Official User Manual

MusicXML Converter

EECS2311 Software Development Project

Date: 11/04/2021

Group 14

Adil Hashmi - adilhashmi.w@gmail.com
Alexander Arnold - alex290@my.yorku.ca
Boho Kim - kimboho614@gmail.com
Kanwarjot Bharaj - kjsingh76@gmail.com
Stanley Ihesiulo - stanihe1901@gmail.com

Table of Contents

1. About TAB2XML

1.1. Product Name and Intended Use

TAB2XML is a tool designed to convert musical tablature in a text format into MusicXML, a popular open-source file format used for exchanging digital sheet music. TAB2XML is currently a simple java-based application with more features being added every week.

1.2. Features of Product

This product can convert tablature of the following instruments: Guitar, Drum, Bass.

Note: Since the product has not been implemented fully, instruments that the product can convert are limited, as of now. Reference How to Use.

2. System Requirements

Available Disk Space	50 MB
RAM	256 MB
Java version	Java 15
Operating System	Windows, macOS, any platform with Gradle

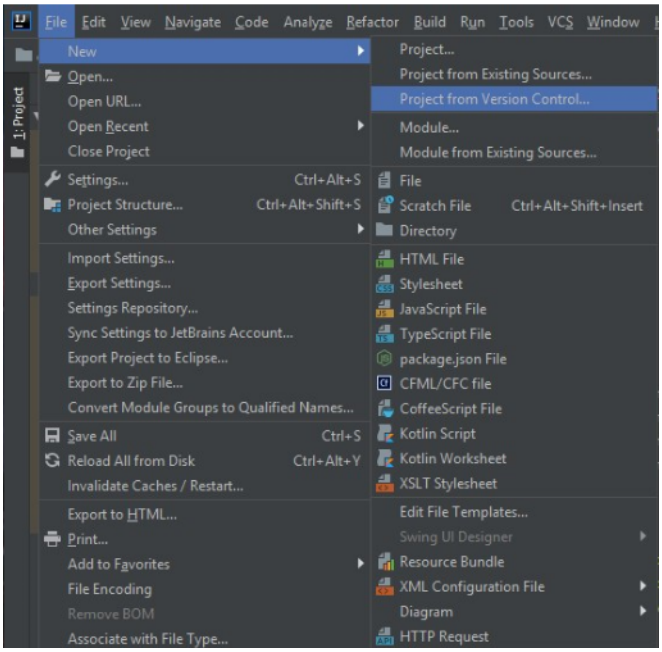
Our system uses Java features such as text blocks, switch expressions, and some regex features which are not present in previous versions of Java. Thus you will have to ensure that you both have these

3. Installing TAB2XML using Gradle

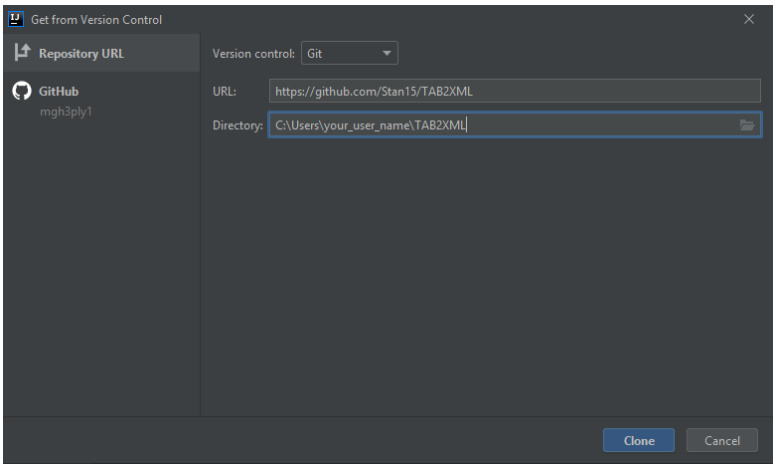
TAB2XML is built as a Gradle project and thus should work on any IDE of your choice. However, we will only go over how to run the program using the IntelliJ and Eclipse IDE.

3.1. IntelliJ

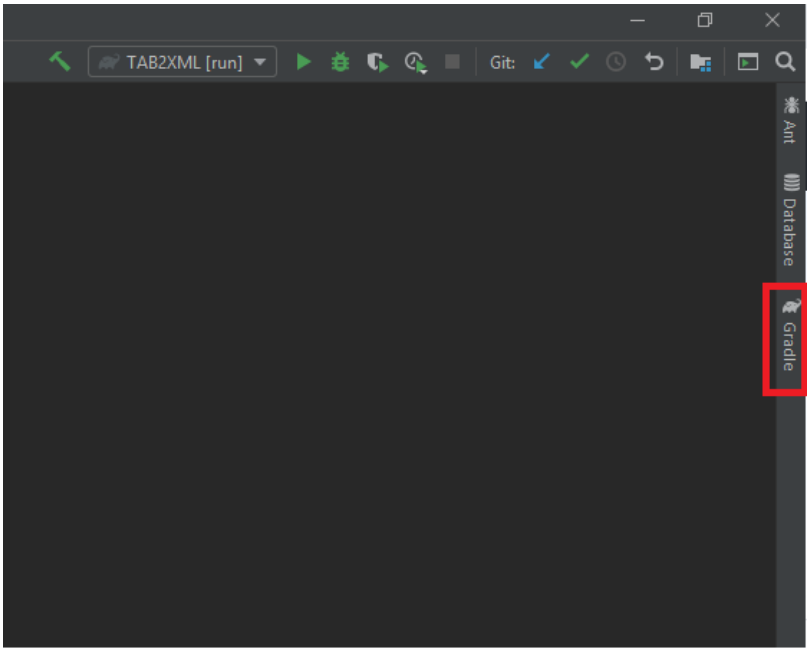
1. *We first need to clone the project from the online GitHub repository onto our local device. To do that, first, open IntelliJ. At the top right, select File > New > Project from Version Control*



