MusicXML -User Manual

-By: Group 3 EECS 2311

Group 3:

Aman Patel
Phuong Tran
Mike Shen
Maksim Kolotev

# Table Of Contents   Page No.

Introduction

MusicXML is a digital sheet music interchange and distribution format. The goal is to create a universal format for common Western music notation, similar to the role that the MP3 format serves for recorded music. The musical information is designed to be usable by notation programs, sequencers and other performance programs, music education programs, and music databases. This software converts musical tabs into a MusicXML file. So, users can create musical notations out of the MusicXML file.

System Requirements

1. Windows 7, 8, 10, Server / MacOS / Linux
2. Platform: Eclipse / Intellij
3. RAM: 4GB
4. Storage: 50MB
5. Java SE-15 Environment (build by Gradle)

Installation

1. Go to the specified GitHub repository: https://github.com/mike-jshen/2311-Project
2. Once there go to prototype under Switch branches/ tags
3. Once there copy the link under Code or the link from step 1
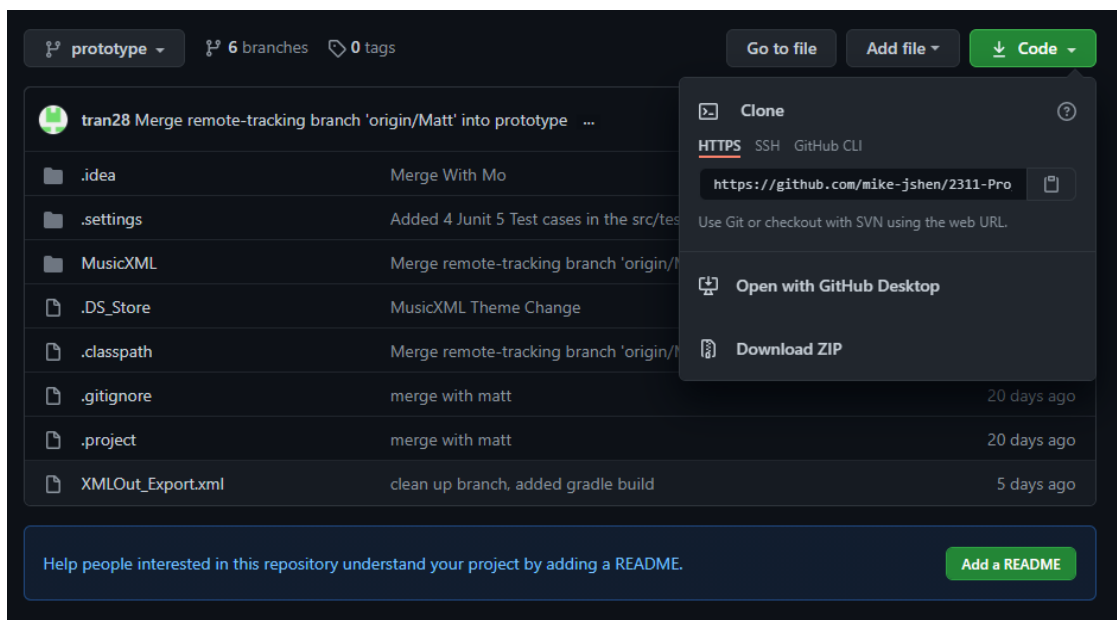
4. Once in your Java IDE, click File and then import
5. Select "Git" and once there click "Projects from Git". Then click "Clone URI"

