

TAB2XML User Manual

For Version 1.0.0

2021 April 13

Contents

1	Introduction and Purpose	2
2	System Requirements	2
3	How to Install TAB2XML	2
3.1	Eclipse	2
3.2	Commandline	8
4	How to Use TAB2XML	9
4.1	Convert Text Tab	9
4.2	Editing Text Tabs	11
4.2.1	Editing Measures	11
4.2.2	Editing Metadata	12
4.3	Other Use Cases	12
4.4	How to Use in Code	12
5	Supported Tabs	13
5.1	Guitar/Bass	13
5.1.1	supported features	13
5.2	Drum	13
5.2.1	supported drum types notation	13

1 Introduction and Purpose

The TAB2XML system can be used to convert text tablature into MusicXML. This document outlines how to install and use the system in many use cases.

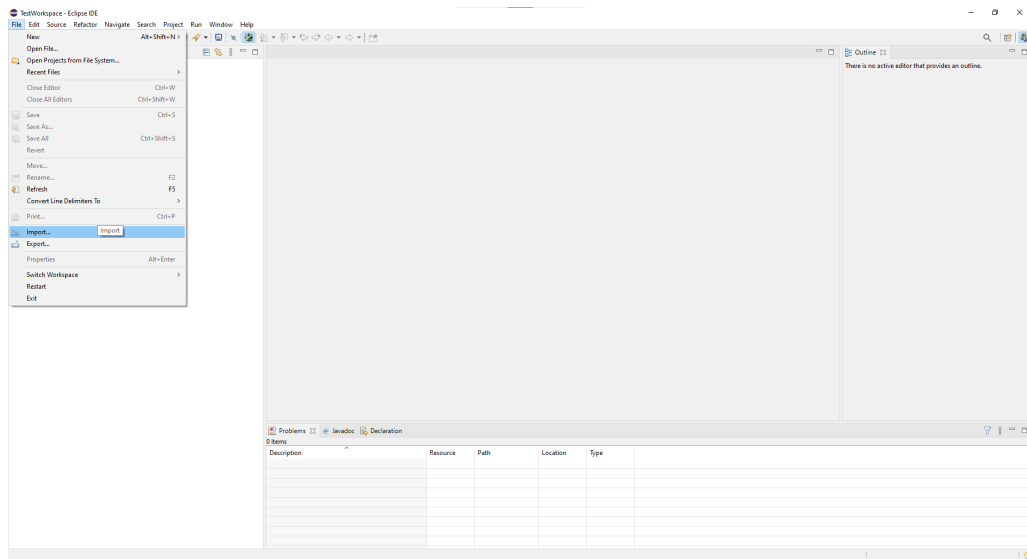
2 System Requirements

- Works on all major operating systems
- Java version 11-15 required

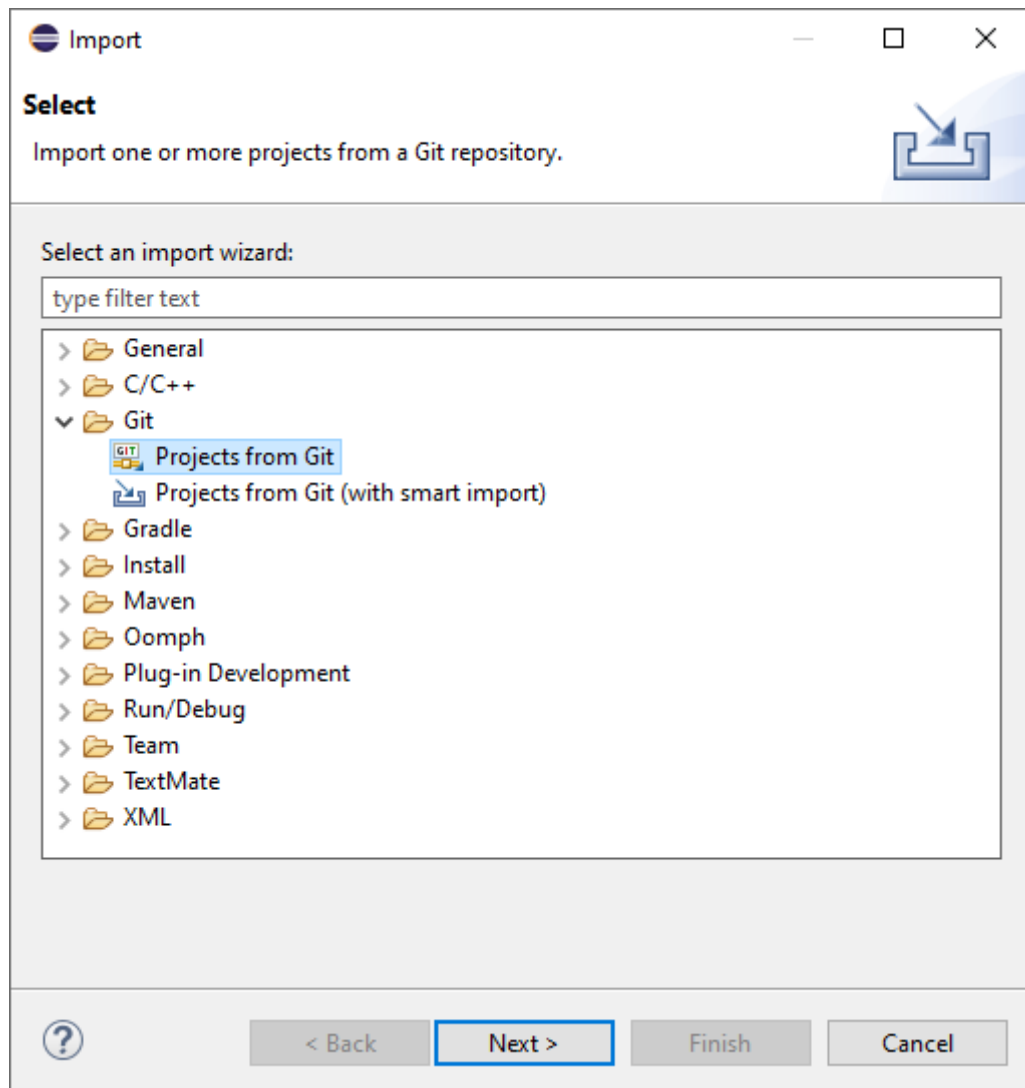
3 How to Install TAB2XML

3.1 Eclipse

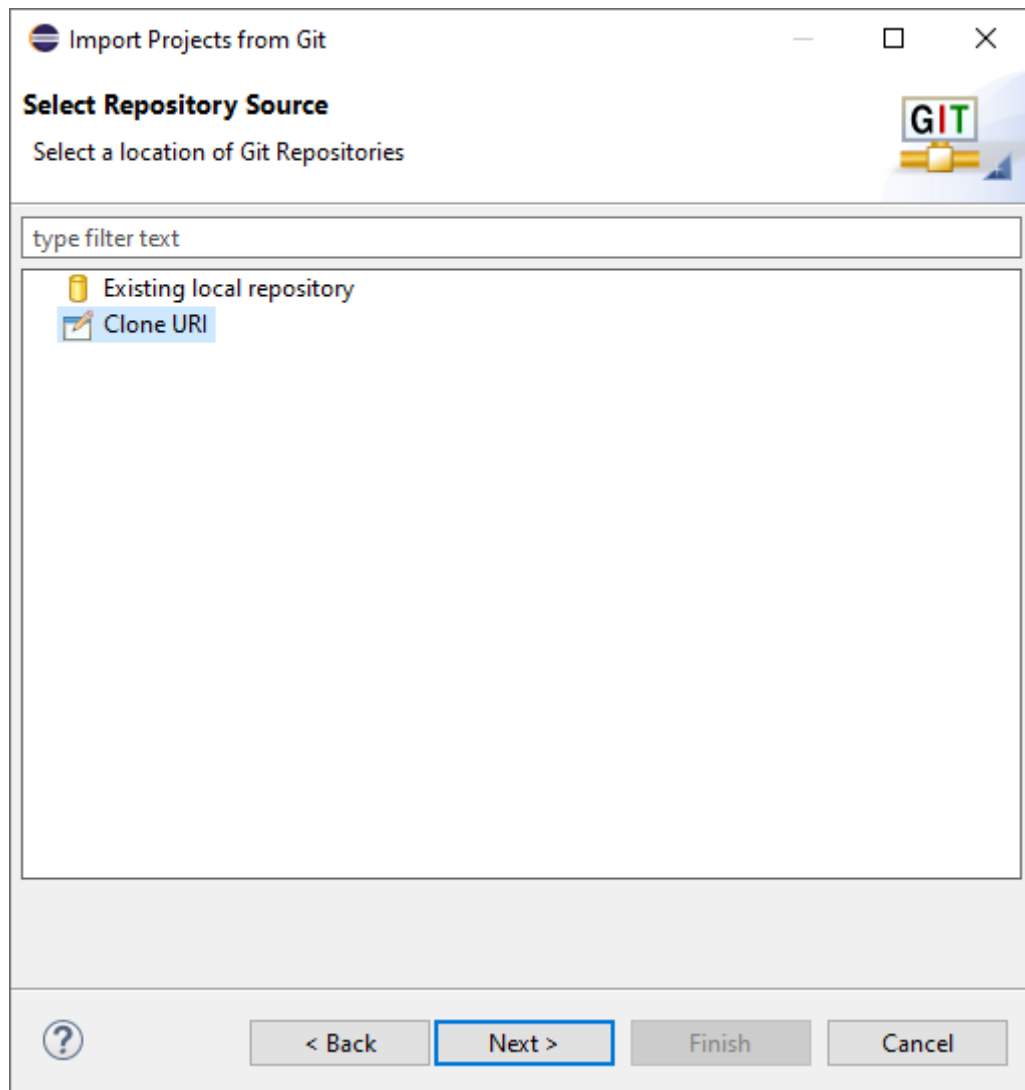
1. Open Eclipse, and press **File > Import** in the menus.



