MutIcube Install GuideBook :

Author : Baptiste Lafabregue

Version : 0.9

Date : 21/08/2018

Requirements :

You will need java jre 1.8 or above (jdk is recommended) and git installed.
Note that all commands are designed to work with unix bash.

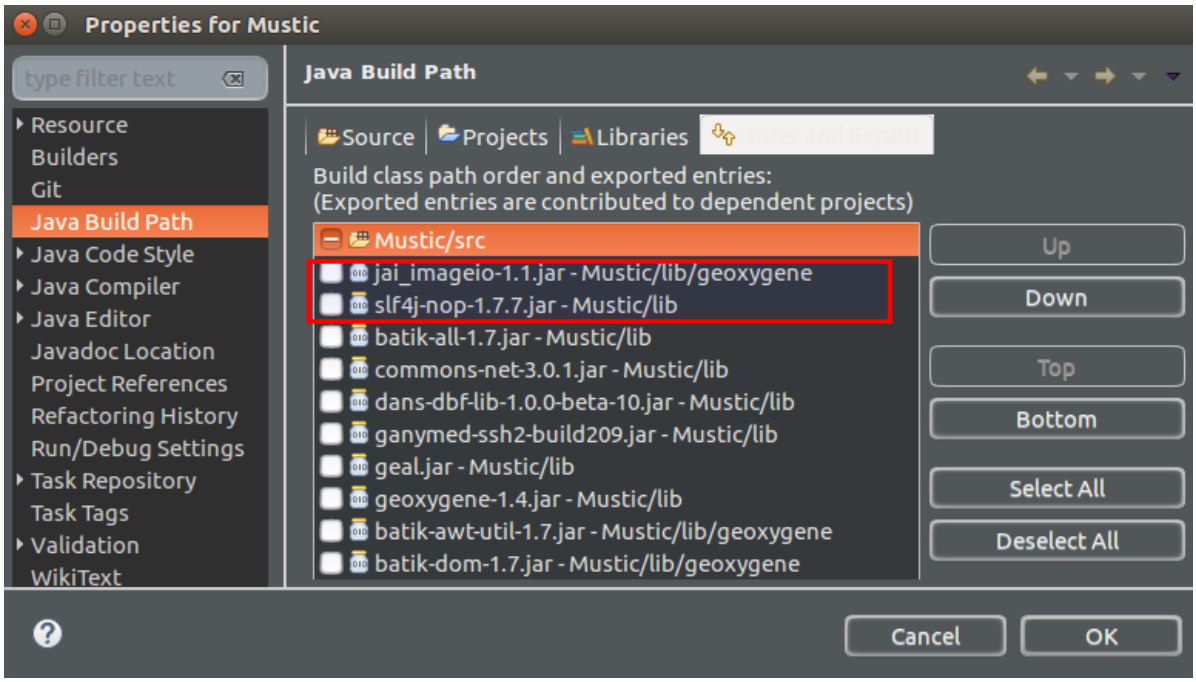
Installation :

Create a MutIcube directory and clone the four git project in it :
mkdir MultIcube
cd MultIcube
git clone https://icube-forge.unistra.fr/lafabregue/MultiCubeTools.git
git clone https://icube-forge.unistra.fr/lafabregue/JCL.git
git clone https://icube-forge.unistra.fr/lafabregue/JSL.git
git clone https://icube-forge.unistra.fr/lafabregue/Mustic.git