6. Please paste the URI of the github after that, everything should automatically fill and Authentication is not needed. Click Next, then Next again, then please select Initial branch as "prototype".
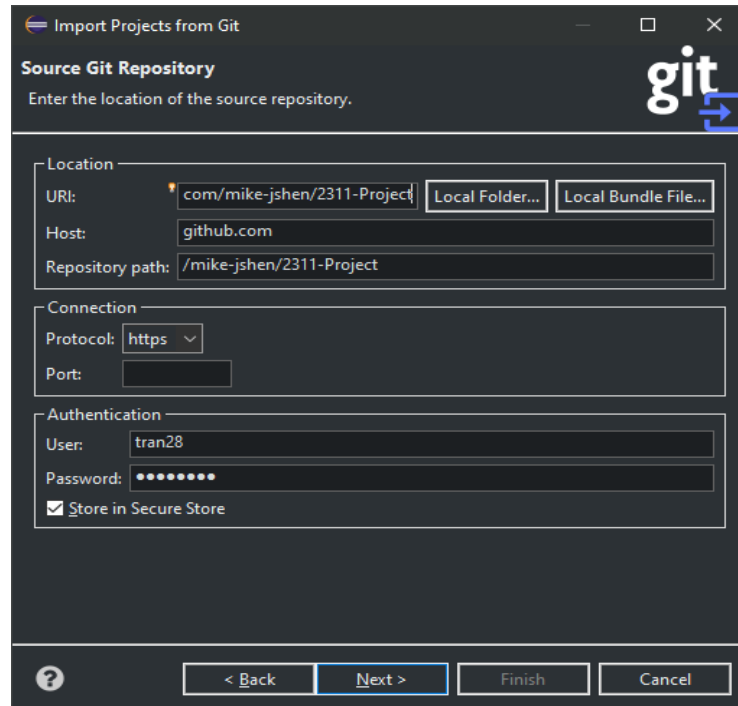
7. After initial branch is selected, browse Directory and place it in the file you choose Example:
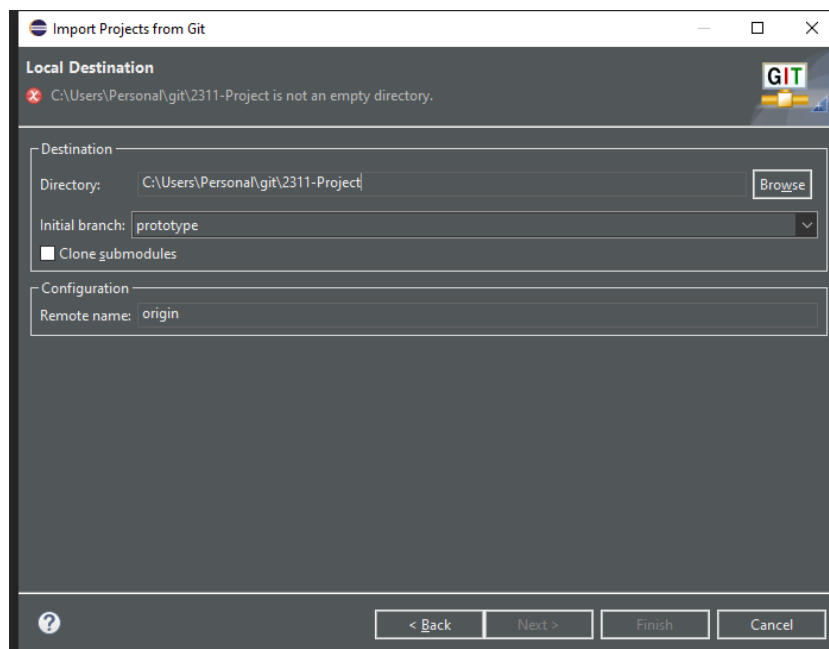
8. Click Next, then Finish.
9. Your Java will now have the project.
10. Once finished, right click the project and navigate to Gradle and click refresh
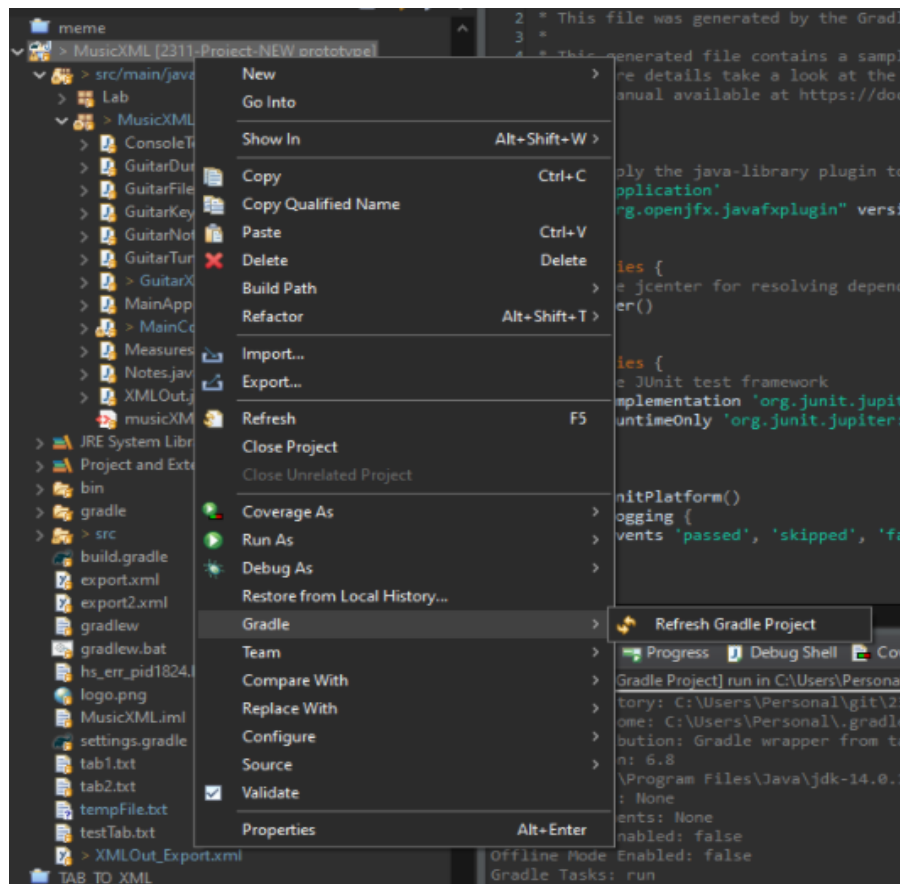
11. Navigate to Gradle Tasks, build the program[1], then run the program[2].

Overview

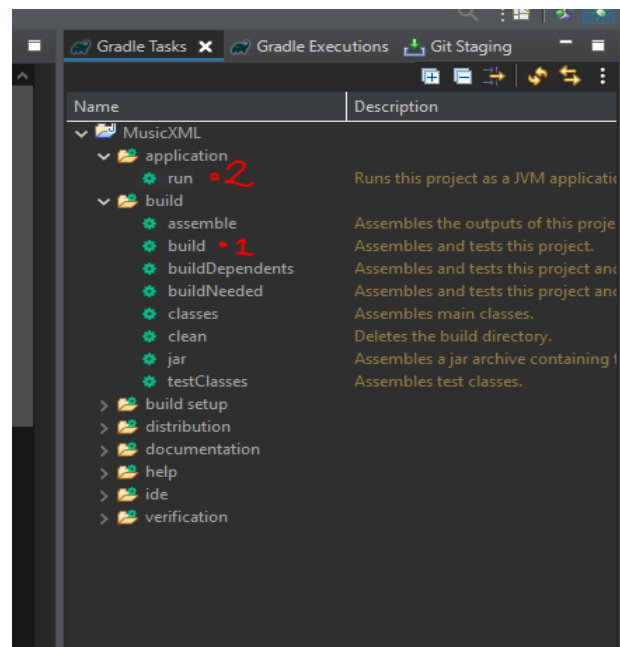Musicians use sheet music to represent musical compositions, known as scores, in a textual form. Traditionally, paper has been the medium for creating and sharing sheet music; however, this is changing. For over 10 years now developers have been working on and improving a product that stores sheet music digitally in plain text so that it can be easily shared and manipulated by musicians. This product is called MusicXML, it uses XML to store musical notation digitally, and it has become the standard open format for representing sheet music. The purpose of this project is to create a desktop application built in Java that can keep track of a library of MusicXML files in a database and parse these files on the fly. Covered in this report is the reasoning for creating the application, what tools were used, and how these tools were used to implement the project. As well, the design of the project, issues encountered when creating the application, and what was accomplished is detailed.