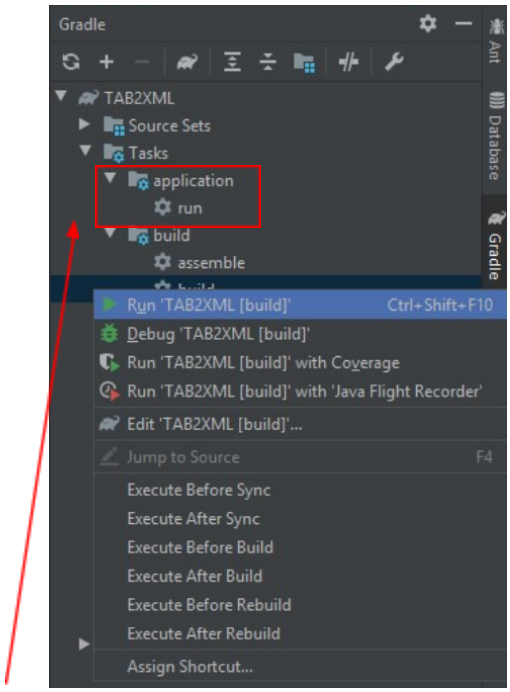
2. *This will direct you to a window that prompts you to enter a repository URL. Enter the link https://github.com/Stan15/TAB2XML as the URL, and select a directory where the the project should be saved.*



3. *Once the project opens, click the Gradle toolbar on the right side of your screen.*



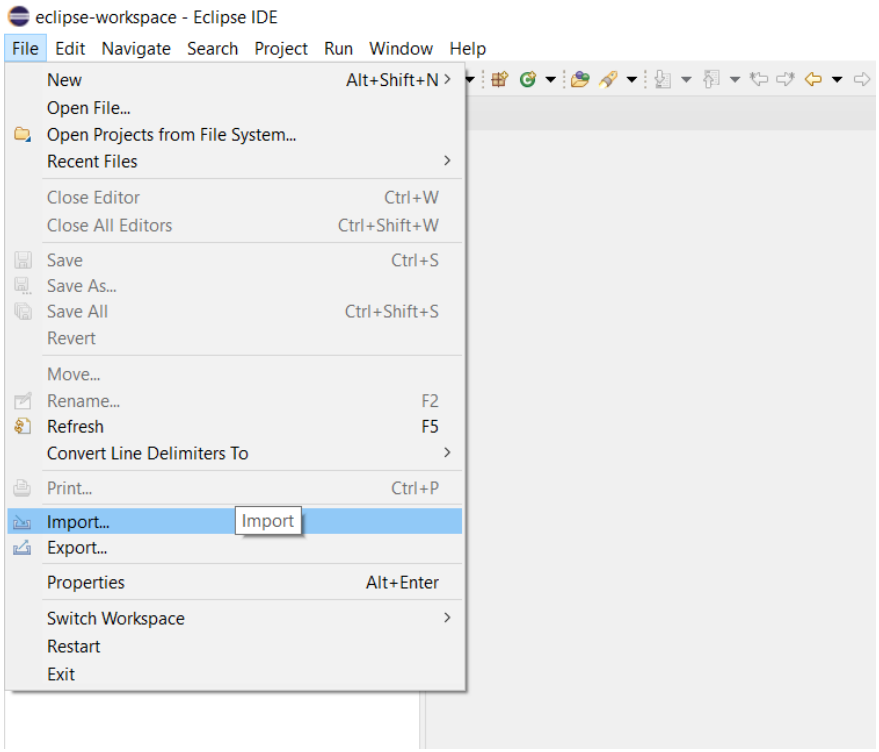
4. *When the toolbar opens, click Tasks > build, and then right-click "build" and the following dialog should pop up. Click "Run TAB2XML [build]"*



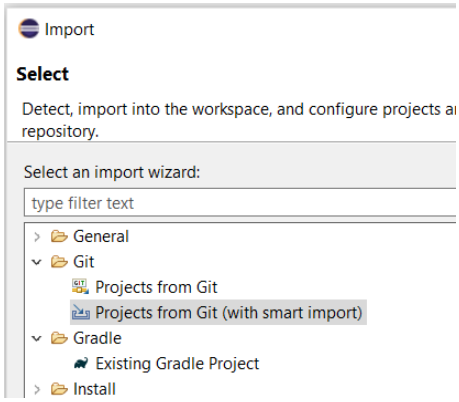
5. *Finally, select Tasks > application, and then right-click the run task and select Run 'TAB2XML [run]'. The program will now launch.*

3.2. Eclipse

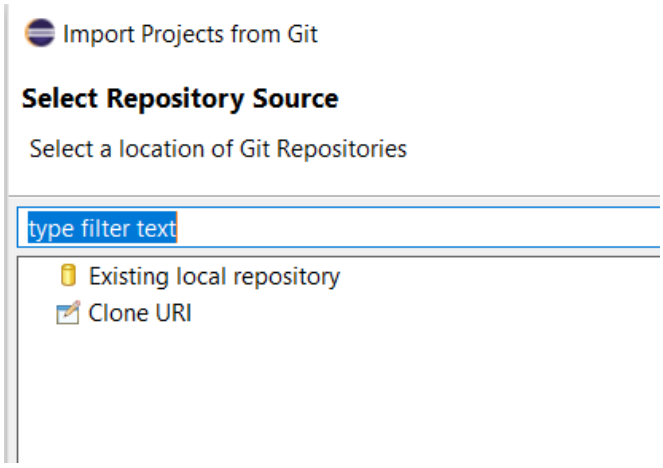
1. *Like with IntelliJ, we want to clone the project from the GitHub repository. To do this, we select File > Import.*



