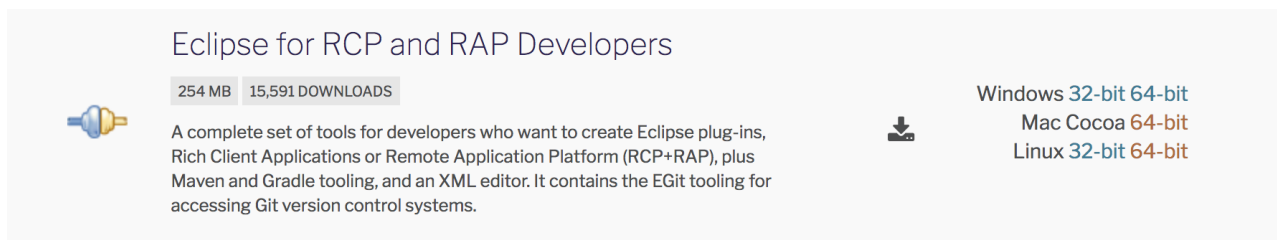


# TextGridLab Development Environment „bauen“

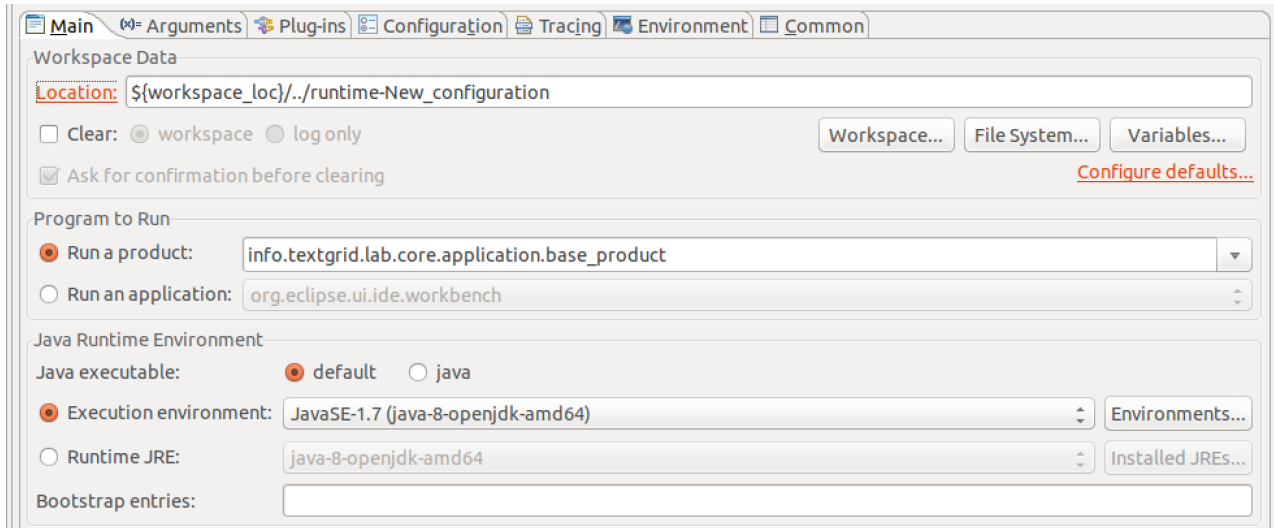
Stand: 09.08 2018

1. Java JDK herunterladen (<https://java.com/de/download/>), getestet mit V 8\_171) und installieren.
2. Eclipse Neon Installer (Eclipse Version 4.6) herunterladen (URL <https://www.eclipse.org/neon/>). Auf „Download“ klicken.
3. Jetzt „Eclipse for RCP and RAP Developer“ in der 64-Bit Version auswählen (maven bzw. m2e ist hier bereits Teil des Pakets) und herunterladen.