Graphical User Interface (GUI)

The startup interface of the application contains 4 menus to choose from. *Requirements Document*, *Testing Document*, *User Manual*, and *Launch Program*. Clicking on *Requirements Document*, *Testing Document*, and *User Manual* will open up the corresponding PDF using the system's default PDF viewer. Pressing the Launch Program button will launch the main application.
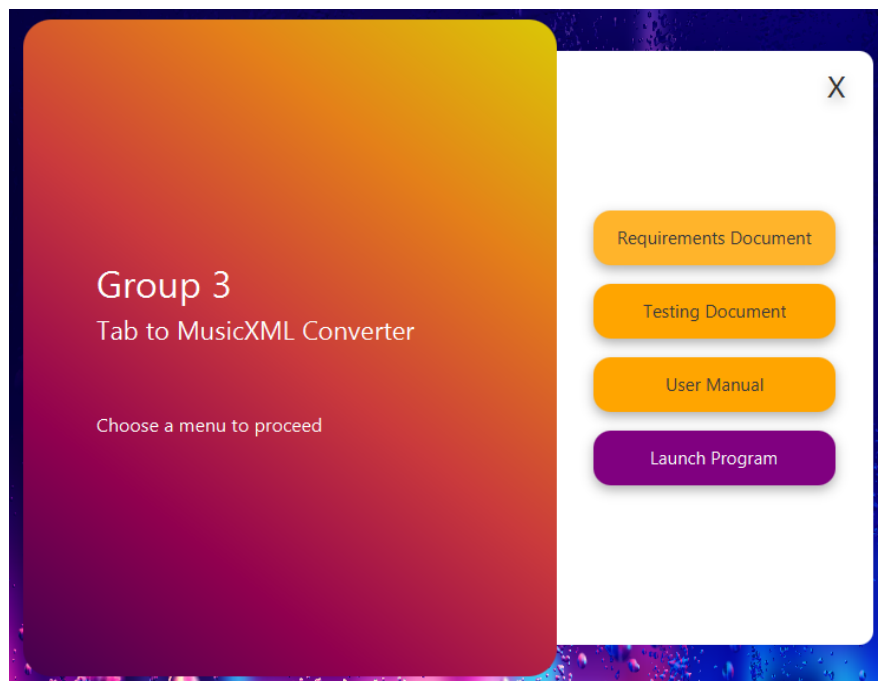
The main interface is divided up into a footer bar, toolbar, tabs, customization panel, Text, and XML Panel. On the left side of the interface, there is a preview where you can see the added Songs, and right next to there is a preview panel, where you can see the Tabs and edit them if need
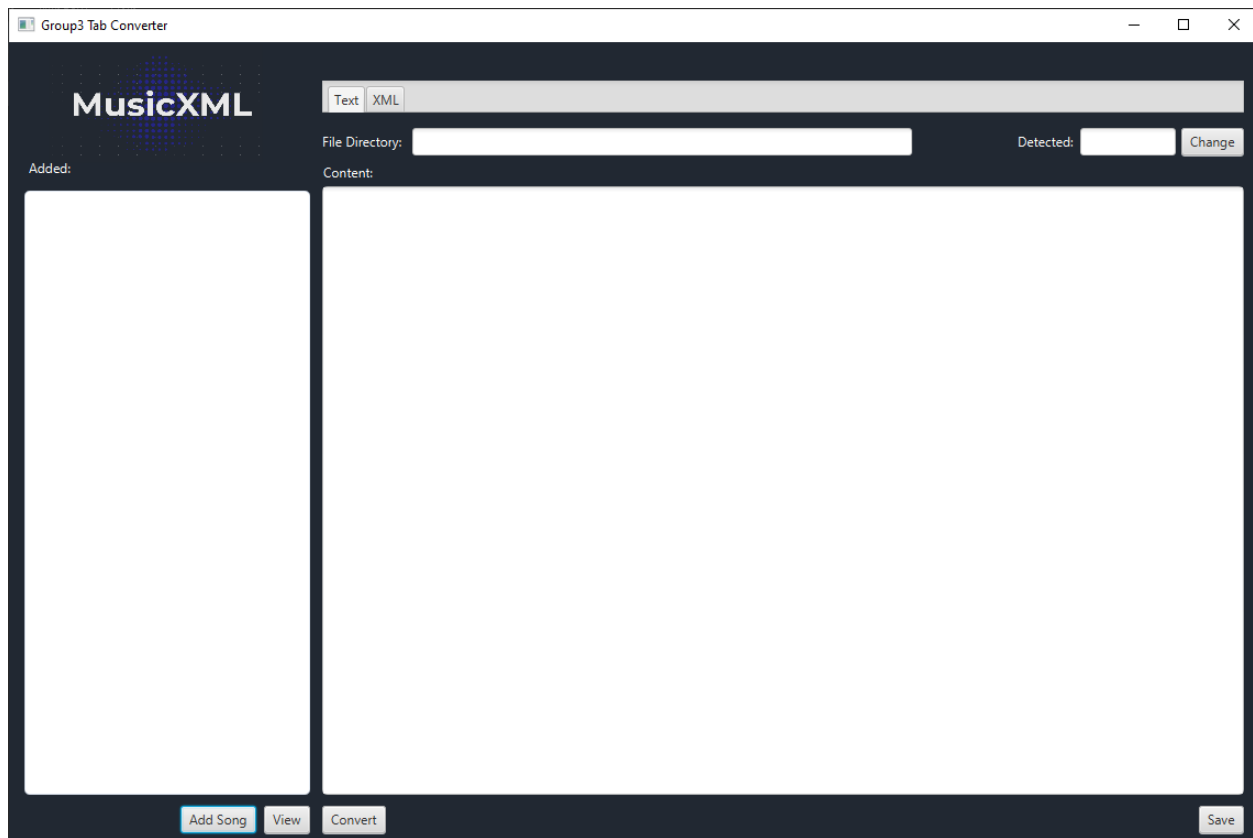
In the footer bar contains the options to Add Song from Directory, View it into the preview panel and Convert tabs into MusicXML format and Save them.