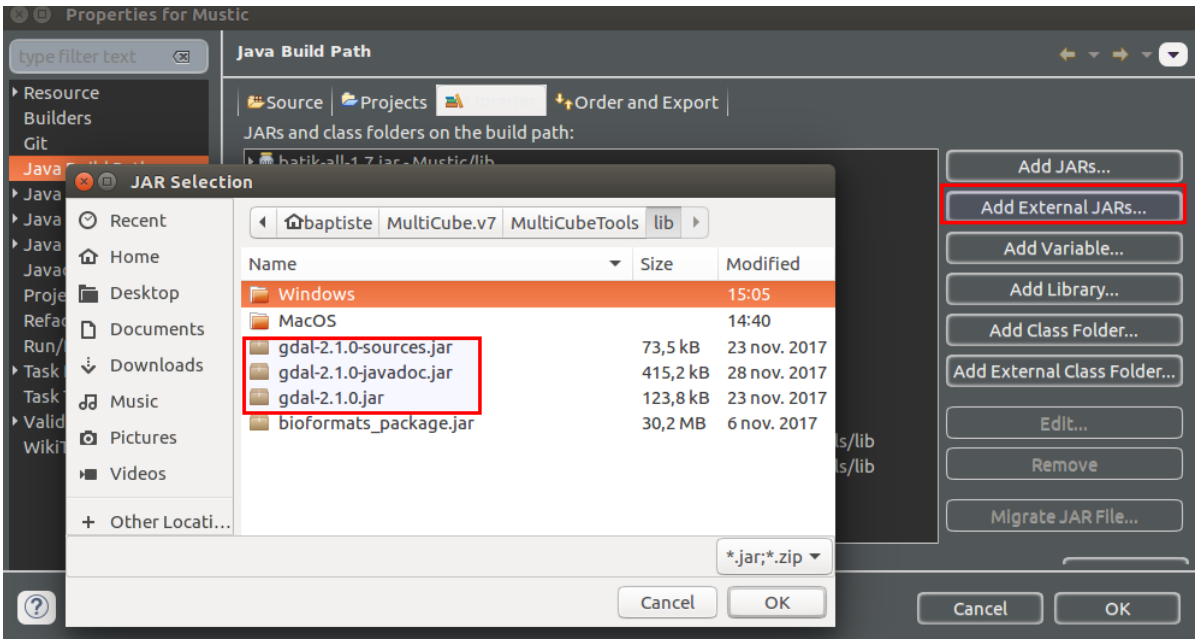
Open your favourite java IDE but it is recommended to use eclipse. The following instruction will be based on eclipse interface.

- Change your workspace to the MultiCube directory
-
- Import the previous four projects into your IDE :
  - Import > General > Projects from Folder or Archive
  - Choose for each four projects the corresponding folder

- For each projet change its build path :
  - JCL : add project MultiCubeTools
  - JSL : add project MultiCubeTools
  - Mustic :
    - add project MultiCubeTools
    - add project JCL
    - add project JSL
    - add the directory « resources » as Class Folder in Libraries build path
    - change the Order of importation of ressources in the Order and Export tab to have the two highlighted libraries on top :



  - Add gdal external jars to the Mustic project :



- Run the mustic project :
  Run > Java Application >
  choose «MultiCube - mustic » among the different Main class
  Use the Mustic/src/mustic/MultiCube.java if you don't use eclipse

- Change the Run Configuration :
  - add the two following Environment variables :
    - Name : GDAL_DATA
      Value : ../MultiCubeTools/share/
      for linux only :
      - Name : LD_LIBRARY_PATH
        Value : ../MultiCubeTools/lib:./lib

      for Windows and MacOS :
      - add the following VM argument : -Djava.library.path=../MultiCubeTools/lib:./lib
  - it is also recommended to change the memory allocated, by adding the VM arguments :
    -Xms<*your_min_memory_to_use*>m -Xmx<*your_max_memory_available*>m

On x64 unix platfrom you should normally be able to launch Mustic.
But if you import a picture and still have an error, or you want to use the project on MacOS Windows, or x32 Unix you will have to change the dynamic libraries :

- For Windows : copy paste content of MultiCubeTools/lib/Windows/xx, xx beeing 64 or 32 depending of your architecture
- For MacOS : copy paste content of MultiCubeTools/lib/MacOS

If it does not fix your problem, you need to get the gdal binaries with java bidings :

- for Windows, you can refer to :
  https://trac.osgeo.org/gdal/wiki/DownloadingGdalBinaries
- for Linux, you need to compile GDAL with the –with-java option, make the gdal-*/swig/java sub-projet, and copy the resulting .so files and jar. For more details, you can refer to :
  https://github.com/VertNet/reproject/wiki/GDAL-Java-Bindings (note that now there only one .so file generated)
- for MacOS, the homebrew build no longer support the –with-java option, you need to compile it yourself with the –with-java option, make the gdal-*/swig/java sub-projet, and copy the resulting .so files and jar. For more details, you can refer to :
  http://www.ecgs.lu/gilles/enabling-gdal-java-binding-for-geoserver-on-macos/