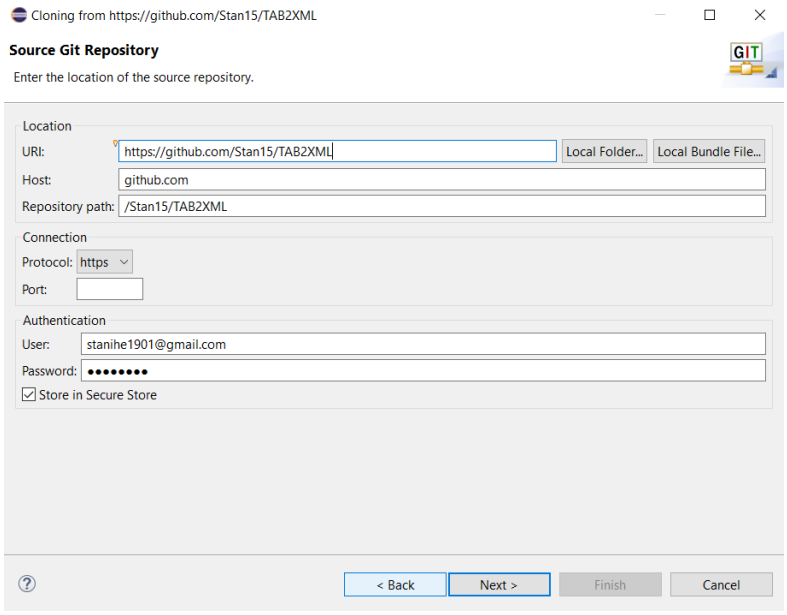
2. *This will take you to a new window like the one below. Under the Git folder, click Projects from Git (with smart import)*



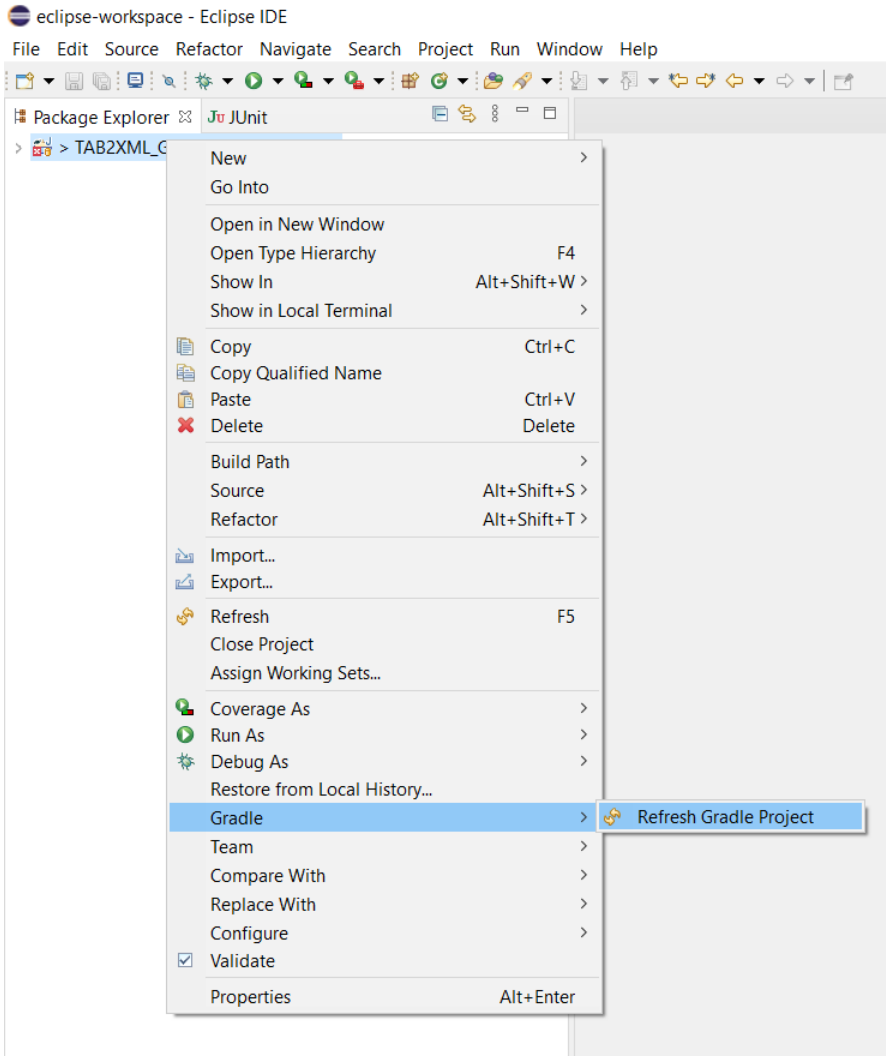
3. *Next, click Clone URI*



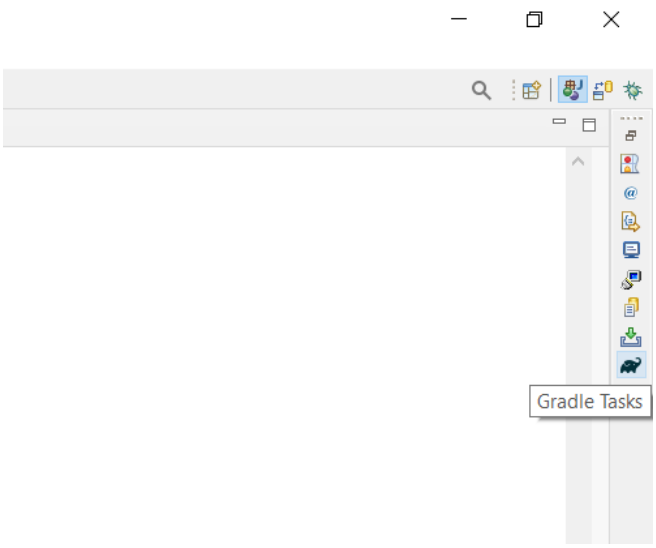
4. *This will take you to a window where you are prompted to input a URI. Paste in the link https://github.com/Stan15/TAB2XML and click proceed with the steps (clicking next) until you see the finish button. Click on that.*