The Top right hand side toolbar contains the options to select the Text or XML panel and Change the Detected instrument Tab type. It also shows the file Directory of the previewed file.
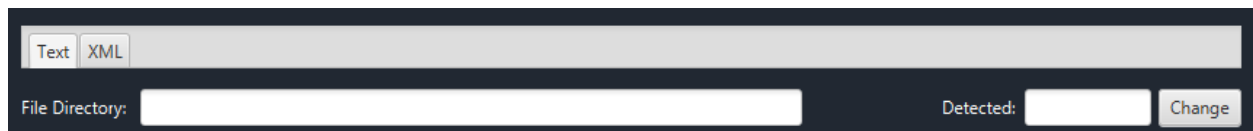
Music Notes Restrictions & Requirements (Currently)

- Prototype recognises key tuning, user can input either a tab with or without key at the beginning but the first or second character must be '|' to be considered the beginning of a staff. If the first or second character is not '|', the whole staff will be ignored.

    1. Without key followed by '|' (valid)

    ```
    |----------0-----|-0--------------|
    |--------0---0---|-0--------------|
    |-------1-------1-|-1--------------|
    |-----2----------|-2--------------|
    |---2------------|-2--------------|
    |-0--------------|-0--------------|
    ```

    2. With key followed by '|' (valid)

    ```
    E|----------0-----|-0--------------|
    B|--------0---0---|-0--------------|
    G|-------1-------1-|-1--------------|
    D|-----2----------|-2--------------|
    A|---2------------|-2--------------|
    E|-0--------------|-0--------------|
    ```

    3. Does not contain '|' (invalid)

- Supported up to eighth notes (whole, dotted-half, half, dotted-quarter, quarter, eight)
- Tempo changes are not supported.
- Spaces between the staffs can be safely ignored. However, the strings must not have spaces between them (that is, all six guitar strings must have no spaces between them)

```
valid:
|--0-------------------|----------------------|--0--------------------|
|----------------3-----5-|-2--------------------|----------------3-----5-|
|----------------3-------|-2--------------------|----------------3-----5-|
|----------------5-------|-2--------------------|----------------3-----5-|
|----------------------|-0--------------------|----------------3-----5-|
|----------------------|----------------------|----------------3-----5-|

|--0-----------------|------------------|--0-------------|
|----------------5-|-2----------------|----------------|
|------------------|-2----------------|----------------|
|------------------|-2----------------|----------------|
|------------------|-0----------------|----------------|
|------------------|------------------|----------------|
|

|----------0-----|-0--------------|
|--------0---0---|-0--------------|
|-------1-------1-|-1--------------|
|-----2----------|-2--------------|
|---2------------|-2--------------|
|-0--------------|-0--------------|
```

```
invalid:
|--0-----------------------|------------------------|--0-----------------------|
|------------------3-----5-|-2----------------------|------------------3-----5-|

|------------------3-------|-2----------------------|------------------3-----5-|
|------------------5-------|-2----------------------|------------------3-----5-|
|--------------------------|-0----------------------|------------------3-----5-|

|--------------------------|------------------------|------------------3-----5-|

|--0-----------------|------------------|--0---------------|
|-----------------5-|-2----------------|------------------|
|--------------------|-2----------------|------------------|

|--------------------|-2----------------|------------------|
|--------------------|-0----------------|------------------|
|--------------------|------------------|------------------|
```

- Each measure must have the same spacing for each string; however, not each measure needs to be the same length. **Important: make sure there are no extra spaces at the end (common cause of runtime error) - fixed?**

```
valid:
|--0--------------------|--------|--0----------------|
|------------------3-----5-|-2------|-----------3-----5-|
|------------------3-------|-2------|-----------3-----5-|
|------------------5-------|-2------|-----------3-----5-|
|--------------------------|-0------|-----------3-----5-|
|--------------------------|--------|-----------3-----5-|
```

```
invalid:
|--0--------------------|--------|--0----------------|
|------------------3-----5-|-2----------|-----------3-----5-|
|------------------3-------|-2------|-----------3-----5-|
|------------------5-------|-2----------|-----------3-----5-|
|--------------------------|-0------|-----------3-----5-|
|--------------------------|----------|-----------3-----5-|
```

```
note the extra space at the end will cause an error
|-----------0-----|-0---------------|
|---------0---0---|-0---------------|
|-------1-------1-|-1---------------|
|-----2-----------|-2---------------|
|---2-------------|-2---------------|
|-0---------------|-0---------------|
```

- Repeated measure not correctly working.
- Editing and saving tabs inside console is not working correctly
- **Observe console output for error messages** (for now), it will be implemented to show in the GUI at a later time.

Launch Program

Once the application is launched from Gradle, press *Launch Program* in one of the menu buttons.

