# Typesetting Lectionaries Using SILE

## Introduction

Traditionally Bibles are translated and then typeset in book form with the books in canonical order. In some faith traditions the Biblical text is also read as a predetermined set of readings for each Sunday/service. This form is called a *Lectionary*. For Roman Catholics these lectionary volumes are produced in a three year cycle called Year A, B, and C.

This project is a MS Windows port of the excellent Sile typesetting tool. Some additional code has been added to do the lectionary specific typesetting operations.

This tool currently comes with the control files necessary to create Roman Catholic Lectionaries with headings in the Tok Pisin Language used in Papua New Guinea and text taken from a target language translation in Paratext. The same code could be used with other control files to produce lectionaries with headings in other languages or lists of references suitable for other faith communities.

The first part of the README discusses the technical steps necessary to typeset a lectionary. The last section of the README, *Project Considerations*, discusses some of the issues that need to be considered when planning and implementing the adoption of a lectionary.

## Status

The code has been succesfully used to produce lectionaries for Year A, B, and C for the Aruamu language of PNG. It mostly works.

## Installing the Lectionary Tool

1. Go to https://goo.gl/E5T30a and download the SileLectionary.zip file.

2. Unzip the file into its own directory. For the rest of this document we will assume you unzipped this in a directory called SileLectionary.

3. Run LuaFowWindows*.exe to install Lua for Windows.

## Creating a Lectionary Project in Paratext

**Note:** It would be possible to do the lectionary typesetting directly in the main Paratext project but it feels safer to me to create a separate project to ensure that nothing that is done working on the lectionary can impact the main translation project.

1. In Paratext use *File* > *New Project* to create a project to contain the lectionary. Give it the same language as the original project.
2. If the project you are typesetting is a NT only project and you wish to create a lectionary that includes OT/DC readings you will need to copy OT/DC from a language or wider communication. You will need copyright permission from the copyright holder to do this.
   - *Project* > *Copy Books* copying from language of wider communication to your lectionary project

3. *Project* > *Copy Books* copying all books from your target language project to the lectionary project
4. Create a module to hold the lectionary file for a year
    - *Tools* > *Open Bible Module*.
    - Set *Open in Book* to *Extra A*.
    - Choose *Copy from specification file*. Browse to the directory where you installed the lectionary tool and go to the *Modules* subdirectory.
    - Pick the control file corresponding to the year you wish to typeset, e.g. Year_A.sfm, B, or C.
        - There is also Year_A_NT.SFM (etc). This only has the NT references. That means you don't have to provide any text for OT references but the reader will have to juggle two different lectionaries to do the morning reading. That may make using your lectionary just "too much trouble" and limit its use. These files have not been very carefully checked yet.
5. **IMPORTANT:** Verify that all references are found by clicking on the icon in the top right of the winow (icon is a sheet of paper with a blue checkmark). This will also cause Paratext to pull the text of all the references and create the file that the typesetting software needs as its input.
6. If any references are missing you will need to either modify the references in your project text or modify the references in the module.
    - If you need to modify the lectionary control file you will find it in the *My Paratext Projects\XYZ\Modules* directory.
    - You will probably need to use an editor like Notepad to edit this file. In theory it could be edited in Paratext but this is difficult because when the Module support was originally created people were thinking of 10 page pamphlets and not 300 page lectionaries and so the in Paratext editing support is current buggy and slow for modules of this size.

# Typesetting a Lectionary Volume

1. Go to the SileLectionary directory.
2. To typeset XXA double click the file XXA.bat.
3. The first time you do this you will be prompted to enter the name of the project you are typesetting. This is the Paratext short name, e.g. TPLCT, of the Paratext project you setup in the previous section. This will be stored in the *ProjectName* file in the software directory. You can change this at any time by deleting the file causing the software to reprompt you for the project name.
4. Typesetting should begin.
    - WARNING: The first time you run this the program gathers information on all the fonts on your machine. This might take a minute or two.
    - Next, a list of the page numbers being typeset should be shown, e.g. [1] [2] ... . On my laptop it takes around ten minutes to typeset a 200 page volume.
    - The resulting .pdf will be in the *SileLectionary/lectionary* directory.