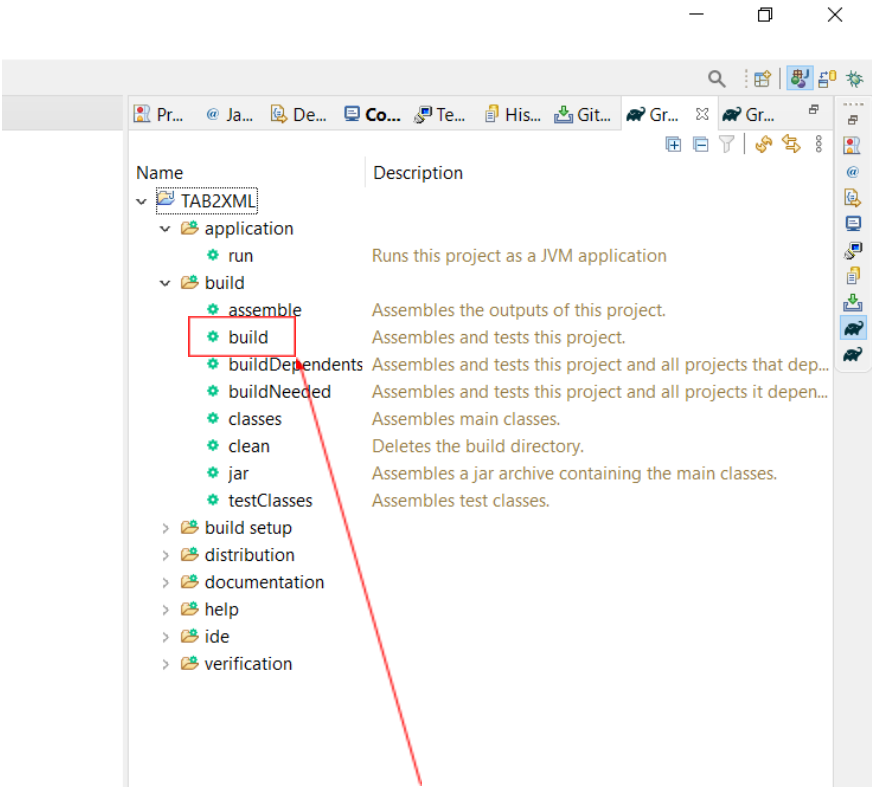
5. *Now we have imported the project, we want to refresh it as a Gradle project. Right-click the project folder in the project explorer then click on Gradle -> Refresh Gradle Project to refresh the gradle project.*



6. *Now we can build and run the project. Click on the "Gradle tasks" icon on the right of your screen.*



7. *Double click on the Gradle "build" task at <u>TAB2XML > build > build</u>*



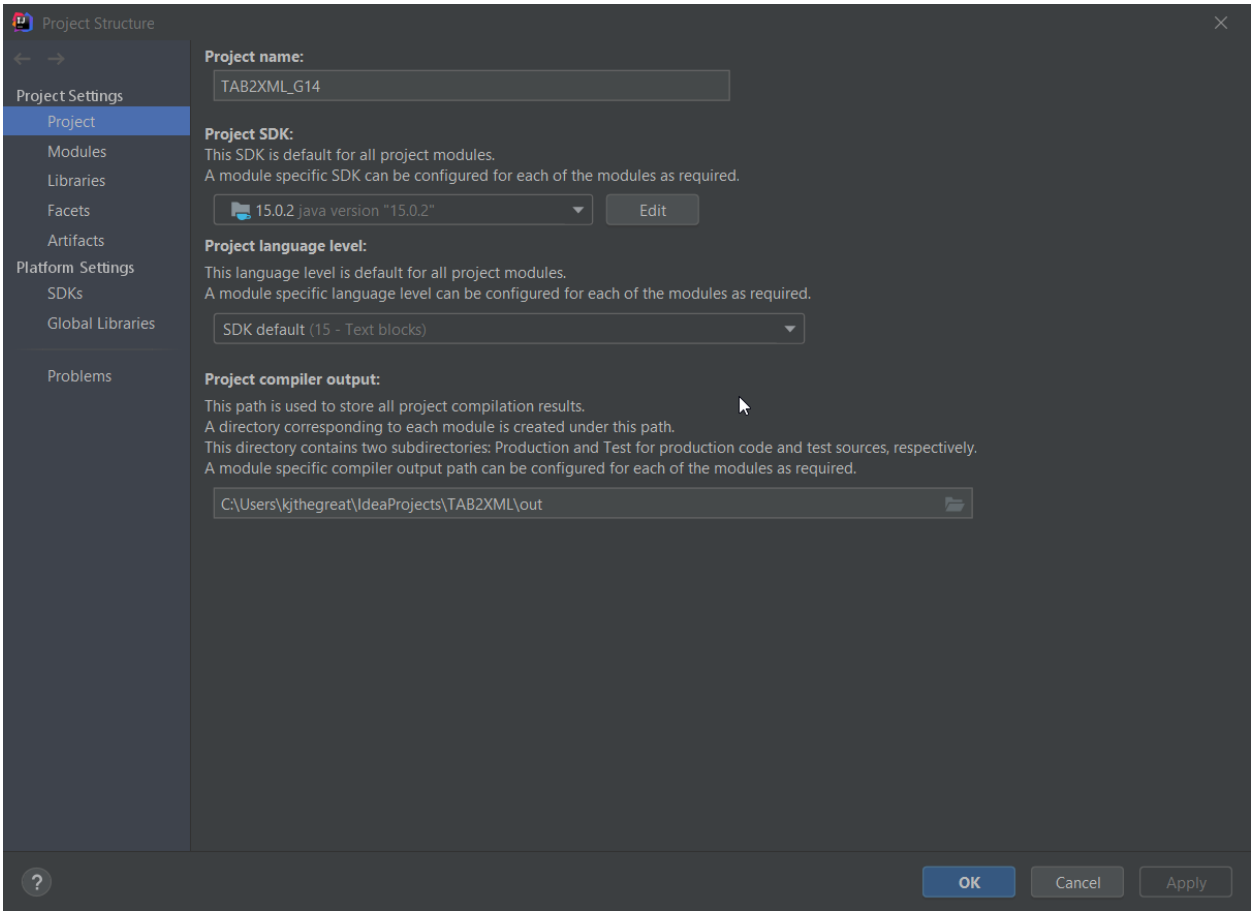
8. *Finally, click on the run task on TAB2XML > application > run*