Selecting Different Documents

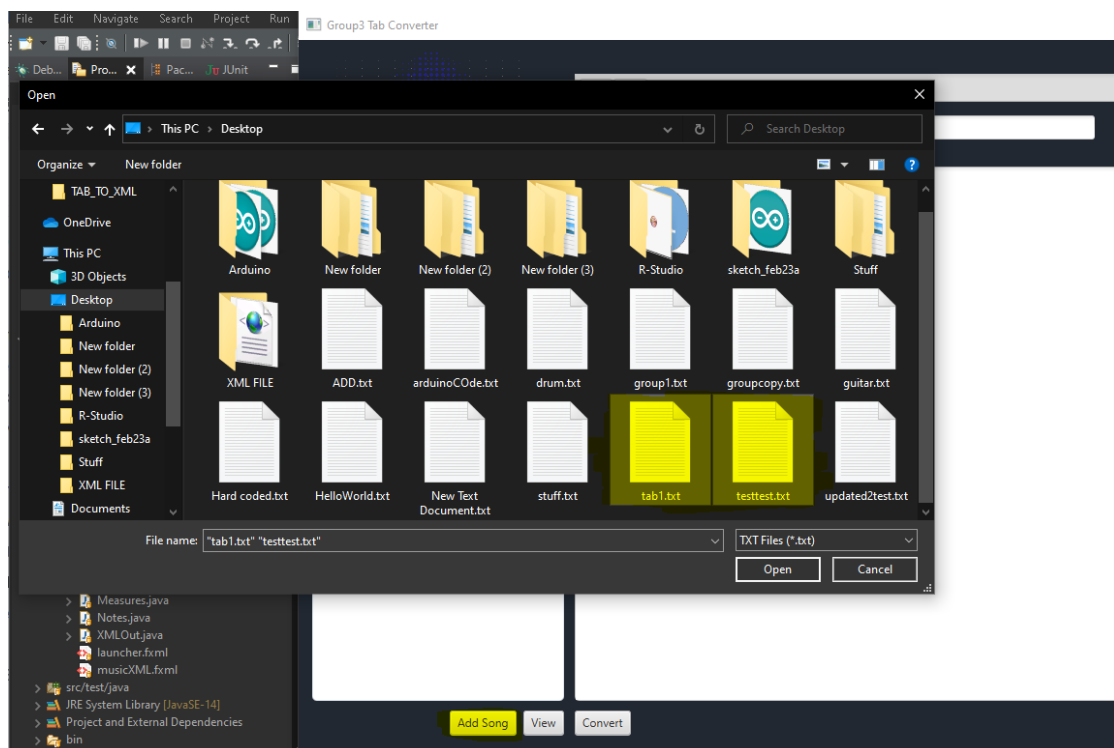Once the application is launched from Gradle, press any of the Document buttons

Selecting Files For Conversion

Users have the option to **select one or multiple files** to load into the program. All the tabs must be in the .txt format.

Steps for selecting file(s):

1. In the Footer menu in the menu bar click the "Add Song" button.
2. Browse your file system and locate the file to convert.
3. Click on a single file or drag the mouse over files to select them (or alternatively, gold down CTRL/Command key while selecting the files) and click "Open".
4. The name of the file(s) is now shown on the left side to be selected.

Preview Files/Copy and Paste

From the left-hand side of the interface, select the file to be previewed and then press the "View" button from the footer bar. The content of the file will be shown in the Text Panel on the right. You may also choose to copy paste a tab into the Text Panel. If you want to view another file, simply click on another file in the left-side list, the Text Panel will clear. Proceed to click "View" again to view the newly selected file.
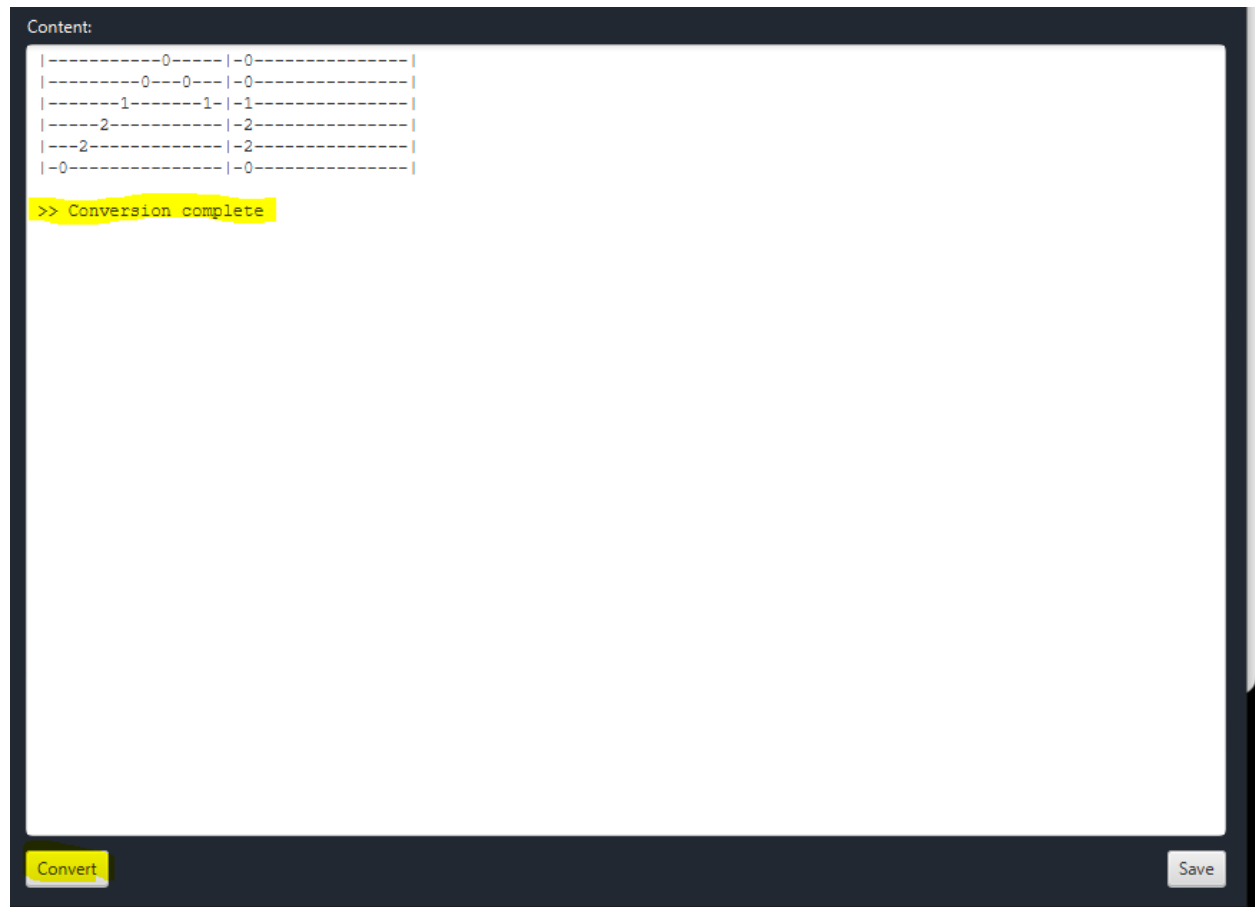
Copy and Paste method:
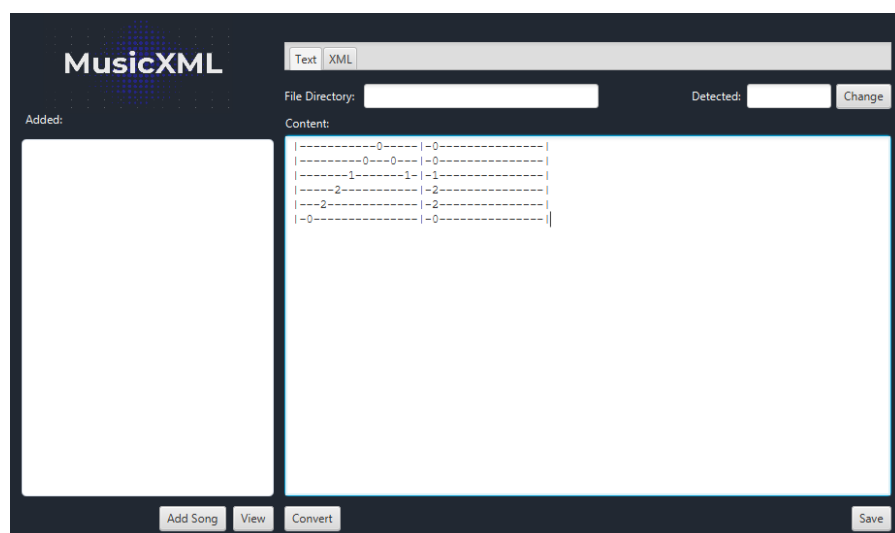
Selecting from imported list:
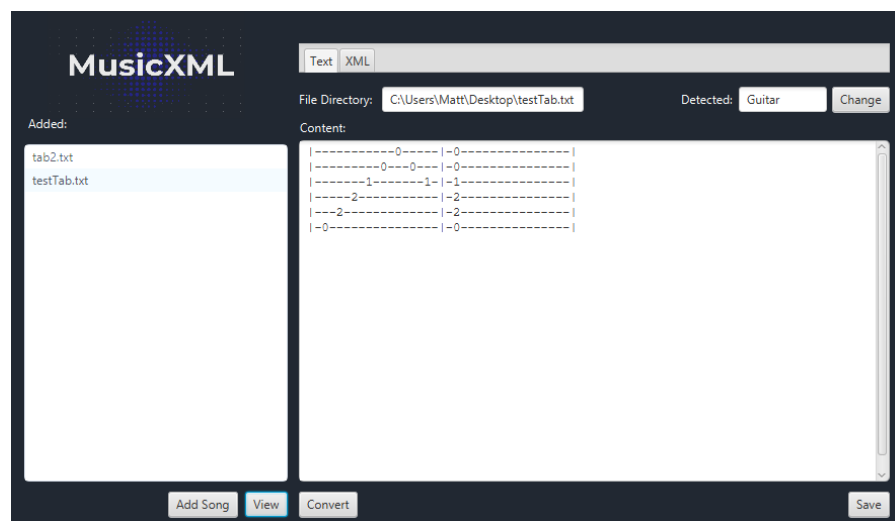
Convert File

