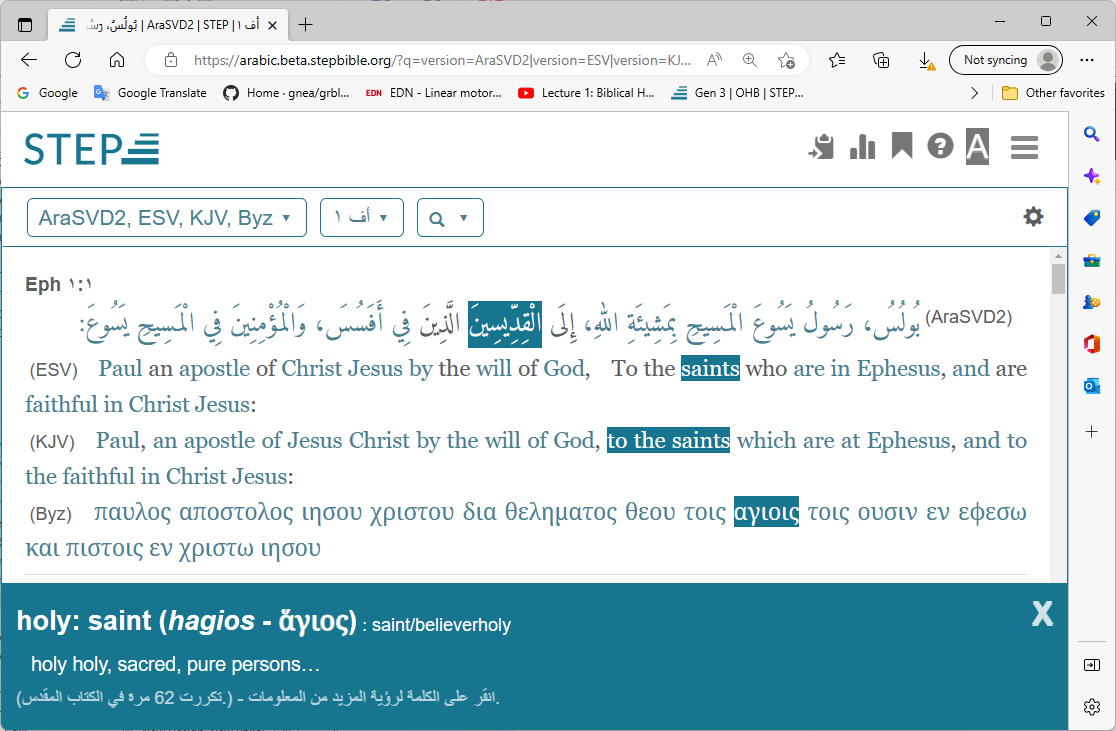
**Tagging Bible Text**

# Overview

This project was started after I used [STEPBible](https://www.stepbible.org/) and recognized the great benefit of being able to hover over or click on Bible words, for specific versions, to get a wealth of information about them. It is a great study tool. The secret behind this capability, besides the clever programming driving the web site, is that the words of those versions are tagged (associated with) coded numbers to tie them to certain lexicons and other resources. The simplest tagging is using [Strong’s numbers](https://en.wikipedia.org/wiki/Strong%27s_Concordance#:~:text=in%20the%20Bible.-,Strong%27s%20numbers,as%20the%20%22Strong%27s%20numbers%22.).

The version of Bible that I use, the Arabic Smith & Van Dyck version, did not have such tags and I wondered, what does it take to tag the Arabic Bible? I wrote a small program to help me tag the Arabic words based on King James Version tagging. If I continued in this direction, it would’ve taken me a few years to complete the task. This is when Rev Dr. David Instone-Brewer told me about his process of Automatically tagging a Bible of virtually any language. I then used his process which successfully tagged over 80% of the Arabic Bible. The result is shown in the screenshot below:



In the above screen shot, the Arabic text of Eph1:1 is displayed along with the same verse from two English translations and an original language (Greek) version. By Hovering the mouse pointer over the word “الْقِدِّيسِينَ”, it automatically highlights the equivalent word in both English translations, and in the Greek. It also shows the word meaning at the bottom. Clicking on the same word, brings up even more details, but not shown here because it is not the purpose of the screen shot.

At this point, I started using the program that I originally developed, to review and complete the tagging. This document describes the process used, and associated programs.

# Acknowledgment

This project wouldn’t have been possible without the help, support and guidance of Rev Dr. David Instone-Brewer. He provided me with the necessary material and patiently answered all my questions.

# Process Summary

# Process Flow Diagram



# Process

1. Diacritics are grammatical tool to ensure the meaning is conveyed correctly. For example using different diacritics can change the word from being a subject of a verb to being an object without changing the position of the word in the sentence. They cause a lot of noise in the text which can hinder the efficient use of Partext Interlineariser and the Berkley aligner. So, the first step is to remove all diacritics and punctuations from the text. This is done using a small c# program: CleanArabicText.