3.3. Issues with SDK and Gradle

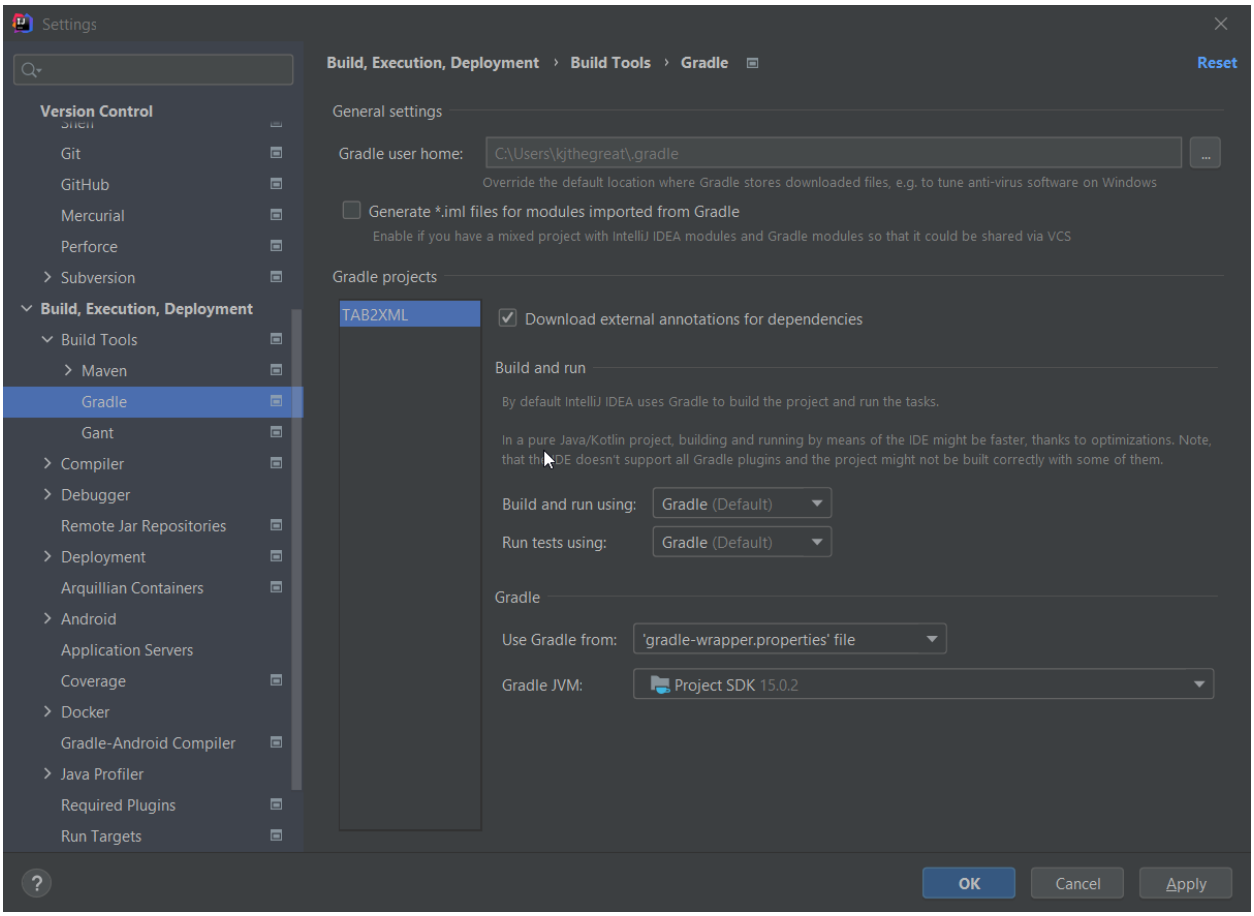
Make sure that both your Java SDK and Gradle are using Java Version 15.0.2 otherwise some of the features will not be supported or the application may not even run.

Intellij

To check Java SDK on Intellij, Click File > Project Structure to see your SDK.