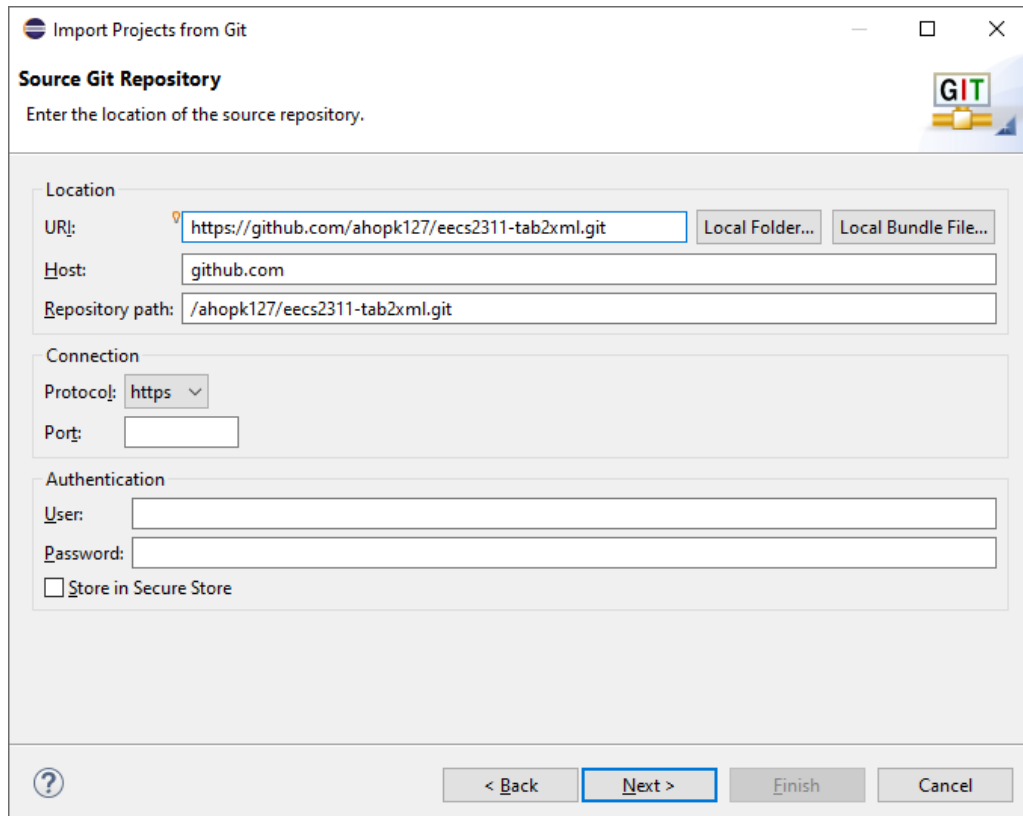
2. In the window that opens, select "Projects from Git", in the folder called "Git". Then, click "Next"



3. Click "Clone URI" then click "Next".



4. Enter "`https://github.com/ahopk127/eecs2311-tab2xml.git`" in the first field ("URI"), then click "Next". No authentication is required.



Import Projects from Git

Source Git Repository
Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

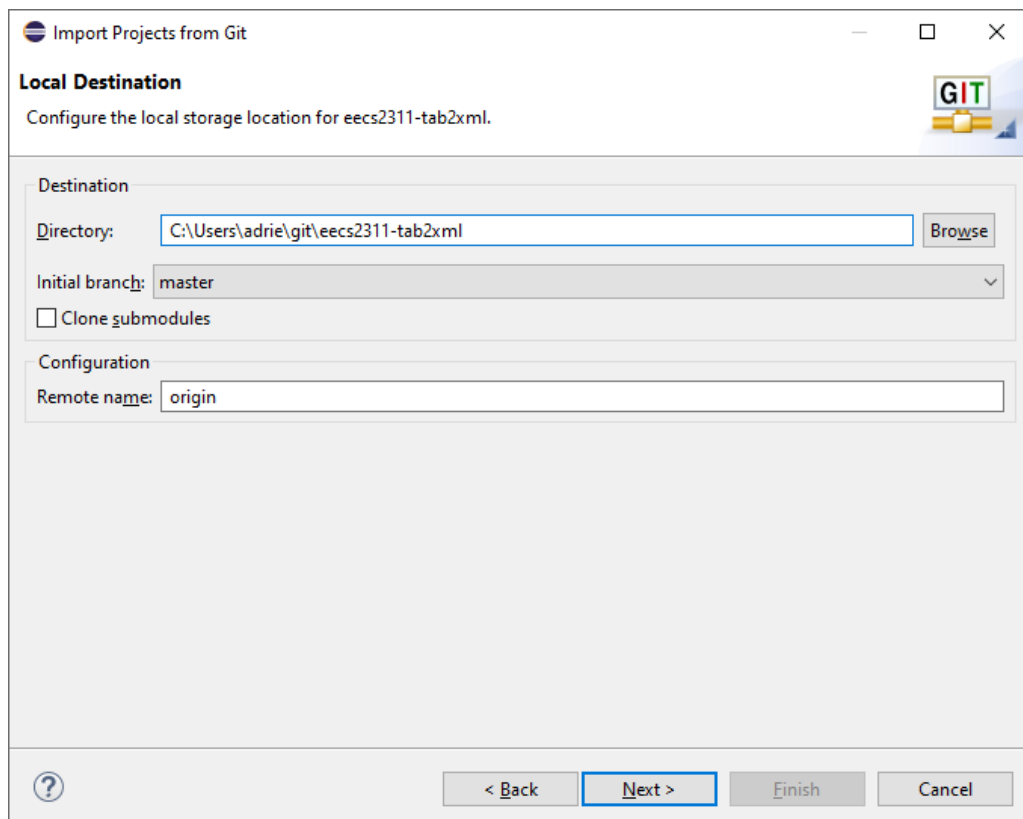
Authentication

User:

Password:

☐ Store in Secure Store

- Choose where you want the program to be saved on your computer (or just use the default location), then click "Next".



Import Projects from Git

Local Destination
Configure the local storage location for eeecs2311-tab2xml.