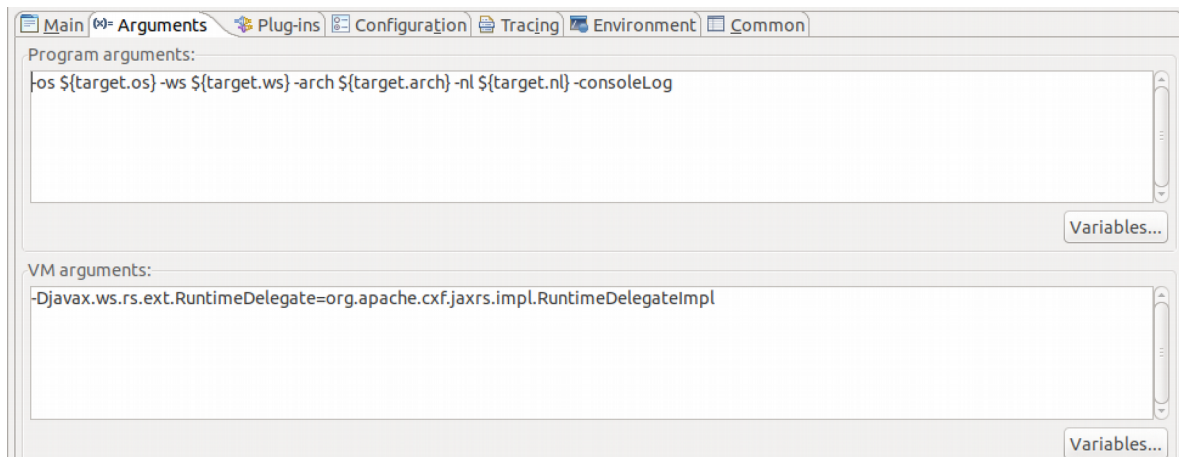
4. Eclipse installieren/entpacken und anschließend starten.
5. In Eclipse unter Window → Preferences → Maven → Discovery → Open Catalog → folgende Adapter suchen und auswählen:
  - m2e-jdt-compile
  - mavenarchiver pomproperties
  - Tycho Configurator
  - m2e-egit→ anschließend auf Finish klicken. Nach kurzer Ladezeit auf → Next → Next  
→ I accept... → Finish → OK  
→ Eclipse neu starten
6. Neue Target Platform anlegen  
In Eclipse unter Window → Preferences → Plug-In Development → Target Platform  
→ Add → Nothing: Start...empty... → Next  
→ Name vergeben (z.B. TGLab TP )  
→ Add → Software Site  
→ Add → Name eingeben (z.B. TGLab Nightly)  
→ bei Location z.B. die Nightly-Site eintragen (URL: <http://www.textgridlab.org/updates/nightly>)
7. TextGridLab SDK auswählen (oder je nachdem weiter separieren)  
→ finish

- die neue TargetPlatform auswählen und auf Apply klicken
- 8. Einstellungen → Java → Compiler compliance level → 1.7 auswählen → Apply
- 9. Run → Run Configurations → Eclipse Application → New Launch Configuration
  - oben Namen vergeben
  - Im Tab Main → Run a product → auswählen:  
 info.textgrid.lab.core.application.base\_product



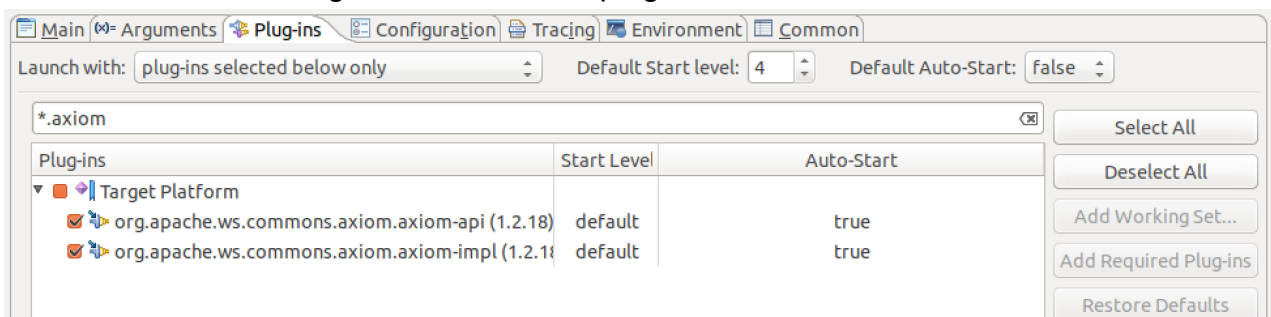
→ Im Tab Arguments → VM Arguments → folgende Zeile einfügen (ohne Leerzeilen!):