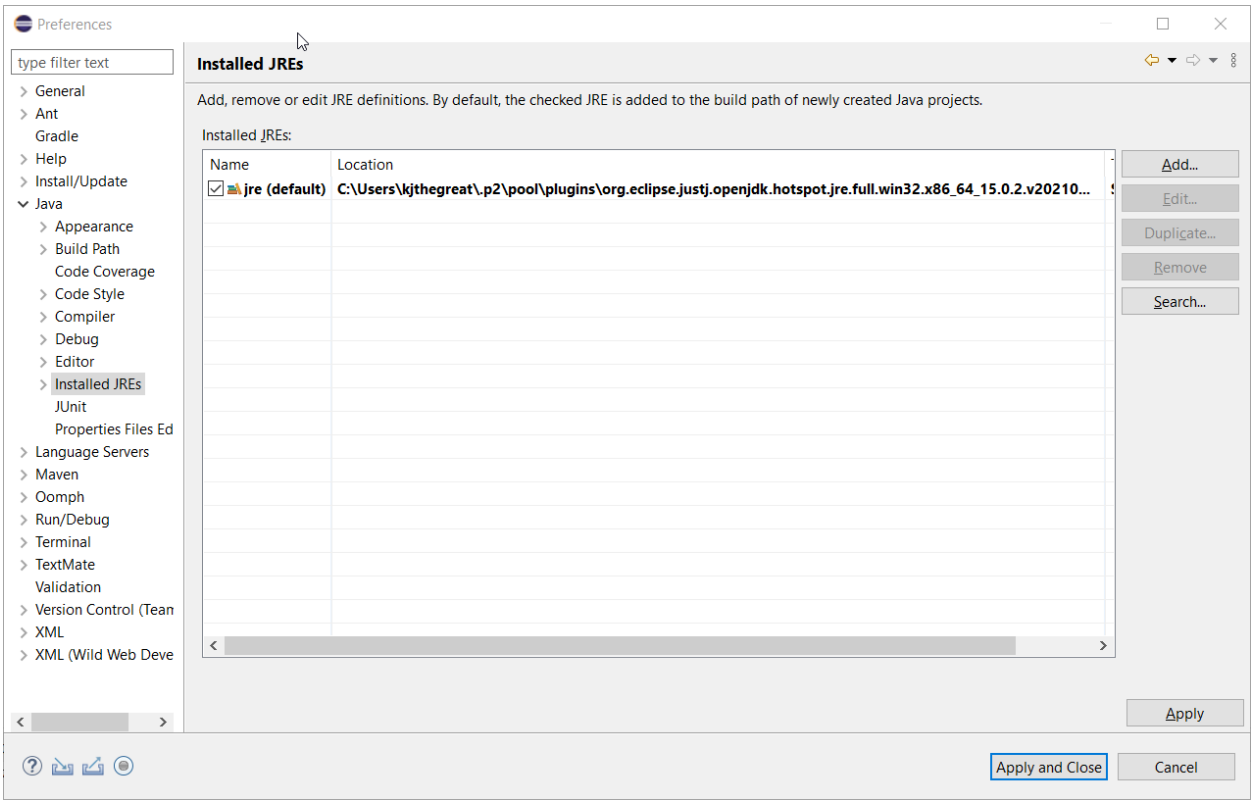


To check your Gradle Version, Click File > Settings > Build , Execution, Deployment > Build Tools > Gradle



Eclipse

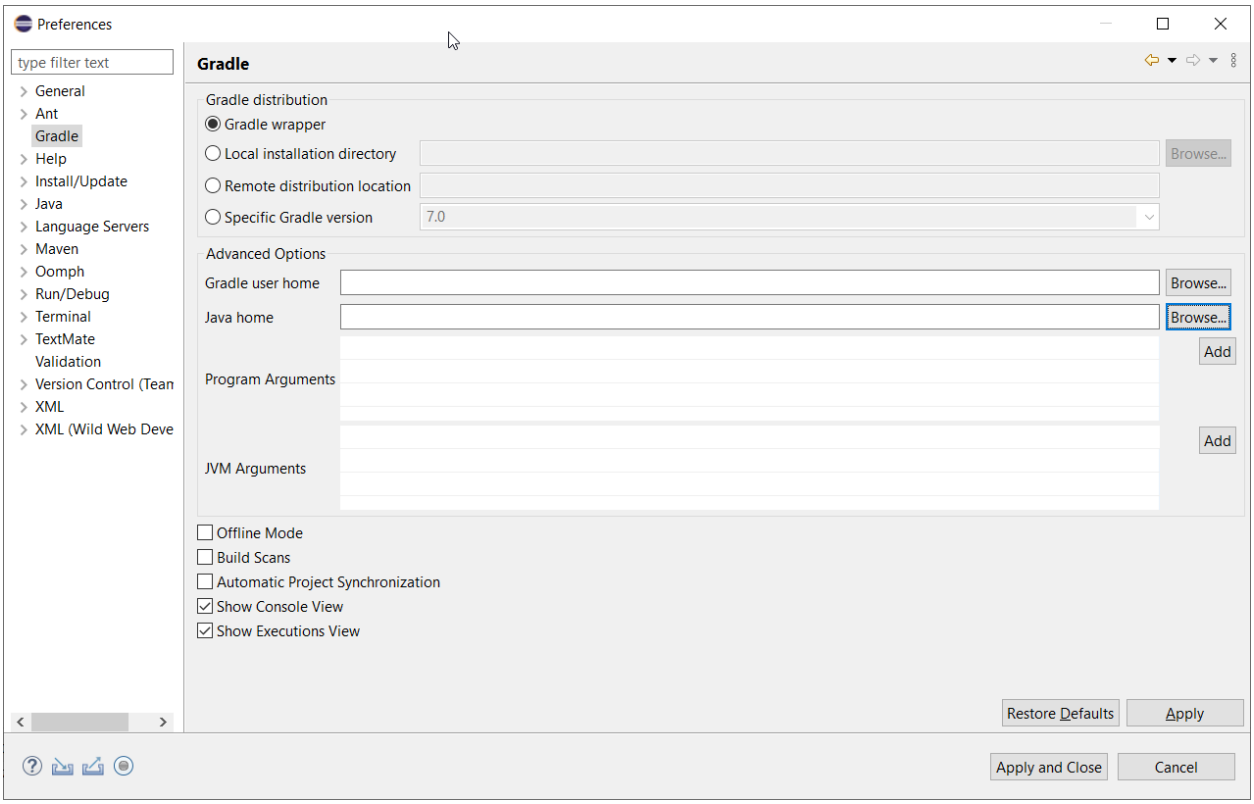
To check Java SDK on Eclipse, Click Window>Preferences>Java>Installed JREs



For Eclipse it is a bit tricky as it does not explicitly show the Java Version it is using for Gradle. If your default version is Java Version 15.0.2. and the application runs, DO NOT CHANGE ANYTHING HERE.

If it's still not working, look at the steps below.

To change your Gradle Version, Click Window > Preferences > Gradle



Then, under Advanced Options, Locate Java Home, click browse and locate where you have installed your Java Version 15.0.2.

Since Eclipse does not handle Gradle customization well, there may be a case where even after following the steps above, the application does not run. In this case, you must uninstall all JRE from the platform, only keeping Java version 15.0.2 and reinstalling Eclipse. While reinstalling Eclipse, make sure that the DEFAULT Java Version is 15.0.2. This should fix the issue.

4. How to use TAB2XML

1. When you run the program, you will be able to see a text field at the center of the screen *(Figure1)*. This is where you paste in your tablature .txt file.

2. To put your input, click '*file -> open'* and choose your file, or copy and paste your text file into the text field.

3. Once you put your text input, the system lets you know which information it is not able to recognize and which information should be fixed for appropriate conversion. It highlights the erroneous areas of your input and displays a popup message when you hover over the highlight with your cursor.

There are 4 levels of highlighting:

i. Red highlight: This is used to identify errors that may critically affect the output of the conversion.

ii. Yellow highlight: Errors with this highlight are less critical, but we do not guarantee an accurate output with these errors.

iii. Blue HIghlight: Errors with this highlight will likely still produce an accurate output but may lead to a different output than is expected.

iv. Grey highlight: This highlight is used to identify content that may have little to no effect on the output.

