance with the plug-ins that are currently open in the workspace, plus various plug-ins that are configured for the current target platform. Target platforms are configured under *Preferences > Plug-in Development > Target Platform*.

Deprecated

The Plug-ins tab configuration was different in earlier versions. The old, deprecated guide is as follows:

If we just use the defaults, your KIELER run configuration will include loads of plug-ins that ship with Eclipse that you don't actually need. They would only slow down the startup process and add unnecessary clutter to the menus, so we'll be sure to disable them. To that end, set the *Launch with* setting to *plug-ins selected below only* and click the *Deselect all* button to build your plug-in configuration from scratch. Then, select all the plug-ins from the workspace by checking the *Workspace* item in the tree view. To finish off, click the *Add Required Plug-ins* button. This will add only those Eclipse plug-ins that are really necessary. Make sure that your plug-in configuration is correct by clicking the *Validate Plug-ins* button. Your configuration should now look something like this:



Click the *Apply* button to save your changes, and the *Run* button to finally run KIELER!

By the way: You don't have to go through the *Run Configurations* dialog every time you want to run KIELER. You can conveniently start your new run configuration from the pulldown menu next to the *Debug* or *Run* buttons on the main toolbar.

Normal Mode vs. Debug Mode

During development, it is common to change small things and test the changes. Usually, you would then have to exit KIELER and start it all over again; a lengthy process. But fortunately, help is at hand! Our run configuration also doubles as a debug configuration. When started as such, Eclipse enables a handy little feature called *Hot Swapping*: Whenever you change the source code and save the file, Eclipse tries to merge the changes into the running KIELER instance. This doesn't work for all kinds of code changes. Adding new imports to a class requires a restart, for instance. But it does save significant amounts of time when it does work.

Memory Problems

If your KIELER instance should run out of space, navigate to the *Arguments* tab of the *Run Configurations* dialog and add the following parameters to the *VM arguments*:

```
-Xmn128m
-Xmx768m
```

This will increase the amount of memory available to KIELER. The value in the last line can be set even higher.

JVM-Crash on MacOS

Some users got a harsh crash from the JVM when starting the launch config on Mac. If you are on Mac and your JVM crashes after you see the KIELER splash-screen, then try to add '-nosplash' in Arguments tab of run configuration (details on StackOverflow: <https://stackoverflow.com/questions/53117163/sigsegv-signal-while-executing-eclipse-application-run-configuration-for-my-xt>).

JVM-Crash on MacOS that is fixed by '-nosplash' argument:

```
#
# A fatal error has been detected by the Java Runtime Environment:
#
# SIGSEGV (0xb) at pc=0x00007fff58cf4a29, pid=77104, tid=0x0000000000000307
#
# JRE version: Java(TM) SE Runtime Environment (8.0_144-b01) (build 1.8.0_144-b01)
# Java VM: Java HotSpot(TM) 64-Bit Server VM (25.144-b01 mixed mode bsd-amd64 compressed oops)
# Problematic frame:
# C [libobjc.A.dylib+0x7a29] objc_msgSend+0x29
# ...
```