## Unknown Styles or Changing Style Formatting

1. The specifications for the styles used in the lectionary are found in the *lectionary/styles.sil* file. The contents of this file are described in The Sile Book. Chapter 6 might be the most relevant. If you feel lucky you might be able to just look at some similar style in the file and adapt that one.
2. Another possible option for dealing with unknown styles is to change them to a simpler known style in the lectionary project. We are not really aiming for fancy typesetting in the lectionary ... just that the text be layed out in a way that is relatively easy to be read out loud.

# Project Considerations

You will need to get formal permission from the copyright holders for the target language text in order to use it in a lectionary.

If the target language text translation is a New Testament (NT) only you will need to decide whether to

1. Print a lectionary with the NT only. This will make a smaller lectionary and be simpler to produce. The downside is that the lectionary reader will be forced to juggle two different lectionaries when reading, i.e. the NT lectionary you produce and the lectionary in the language of wider communication used to read the OT and DC readings. Forcing the reader to juggle two books may well discourage people from using your lectionary at all. -OR-
2. Print a lectionary with the NT readings from the target language text and the OT and DC readings from another text, in PNG for example you might use the Tok Pisin text. If you are going to use readings from a another text you will need to get formal copyright permission from the owner of that text.

You will normally need to get formal permission from leaders in the faith community for the use of the produced text. In the case of the Aruamu lectionary this was the local leaders and the Archbishop. They may want to review with you how certain doctrinal issues have been handled in your translation.

You will need to discuss with leaders in the faith community whether they want lectionary headings to be in the language of wider communication or to be in the target language. If some of the readers are not native speakers of the target language, it may be better for Scripture Use reasons to leave the headings in the language of wider communication. There are some things that Aruamus decided to leave in Tok Pisin, because the non-Aruamu speakers do not want to be unable to find the place where the readings are, etc. Research should be done on this, ASKING the right people.

If you decide to change the heading to the target language, these will all need to be translated, checked and typed in. This is not simple. There are hundreds of them.

There are some dramas in the Lectionaries around Easter time. These are just scripture, with different people reading the parts of different characters. If these are wanted in the target language (which given the popularity of drama is not unlikely) they have to be translated "by hand"; someone has to go into the text and cut and paste each speakers part into the right place in the drama. It is pretty fiddly, labor intensive; no current way to automate.

There are also some issues with the fact that, because Lectionaries pull verses out of larger context, sometimes things need to be inserted or deleted in order to make sense. Sometimes connectors need to be added, or to be entirely different that what would be published in a Bible. Also, there are places whens the Lectionary will use parts of verses. So, in order to pull the text correctly from a vernacular project (or Tok Pisin) the text will need to be marked to do that.

Currently you can put things like the following the text: {He|Jesus} said ... This will cause the lectionary to read "Jesus said" but allow the typesetter to change this to just "He said ..." before doing the typesetting. This system is not ideal. It makes things more complicated and messes up checking. Alternately you could just edit the text in the lectionary project ... of course the downside to this is that if you want to create an updated version later you would have to repeat the edits, sigh.

In the longer term I think we are going to want to allow the following in the target language translation

```
\rem subst "He said" => "Jesus said"
\p He said, ...
```

By putting the adaptation in a \rem (remark) statement the normal translation/typesetting procedures would not be impacted by the lectionary changes.

## Source Code

This project is hosted at https://github.com/Nathan22Miles/SileLectionary.git and is freely available subject to the licensing restrictions of the SILE code it is based on. The build process is extremely ugly because the component parts were drawn from a wide variety of sources and there has not been time to put them into any kind of common framework. Sadly the Lua code was the first Lua code I ever wrote and looks like it. I can think of a half dozen other things to apologize for about this project the was done way too fast but instead I will just request that you to evaluate it by the "The Dancing Bear Standard", i.e. the amazing thing about a dancing bear is not how gracefully it dances but that it manages to dance at all.