Destination

Directory:

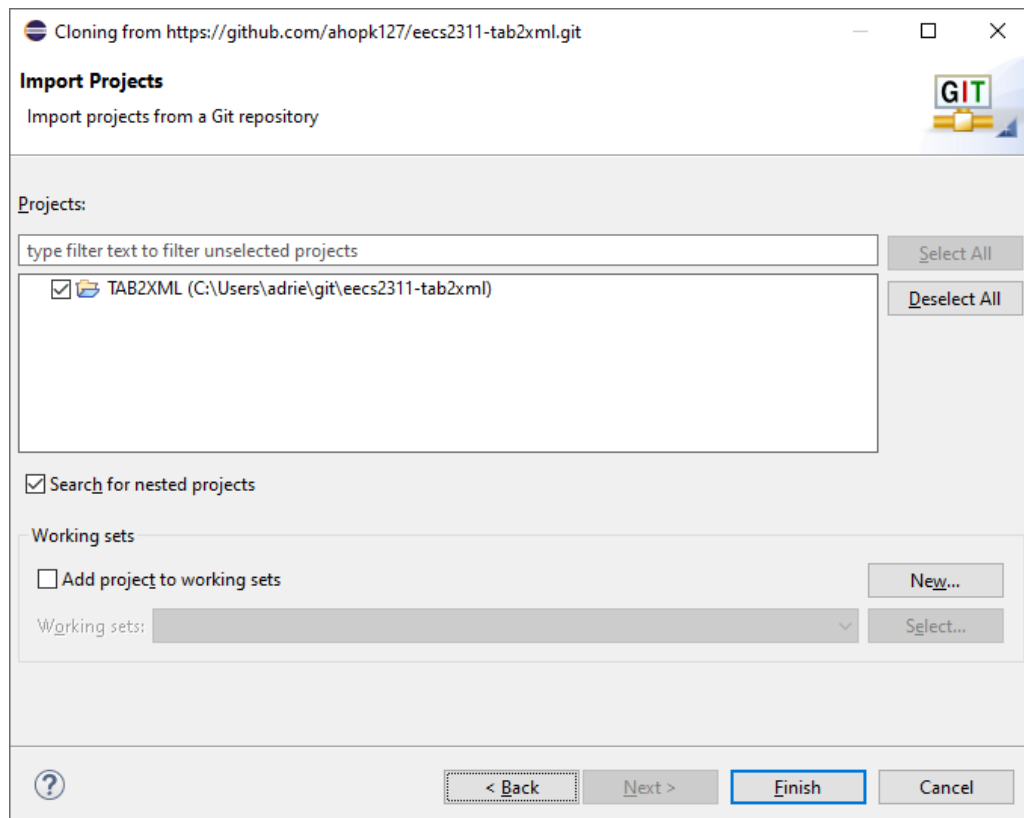
Initial branch:

☐ Clone submodules

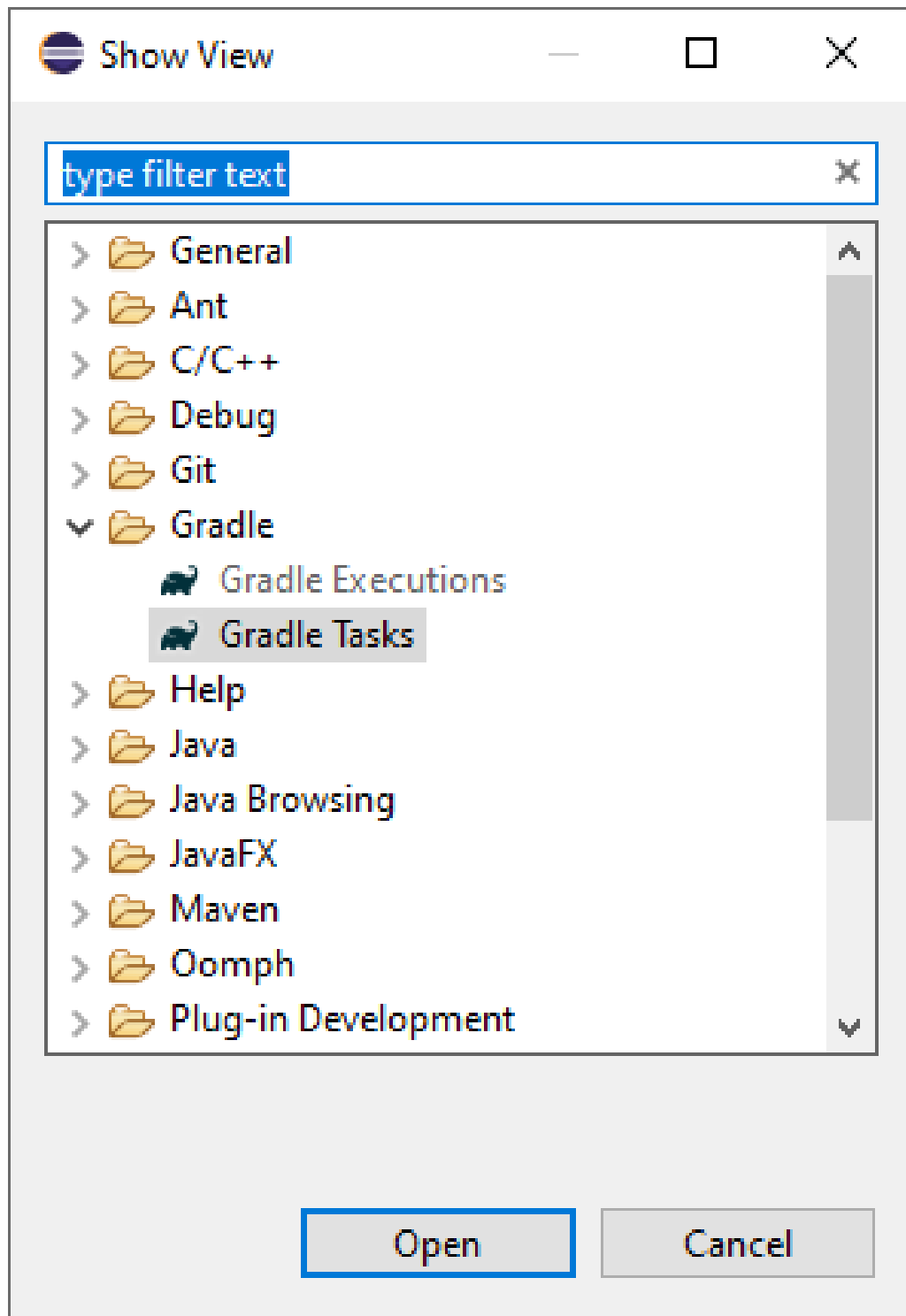
Configuration

Remote name:

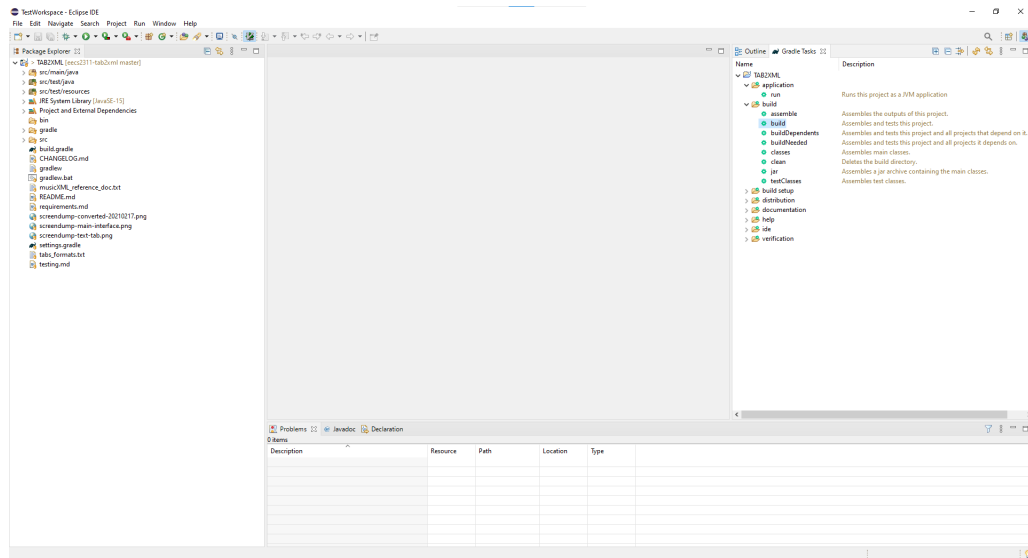
- Click "Next" then "Finish". The program is now installed on your computer, but must be built using Gradle.



7. You will need the "Gradle Tasks" window for the next step. If you can't find it, press **Window** > **Show View** > **Other** in the menu, then find "Gradle Tasks" in the Gradle folder, then click "Open".



8. In the "Gradle Tasks" window, double-click the green "build" item in the "build" folder.



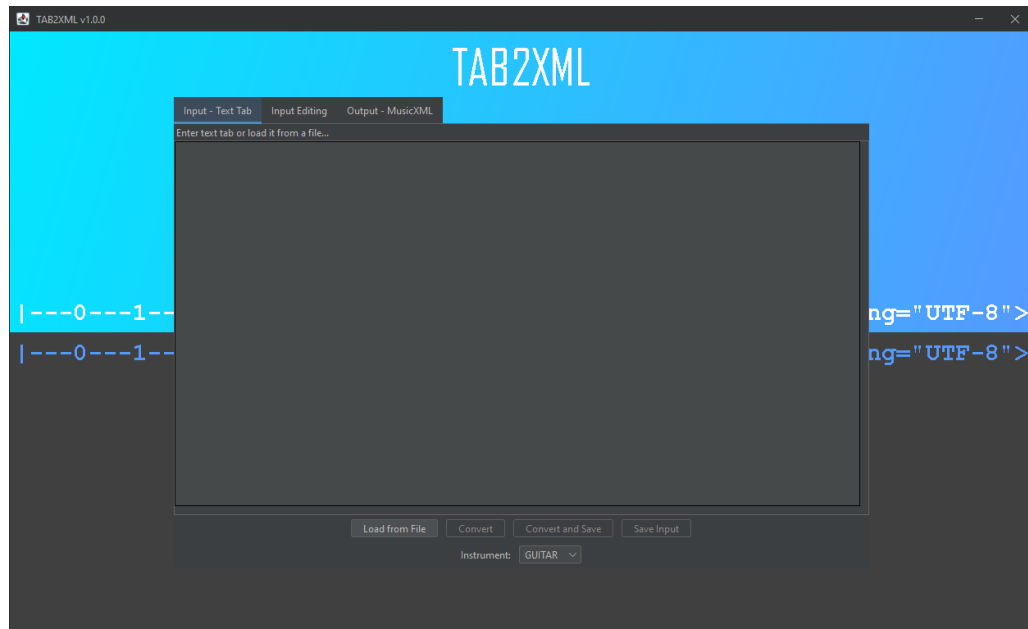
3.2 Commandline

1. Use the command `git clone https://github.com/ahopk127/eecs2311-tab2xml.git` to clone the project to the directory of your choice
2. Change directory to the directory where you installed the project, then use Gradle (`./gradlew build`; if that doesn't work try running `chmod +x ./gradlew` then `./gradlew build`) to build the project.