Example:

|  |
| --- |
| Original text:  Gen 1:1 فِي الْبَدْءِ خَلَقَ اللهُ السَّمَاوَاتِ وَالأَرْضَ.  After running the program  Gen 1:1 في البدء خلق الله السماوات والارض |

1. Next, we need to generate the morphology file from Partext Interlineariser. This will help us identify the stems of the Arabic words. Paratext requires the files to be in USFM (Unified Standard Format Markers) format, therefore we use a small c# program: Convert2USFM. to convert the cleaned Arabic text into USFM format Bible books. The first section of Genesis USFM file looks like so:

|  |
| --- |
| \id GEN Smith & Van Dyck Arabic translation  \ide UTF-8  \h تكوين  \p  \c 1  \p  \v 1 في البدء خلق الله السماوات والارض |

1. In Paratext, we create a new project (one project per Testament) and add to it all the Bible books from step 2. Then, start the Inerlineariser. This will automatically generate a morphology xml file.
2. Now, we can use the morphology file from step 3, and the clean text from step 1, and use a small c# program: StemmingArabicText to reduce the Arabic words to their stems.

|  |
| --- |
| Noe the above verse becomes  Gen 1:1 في البدء خلق الله سماوات الارض |

1. We need to Generate separate file containing Strong’s numbers for replacing each corresponding Hebrew word. We call this the Tags file, since it will ultimately be the tags in the text. Use a small c# program: GenerateHebrewAndTags & GenerateGreekAndTags.

|  |
| --- |
| The corresponding Tag verse to Gen 1:1  Gen 1:1 7225 1254 0430 8064 0776 |

1. Use the program: GenerateAlignerFiles to mainly put a version of the stemmed Arabic and the tags file in the Berkley Aligner train folder, after removing the references from the beginning of each verse.

|  |
| --- |
| So, the above becomes:  في البدء خلق الله سماوات الأرض  and  7225 1254 0430 8064 0776 |

1. Run the Aligner to generate the h-a-align file and g-a-align.

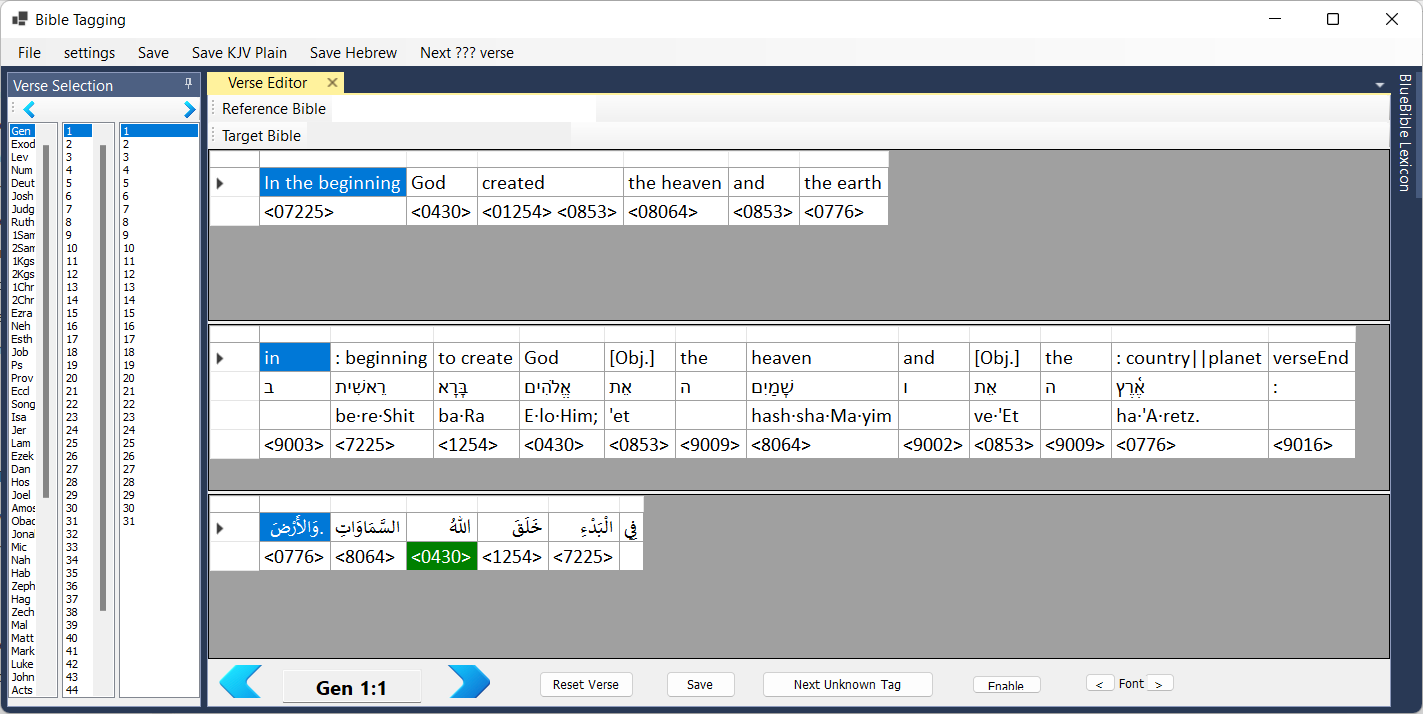
java -server -mx1000m -cp berkeleyaligner.jar edu.berkeley.nlp.wordAlignment.Main ++myConfs/arabicot.conf

java -server -mx1000m -cp berkeleyaligner.jar edu.berkeley.nlp.wordAlignment.Main ++myConfs/arabicnt.conf

1. Use the AlignerMapping program to generate the Arabic tagged text.

|  |
| --- |
| The aligner mapping for G1.1 will look like this  1-0 5-4 4-3 3-2 2-1  In each pair of numbers, the first represents the Hebrew Arabic number and the second represents corresponding tag word number. The first word in the line is word 0. Notice that Arabic is displayed right to left, so, the rightmost word is word 0.  Arabic line Hebrew line  0 في Maps to nothing  1 البدء Maps to 0 7225  2 خلق Maps to 1 1254  3 الله Maps to 2 0430  4 سماوات Maps to 3 8064  5 الأرض Maps to 4 0776 |

1. The verse editor can be used to review each verse and make correction as necessary.



1. In order to use Sword module generator, we need to covert the Arabic text to OSIS format using the GenerateOSIS program.
2. Finally use osis2mod to generate the Arabic module.

# Directory Structure