On the footer bar, there is "Convert" Button. Upon pressing that button, the program will convert whatever is in the "Content" window into a XML file. If the file was converted correctly, a message will appear saying ">> Conversion complete"
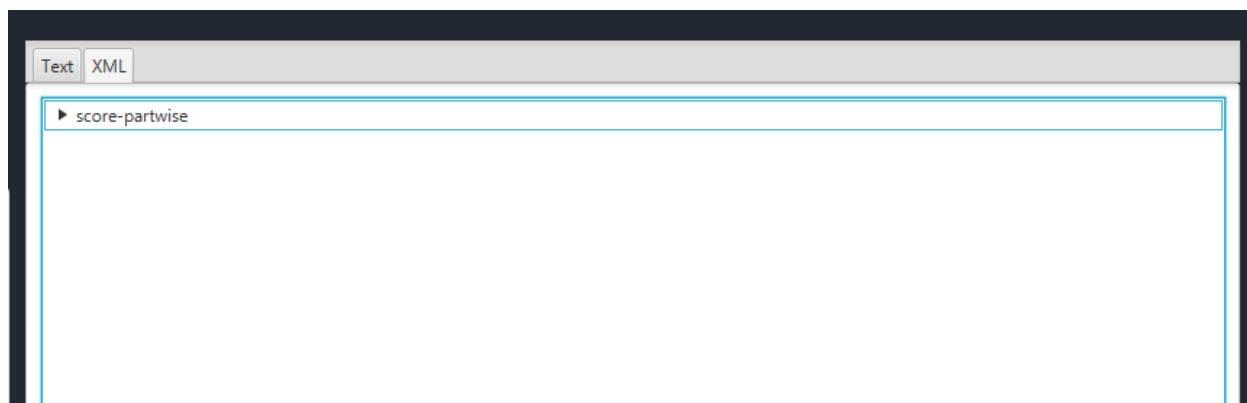