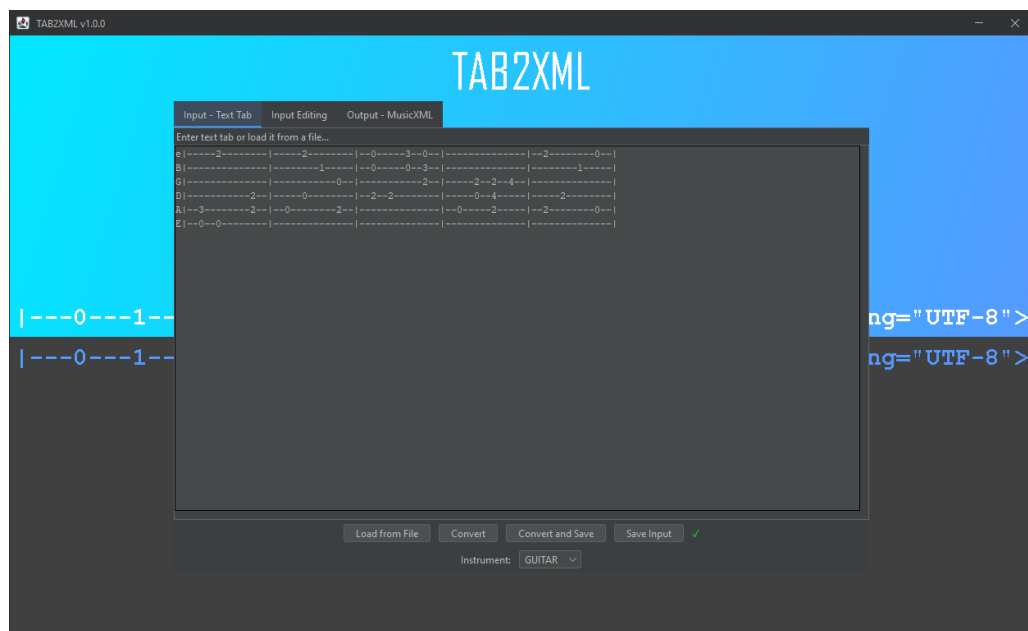
4 How to Use TAB2XML

4.1 Convert Text Tab

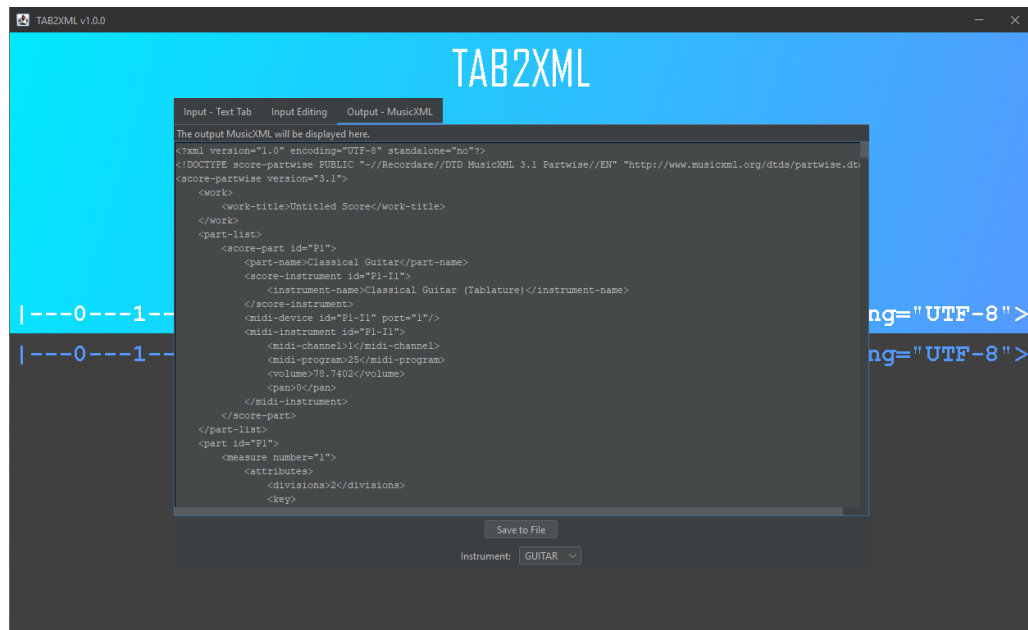
1. Run the application. In Eclipse, double-click the green "run" item in the "application" folder. In commandline, use the command `./gradlew run` in the program directory. You should see a window like this:



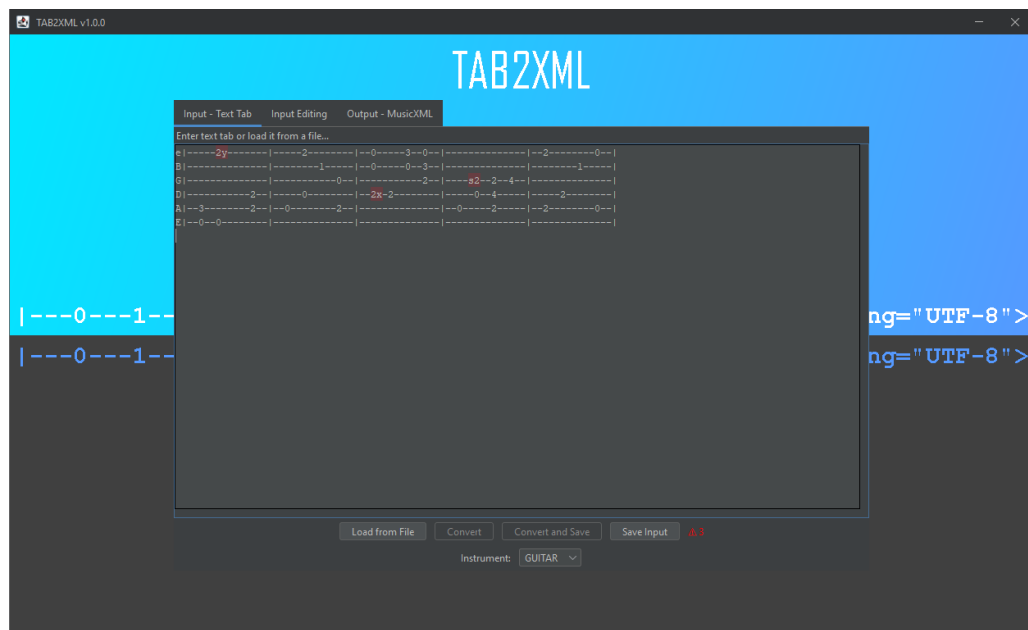
2. Input your text tab into the application. There are multiple ways of doing this:
 - Type or copy-and-paste your text tab into the text box.
 - Press the "Load from File" button then choose a file to load your text tab from a file.
 - Drag and drop a text tab file into the input box
 - Drag and drop text into the input box



- Press the "Convert" button (If there are any errors in your tab, you will not be able to do this; fix any errors first). The text tab will be replaced with the corresponding MusicXML.

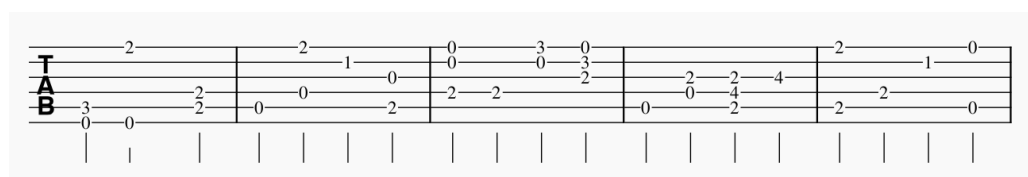


Here is what it will look like if your input tab has errors:



- You can now copy-and-paste the MusicXML, or save it to a file using the "Save to File" button.

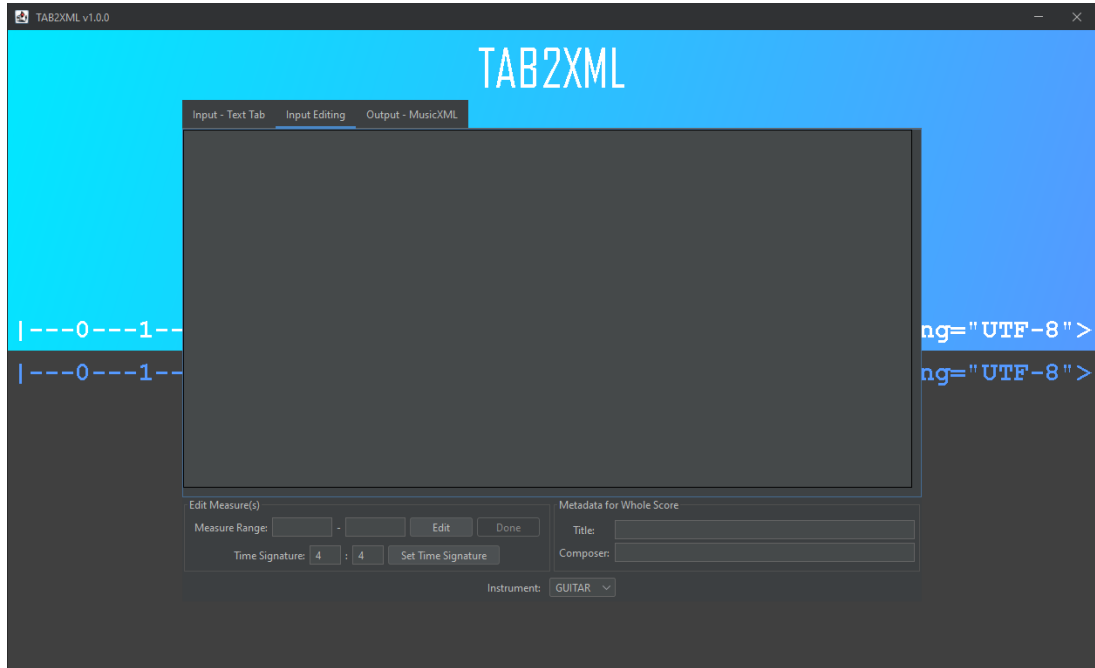
Here is a screenshot of the produced MusicXML, rendered using MuseScore 3.6.2:



4.2 Editing Text Tabs

After converting text tab with the "Convert" button, you can switch to the "Input" tab to edit the input text tab. Use "Save Input" to save the (edited) input to a file.

