Generating ETCBC4

Claus Tøndering

13 January 2023

1 Introduction

This document gives a brief introduction to how to generate the ETCBC4 Emdros database for Bible Online Learner on a Linux computer.

2 Prerequisites

In order to execute this process, you need to clone the GitHub repository `https://github.com/-EzerIT/ETCBC4BibleOL.git`. In the following description I assume that you have cloned the repository into a folder called `ETCBC4BibleOL`.

You also need a version of the original ETCBC4 Emdros database from the Eep Talstra Centre for Bible and Computer. I use what I believe to be version 1.29 from January 2017.

In the `ETCBC4BibleOL` you must locate this line in the Makefile:

```
BHS4=../bhs4/bhs4 # Location of source Emdros database
```

and replace "`../bhs4/bhs4`" with the path name of the original ETCBC4 database.

3 Generating ETCBC4

Executing the command "make" in the `ETCBC4BibleOL` folder should generate the ETCBC4 Emdros database for Bible OL. This takes slightly more than an hour on my computer.

The steps of the process are detailed in the following sections.

3.1 Compiling *emdros_updater*

The C++ source code for *emdros_updater* is compiled. A C++ compiler that supports the "-std=c++20" flag is required.

3.2 Executing *emdros_updater*

Inputs:

- The original ETCBC4 database
- `ETCBC4-frequency4.04_progression-heb.csv`
- `ETCBC4-frequency4.04_progression-aram.csv`

Output: `update.mql`

The files `ETCBC4-frequency4.04_progression-*.csv` are taken from the Excel file `ETCBC4-frequency4.04_progression.xlsx` created by Oliver Glanz. They contain information for the *fa_order* and *verb_class* features.

The *emdros_updater* program generates data for the following Emdros features:

- fa_order
- frequency_rank
- g_lex_cons_utf8
- g_lex_translit
- g_nme_cons_utf8
- g_nme_translit
- g_pfm_cons_utf8
- g_pfm_translit
- g_prs_cons_utf8
- g_prs_translit
- g_suffix_translit
- g_uvf_cons_utf8
- g_uvf_translit
- g_vbe_cons_utf8
- g_vbe_translit
- g_vbs_cons_utf8
- g_vbs_translit
- g_voc_lex_cons_utf8
- g_voc_lex_translit
- g_word_cons_utf8
- g_word_nocant
- g_word_nocant_utf8
- g_word_nopunct_translit
- g_word_nostress
- g_word_nostress_utf8
- g_word_translit
- lex_cons_utf8
- lexeme_occurrences
- monad_num
- qere_translit
- verb_class

The execution of *emdros_updater* takes considerable time, and the progress through the books of the Bible is shown on the console.

Note that *emdros_updater* may warn about illegal final characters in a few cases. The warnings may look like this:

```
Bad final character corrected: בֵּינֶיךָ GEN 16,05
```

In my case I get three of these warnings. They indicate errors in the original ETCBC4 text.

3.3 Applying `update.mql`

Inputs:

- The original ETCBC4 database
- `update.mql`

Output: Updated ETCBC4 database

The "mql" command is executed and updates ETCBC4 with the information generated in the previous step. During this process the "mql" command outputs more than two million lines of text to the console.

3.4 Executing *change_mql.sh*

Input: The ETCBC4 database
Output: Updated ETCBC4 database

The program "mqldump" is executed on the ETCBC4 database, the output is piped through the shell script *change_mql.sh* which adds "FROM SET" and "WITH INDEX" to a number of features.

Finally, the ETCBC4 database is recreated from the updated MQL file.

3.5 Executing *fix_lower_dots.sh*

Input: The ETCBC4 database
Output: Updated ETCBC4 database

This shell script *fix_lower_dots.sh* is executed. It changes the lower dots of the first word of Ps 27:13 from Unicode character U+0323 to U+05C5.

3.6 Executing *extra_tenses/make_update.sh*

Inputs:

- The ETCBC4 database
- `extra_tenses/cohortative.mql`
- `extra_tenses/emphatic_imperative.mql`
- `extra_tenses/jussive.mql`

Output: Updated ETCBC4 database

The shell script *make_update.sh* in the folder *extra_tenses* is executed. It uses three MQL files in that folder to identify certain verbs and adds the extra verbal tenses *juss, coho,* and *emin* to the verbs. The MQL files were created by Oliver Glanz.

At this point the ETCBC4 database is finished.

3.7 Compiling *worddb*

The C++ source code for *worddb* is compiled.

3.8 Executing *worddb*

Input: The ETCBC4 database.
Output: The `ETCBC4_words.db` database.

The *worddb* program is executed. It creates the so-called "words database". For information about this database, see the chapter *Multiple-Choice Questions* in the Bible OL technical documentation.

3.9 Compiling *hintsdb*

The C++ source code for *hintsdb* is compiled.

3.10 Executing *hintsdb*

Input: The ETCBC4 database.
Output: The `ETCBC4_hints.db` database.

The *hintsdb* program is executed. I creates the so-called "hints database". For information about this database, see the chapter *Hints* in the Bible OL technical documentation.

4 Generating Lexicons

There are currently programs available for generating these lexicons:

- Aramaic-English
- Hebrew-Spanish

Executing the command "make build_aram_lex_file" in the `ETCBC4BibleOL` folder compiles the program *build_aram_lex_file*. The program is then executed. It takes input from `ETCBC4-frequency4.02_progression-aram.csv` (provided by Oliver Glanz) and generates `aram_en.csv` which can be uploaded to Bible OL as an Aramaic-English dictionary.

Similarly, executing the command "make build_heb_es_lex_file" in the `ETCBC4BibleOL` folder compiles the program *build_heb_es_lex_file*. The program is then executed. It takes input from `BibleOL_dictionary_Hebrew-Spanish_v1.2.csv` (provided by Oliver Glanz) and generates `heb_es.csv` which can be uploaded to Bible OL as a Hebrew-Spanish dictionary.

These two programs can be used as templates for creating other lexicons.