Save Files

1. Click the "Save" button in the footer bar or in the File menu.

2. Select the destination where you want to save the XML file. It will save as a .xml file.
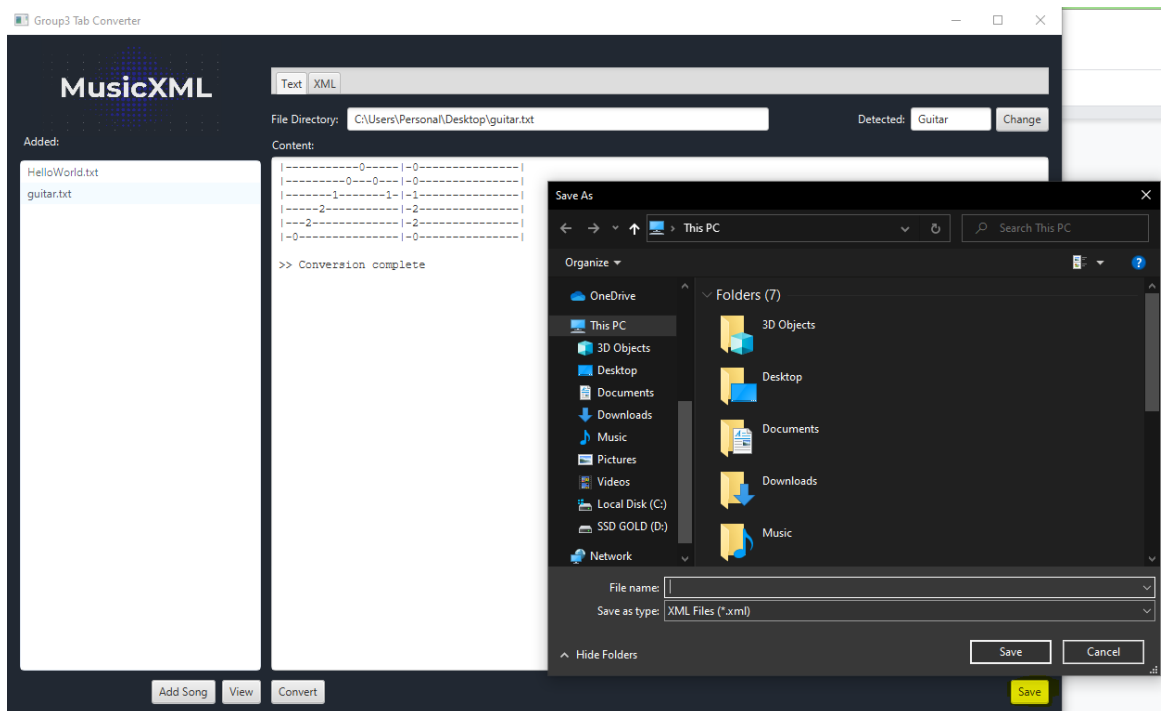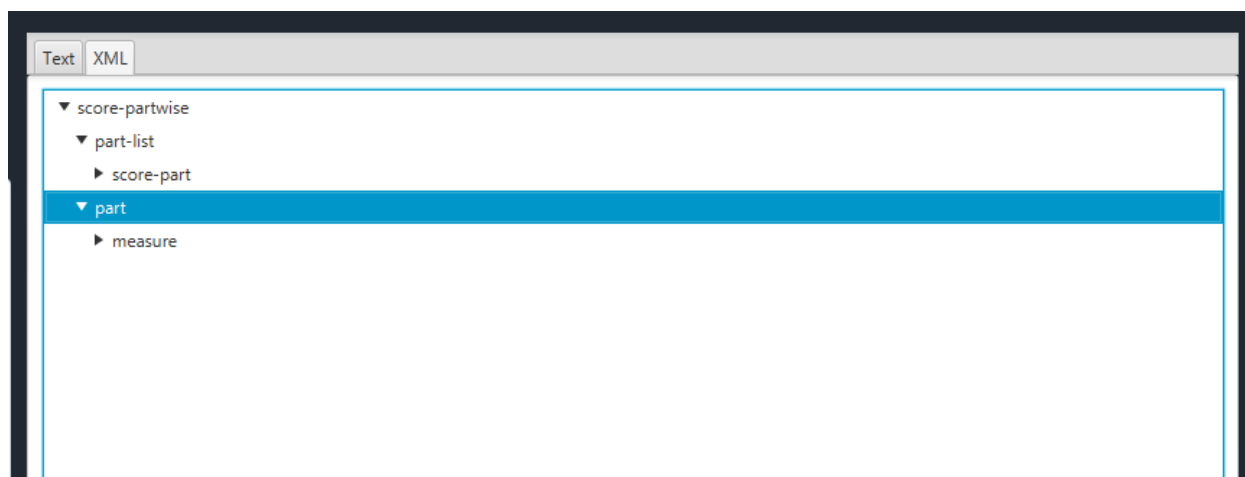
Viewing Converted XML File

1. Convert imported tablature
2. Navigate the header of the program and press "XML"

3. Press drop down menus to view full XML or specific sections

Success Scenarios

Scenario 1:

Testing Guitar tab provided on the class wiki

1. Either create a .txt file or copy and paste the tab into the program
    -   If imported from a .txt file, press view then convert
    -   If already copy and pasted and you can already see the tab, press convert
2. Once converted, press save and save to any location you want.
3. Navigate the class wiki and find the XML to Sheet Music converter
4. Drag the saved XML into the converter
5. Result should look like/similar to this:
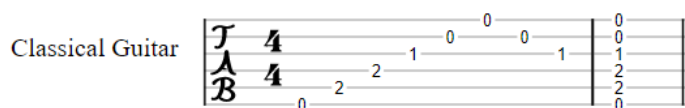
Scenario 2:

Any music of the users choice that meets the restrictions

1. Either create a .txt file or copy and paste the tab into the program
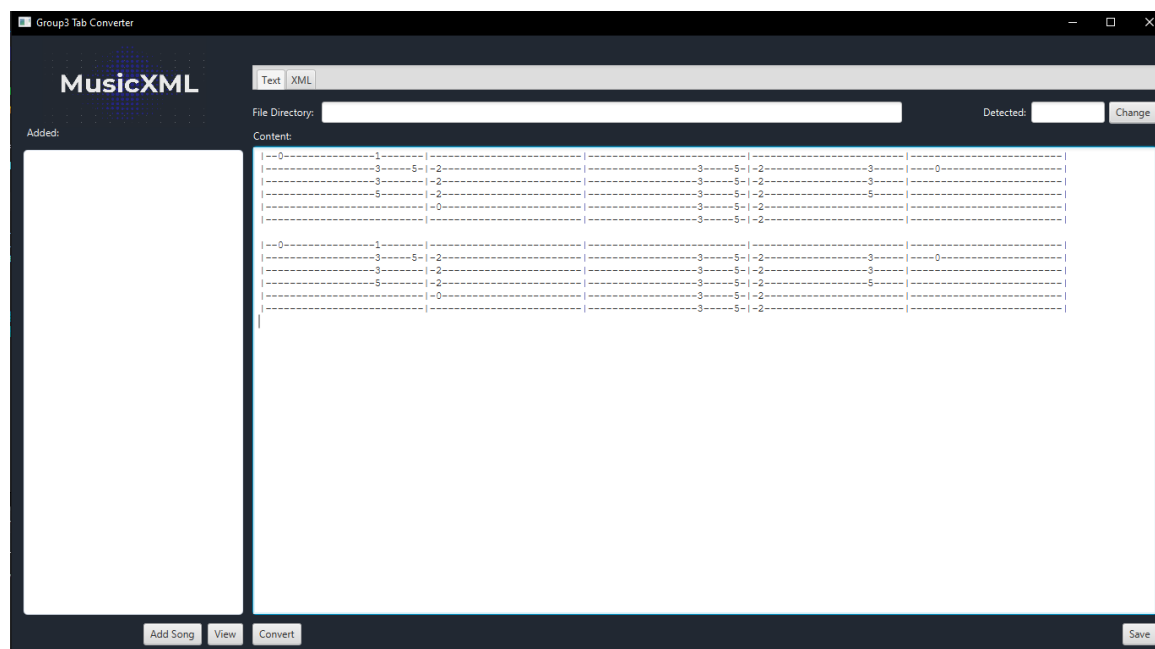   a. If imported from a .txt file, press view then convert