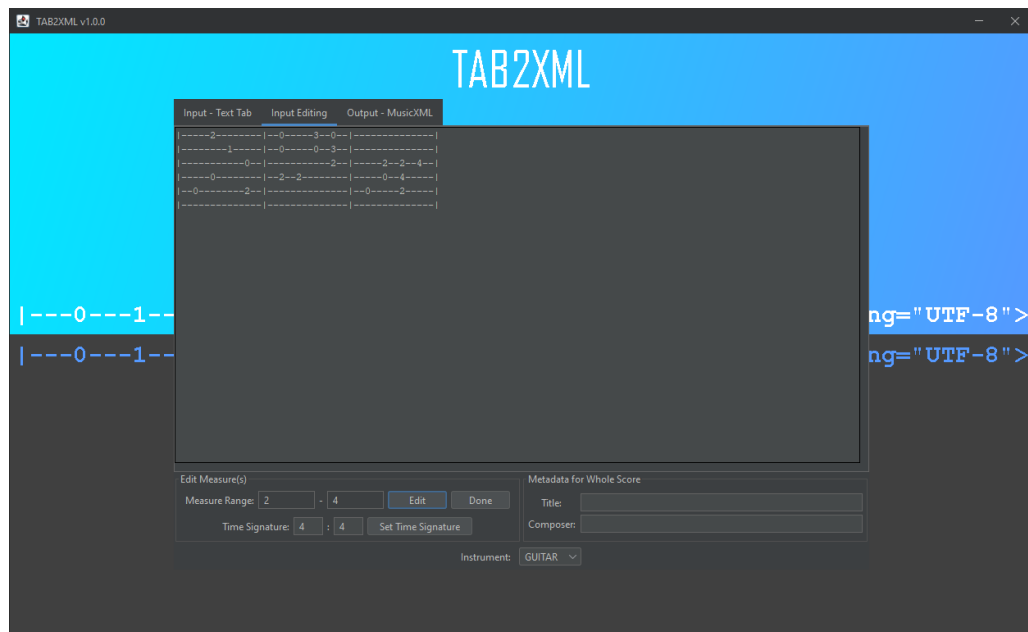
You can use the "Input Editing" tab for more precise editing of the input. This view also allows you to set metadata such as the title of the output score. The view should look like this:



4.2.1 Editing Measures

One thing you can do with this editing view is editing a single measure or a range of measures from the input tablature. To do this, follow the following steps:

1. Input your text tablature in the "Input" tab if it isn't already there.
2. Navigate to the "Input Editing" tab.
3. Input the desired range of measures in the "Measure Range" boxes on the bottom left.
4. Press the "Edit" button. You should see something like this:



5. Edit the tablature in the text box, then click "Done".

Note: This system is not as sophisticated as the parser, and can occasionally fail when exposed to tabs with complex "decorations" that aren't part of the measure text. If this system fails, try removing decorations then trying again.

Warning: Editing a measure region then saving it will remove any decorations from the tab.

4.2.2 Editing Metadata

You can also use the Input Editing tab to edit metadata. To edit the title and/or composer, simply enter them into the text boxes on the bottom right. There is no need to press any "Edit" or "Save" buttons, they will simply be entered in the score when it is converted.

Editing the time signature of a measure range is a bit more complicated:

1. Select the measure range as if you were editing it (follow these steps up to step 4).
2. Input the time signature in the text boxes on the bottom-left.
3. Press the "Set Time Signature" button.

Note: If no time signature range was selected, the "Set Time Signature" button will still work - it will ask you if you want to set the time signature for the whole score.

4.3 Other Use Cases

- You can use the "Convert and Save" button to convert a text tab and save it to a file in one step.
- To clear the text box, press Ctrl-A then Delete. If you want to load in another text tab, you don't need to do this. Simply press "Load from File" or drag and drop to load another text tab.

4.4 How to Use in Code

If you would like to use this project in your own program, the following code can be used to convert tab to MusicXML:

```
tab2xml.parser.Parser parser = new tab2xml.parser.Parser(INPUT, INSTRUMENT);
String output = parser.parse();
```

Notes:

- INPUT is the text tab. It should be a **String** containing the contents of the tablature, not a file or a filepath.
- INSTRUMENT is the instrument the tab is for. It should be an instance of `tab2xml.parser.Instrument`.
- You will need to add a try-catch statement to handle the checked exceptions thrown by `Parser.parse()`.

5 Supported Tabs

You can find a sample of supported tabs here.

5.1 Guitar/Bass

5.1.1 supported features

1. hammer-on notations
2. pull-off notations
3. grace notes
4. repeated measures
5. hammer-on/pull-off combination sequences
6. chords
7. in-line comments
8. multiline comments

5.2 Drum

5.2.1 supported drum types notation

1. BD - Bass Drum 1
2. Bd - Bass Drum 2
3. SS - Side Stick
4. SD - Snare
5. ES - Electric Snare
6. FT - Low Floor Tom
7. HH - Closed Hi-Hat
8. Ft - High Floor Tom
9. PH - Pedal Hi-Hat
10. LT - Low Tom

- 11. OH - Open Hi-Hat
- 12. LM - Low-Mid Tom
- 13. MT - Hi-Mid Tom
- 14. CC - Crash Cymbal 1
- 15. HT - High Tom
- 16. RD - Ride Cymbal 1
- 17. Ch - Chinese Cymbal
- 18. RB - Ride Bell
- 19. TA - Tambourine
- 20. SC - Splash Cymbal
- 21. CB - Cowbell
- 22. Cc - Crash Cymbal 2
- 23. Rd - Ride Cymbal 2
- 24. HC - Open Hi Conga
- 25. LC - Low Conga