Error Examples

Detailed below are a few examples of different error highlighting scenarios:

- Grey highlight: “This text can’t be understood.” (**Figure 3**).

This is used to identify text which was not identified to be a score object (i.e measure, note, repeat instruction, e.t.c.)

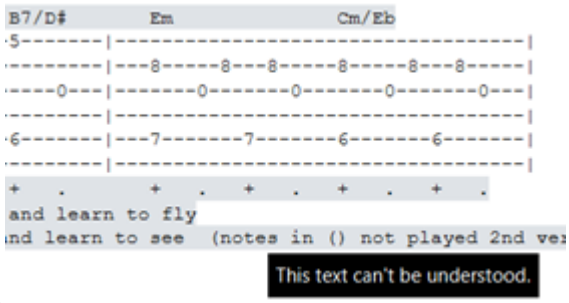


Figure 2

- Blue Highlight: “Adding whitespace might result in a different timing than you expect.”

  This is used to guide the user in making a more reader-friendly tablature. It helps the user easily debug why they may be getting a different timing than they expect to have.

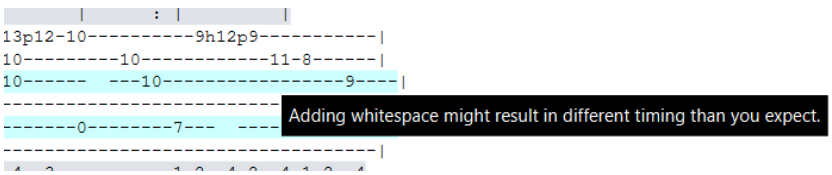


Figure 3

- Yellow highlight: “A guitar measure should have 6 lines.” (**Figure 4**).

As a warning, if you get such an error on a measure that seems to be accurate, make sure no text is written on the side of the measure as this makes the system identify it as two different measures. Reference the *Input Requirements* section of this manual for more information regarding this.

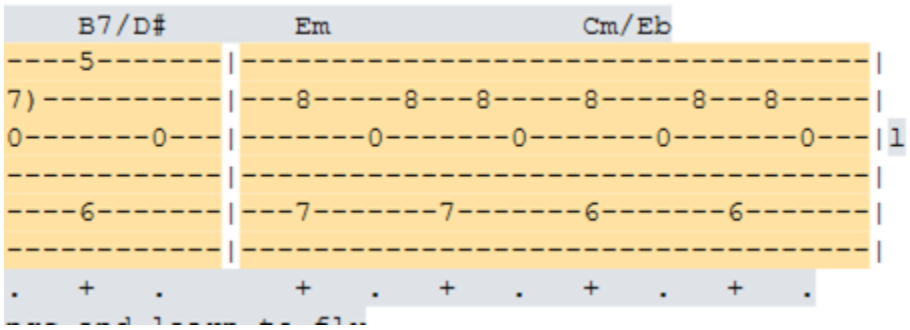
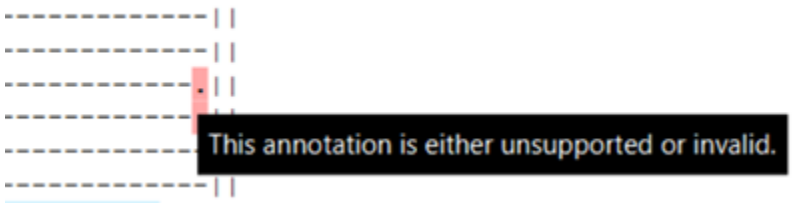


Figure 4

- Red highlight: "This annotation is either unsupported or invalid" (**Figure 5**)

This is used to identify elements that are either not supported or not identified as valid measure annotations.



Note: There are more error scenarios that may occur, but they are all categorized into the three groups identified above.

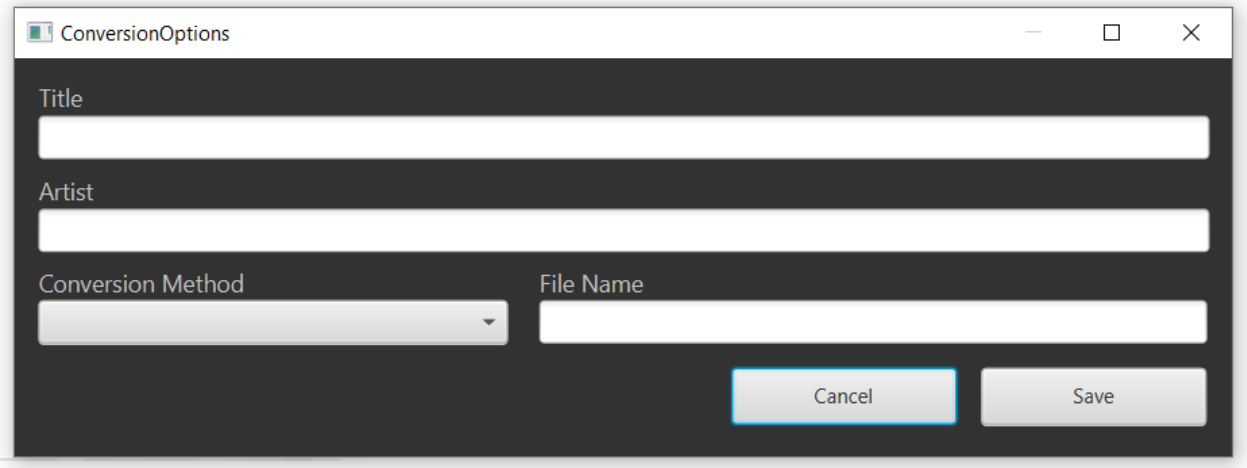
4. For a proper output, make sure there are no red or yellow highlights in your tablature. It is recommended that you resolve blue highlight errors, but grey highlights can largely be ignored.

    - How to fix errors:
      Ensure that the number of lines in a measure is correct for the given instrument.
      Remove all unrecognizable notations in tablature and replace with dash '-'.