      b.   If already copy and pasted and you can already see the tab, press convert

2. Once converted, press save and save to any location you want.
3. Navigate the class wiki and find the XML to Sheet Music converter
4. Drag the saved XML into the converter
5. Self check if the tab meets requirements, example:
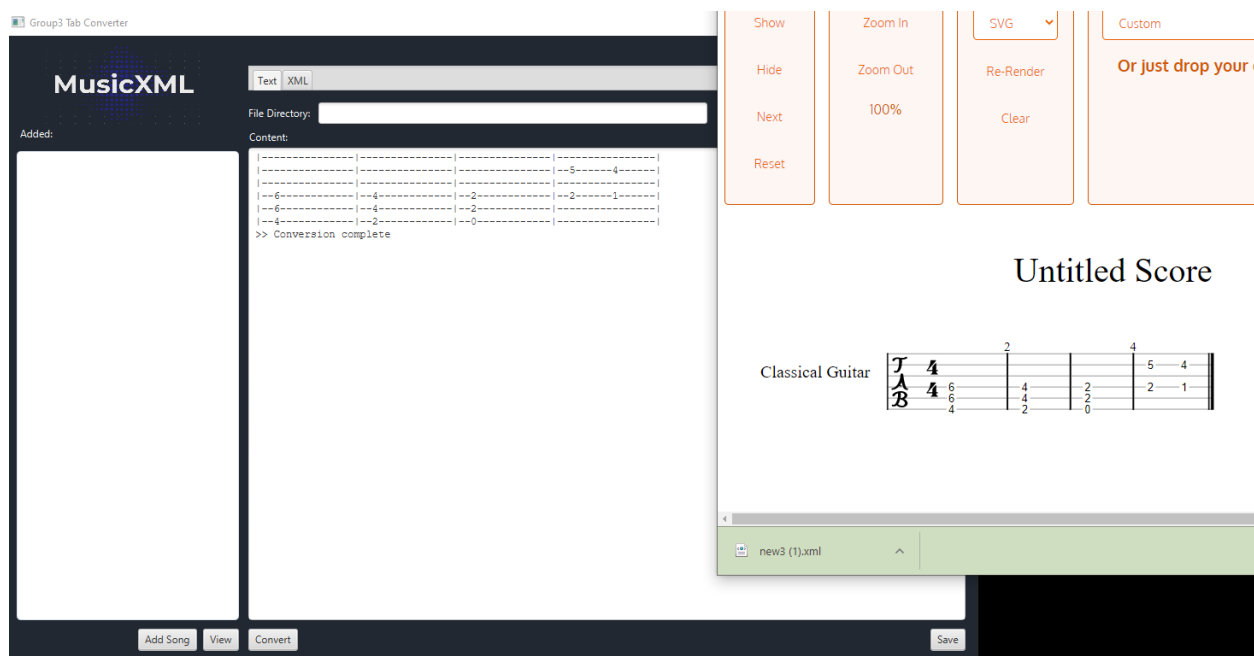
Scenario 3:

Combining different tabs together in Content and convert.

1. Copy paste 2 different tabs into the "Content" window
2. As we can see below, we can copy paste the text file twice and it will convert both tabs on the top and the bottom

Scenario 4:

Testing Drum tab provided on the class wiki

1. Either create a .txt file or copy and paste the tab into the program
   - If imported from a .txt file, press view then convert
   - If already copy and pasted and you can already see the tab, press convert
2. Once converted, press save and save to any location you want.
3. Navigate the class wiki and find the XML to Sheet Music converter
4. Drag the saved XML into the converter
5. Result should look like/similar to this

Troubleshoot / Help

Javafx Error:

https://www.youtube.com/watch?v=bC4XB6JAaoU

Java Version Error:

https://www.baeldung.com/java-lang-unsupportedclassversion

Unsupported File:

Check the Music Note Restrictions & Requirement Section above.

Java Linkage Error:

https://stackoverflow.com/questions/244482/how-to-deal-with-linkageerrors-in-java

Experimenting with Music Tabs

String Lines:

By placing six rows of dashes, the user can represent the string lines of a guitar fretboard. Then in order to add fret numbers, the user should erase the dash in the correct row and relative location within the row, then type the desired fret number. The user is responsible for aligning fret numbers that should be played together but in different rows, vertically in columns. Dashes should be in between bar lines and aligned such that there is the same amount of dashes/notes on each row.
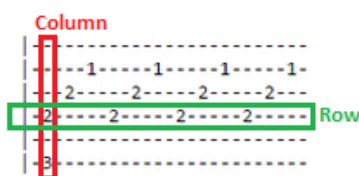
Experiment with the tabs and see what works! Thank you for using our program.

REFERENCES

EECS 2311- Group 1 user manual. (n.d.). Retrieved February 17, 2016, from http://markyc.github.io/Tab2PDF/user-manual/