`-Djavax.ws.rs.ext.RuntimeDelegate=org.apache.cxf.jaxrs.impl.RuntimeDelegateImpl`



→ Im Tab Plug-ins → Launch with → „plug-ins selected below only“ auswählen  
 → „\*.axiom“ ins Suchfeld eingeben

→ die beiden gefundenen axiom plugins auf autostart: true stellen



10. Apply → Run → TextGridLab sollte starten.

---

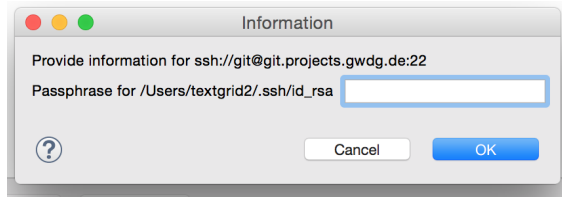
## GIT-Repository clonen

---

1. (Falls nötig) RSA-Schlüssel generieren und wie gewohnt ins `.ssh` Verzeichnis im Benutzerordner ablegen.
2. (Falls nötig) Im `.ssh` Verzeichnis zusätzlich die Datei „config“ mit folgendem Inhalt anlegen (ggf. `id_rsa` jeweils durch den Namen des erzeugten Keys ändern, falls nicht Standardname):  

```
host git.projects.gwdg.de
  HostName git.projects.gwdg.de
  IdentityFile ~/.ssh/id_rsa
  User git
```
3. (Falls nötig) In der Konsole zum `.ssh` Verzeichnis navigieren, dann folgende Befehle eingeben (ggf. siehe Punkt 2):
  - `sudo chown Benutzername:Benutzername *`
  - `sudo chmod 600 id_rsa`
  - `sudo chmod 640 id_rsa.pub`
  - `sudo chmod 640 config`
4. Unter [https://projects.gwdg.de/my/access\\_token](https://projects.gwdg.de/my/access_token) mit Dariah Account einloggen und den RSA Access token eintragen.
5. In Eclipse  
→ Window → Show Views → Other → Git → Git Repositories → OK
6. Clone a Git Repository → Clone URI → next
7. Im Browser öffnen: <https://projects.gwdg.de/projects/textgrid-laboratory>
8. Ggf. Unterprojekt(e) auswählen z.B. Lab core  
→ dann auf der linken Seite „Repository“ auswählen  
→ Feld oben rechts auf „SSH“ stellen und URL rechts daneben kopieren  
→ URL in das Feld Uri in Eclipse kopieren → next

9. Passwort für RSA-Schlüssel eingeben



→ Zugang zu <https://projects.gwdg.de/projects/textgrid-laboratory> muss allerdings vorhanden sein, ebenso wie Schreib- und Leserechte. Wenn dies nicht der Fall sein sollte, am besten an Stefan Funk wenden, der kann dich dann als Entwickler eintragen (auch in den Subprojekten!).

10. → Deselect all

→ Developer-(Branch)-Version auswählen und herunterladen. (aktuellster Stand)

→ Next

→ „Clone Submodules“ aktivieren

→ Finish

11. Anschließend auf dem geklonten Repository die rechte Maustaste drücken

→ “Import Maven Projects..“

→ alle Pakete in den Project Explorer übernehmen.