Remove all texts which is placed around score bars except for key notation.
(How to fix instruction video: https://drive.google.com/file/d/174oWzswHkvnTvyask_AUpYKRjKmuz3_m/view?usp=sharing

Note: *If you removed all yellow and red highlights, it is ready to be converted.*
(If there is no yellow and red highlight in score bars information, you can skip this step)

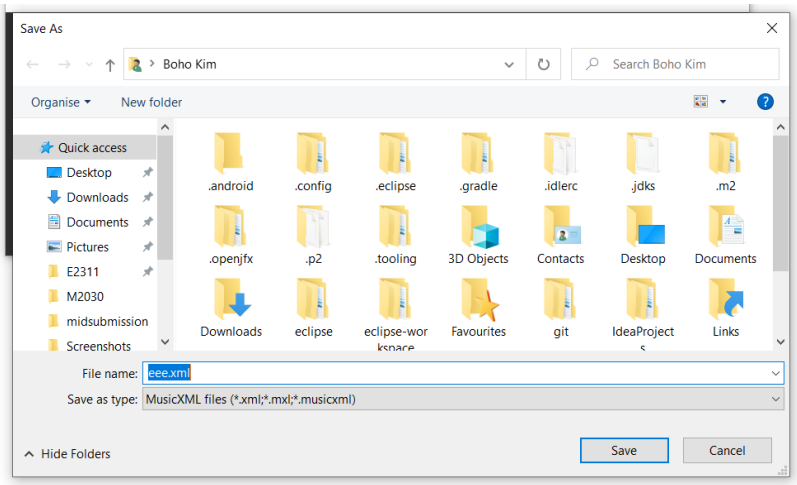
5. Click the "Convert" button. This opens a new window with some options (**Figure5**).



Title – You can set the title of the song.
Artist – You can set the artist's name.
Conversion Method – You can choose your score type. Piano or Tablature type.
File Name – You can name your file.

You can set any or none of these options, then click the convert button.

6. When you are done, click on the “save” button. Navigate to the location where you want to save your converted file and save it. You can also change the file name *(Figure 6)*.



5. Input Requirements

5.1 Measure instructions (Repeats and time signature)

This program allows for the application of repeats and time signatures to individual measures. Here, we will go over the input restrictions governing these features.

Some sample tablature text files that meet the requirements can be found in the project folder in the directory TAB2XML/src/test/resources/test_tab_files.

For instructions in general, the following requirements are outlined:

1. For a line to be interpreted as having instructions, it must only be composed of valid instructions separated by spaces and nothing else.

2. The ‘tab’ button should not be used in your instruction lines as this might result in the system not applying the instructions to the correct measure.

3. For your instructions to be recognized, the line directly below the instruction line must be a measure line or another instruction line (instruction chaining is allowed).

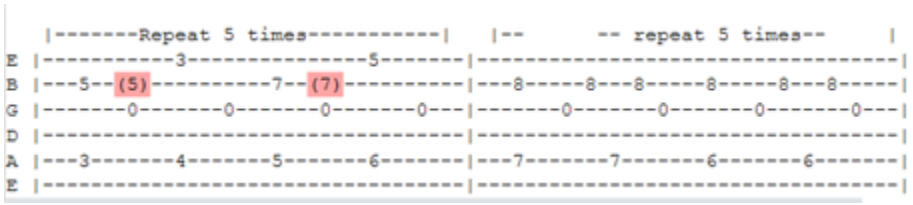
4. Lines of instructions are chained by connecting the lines by one new line.

5. The order of priority for applying instructions is left to right, up to down.

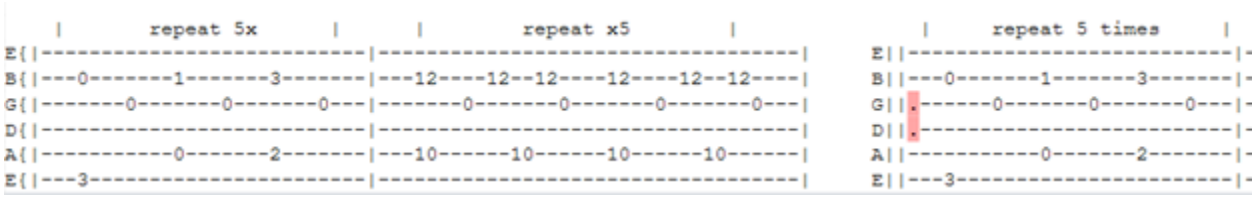
Repeats:

For repeats, the following input requirements must be followed for repeats to be correctly applied

1. Repeats must start and end with a vertical bar, and can have any combination of spaces or dashes "-" in-between, as seen below.



2. Repeats can be notated in any one of the three below-stated ways.



Time Signatures:

1. The list of possible time signatures has been artificially restricted to the following generally accepted time signatures: 2/4, 2/2, 3/8, 3/4, 4/8, 4/4, 4/2, 6/8, 6/4, 9/8, 9/4, 12/8, and 12/4.

2. If an invalid time signature is provided, the following error is received:

3. The default time signature is 4/4 if no time signature instruction is provided.

4. Time signatures with a beat or beat count consisting of three or more values are not recognized as instructions and will make the line of instructions invalid.

5.2 Measures

Some sample tablature text files that meet the below requirements can be found in the project folder in the directory TAB2XML/src/test/resources/test_tab_files. The tablature file input into the program must meet the following requirements:

5. The tablature file must start with a vertical line after the string name.

6. you may not have text by the side of a measure which itself is not a measure.

7. The line names must all be lower caps, except for the E string which can be lower caps to distinguish the lower e string from the upper E string.

Note: The system does not guarantee an accurate output if measure collections do not contain blank lines separating them, and if they do not have clear line names specified. (i.e string names/drum names).

Note: *The system does not guarantee an accurate output if measure collections do not contain blank lines dividing them and if they do not have clear line names specified. (i.e string names/drum names).*

Note: *The system does not support dangling slides; slides that are not connecting two notes (in-between two notes). A slide must be placed in between two notes. (Figure 7).*

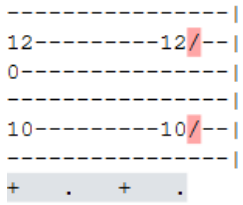


Figure 7