NAME

tlgu – convert TLG (D) CD-ROM txt files to Unicode

SYNOPSIS

tlgu [*options*] *input_file output_file*

DESCRIPTION

tlgu will convert an *input_file* from Thesaurus Linguae Graeca (TLG) representation to a Unicode (UTF-8) *output_file*. The TLG representation consists of **beta-code** text and **citation** information.

OPTIONS

–b inserts a form feed and citation information (levels a, b, c, d) on every "book" citation change. By default the program will output line feeds only (see also **–p**).

–p observes paging instructions. By default the program will output line feeds only.

–r primarily Roman text. Some TLG texts, notably doccan1.txt and doccan2.txt are mainly roman texts lacking explicit language change codes. Setting this option will force a change to roman text after each citation block is encountered.

–v highest-level reference citation is included before each text line (v-level)

–w reference citation is included before each text line (w-level)

–x reference citation is included before each text line (x-level)

–y reference citation is included before each text line (y-level)

–z lowest-level reference citation is included before each text line (z-level).

–B inserts blank space (a tab) before each and every line.

–C citation debug information is output.

–S special code debug information is output.

–V block processing information is output (verbose).

–W each work (book) is output as a separate file in the form output_file-xxx.txt

HISTORY AND INTENDED USE

The purpose of **tlgu** is to translate binary TLG-format files into readable and editable text. It is based on an earlier program written in 80x86 assembly language (1996) outputting codes for a home-made font which used the prevalent hellenic font encodings of that time complemented by dead accent characters - not very attractive, but readable.

Then came Unicode and a plethora of accented character glyphs; nice-looking but with the well-known drawback that special processing is needed to do wild-card searches. Nice polytonic fonts have now been made available (Cardo, Gentium, Athena, Athenian, Porson) and, surely, these will be expanded as special-use code points are included in the Unicode definition (musical symbols, other special symbols) and more fonts will be created.

So, at this point in time, **tlgu** will crunch a file which has been formatted according to the published TLG-D format and produce codes for most glyphs generally available. No attempt has been made to introduce multi-character sequences or formatting codes (font changes). If a code has not been defined, the program will output the respective "code family" glyph. You may use the **–S** option to check such codes against the published beta code definition.

You may not like the character output for a specific code. Check out the **tlgcodes.h** file containing the special symbol and punctuation codes and select one to suit you better. It will probably be a while before the beta to Unicode correspondence settles down.

EXAMPLES

./tlgu -r DOCCAN2.TXT doccanu.txt Translate the TLG canon to a unicode text file. Note the use of the **-r** option (this file expects Roman as the default font).

./tlgu -x -y -z TLG1799.TXT tlg1799u.txt
Generate a continuous file with the texts of granpa Euclides. Available citations (-x -y -z) are Book//demonstratio/line as shown in the respective "cit" field of doccan2.txt.

./tlgu -b -B TLG1799.TXT tlg1799u.txt
Generate the same texts, this time with a page feed and book citation information on the first page of each book and a tab before each line (use with OOo versions earlier than 1.1.4).

./tlgu -C TLG1799.TXT tlg1799u.txt
See how the citation information changes within each TLG block.

./tlgu -S TLG1799.TXT tlg1799u.txt | sort > symbols1799.txt
Check out the symbols used in a work. Book and x, y, z references are printed on a separate line for each symbol. Sort / grep the output to locate specific symbols of interest; save in a file for later use.

./tlgu -W TLG0006.TXT tlg0006u
Will produce separate files for each work, named tlg006u-001.txt etc.

POST-PROCESSING EXAMPLES

I use the OpenOffice suite for most of my work. This example shows one of many possible ways of using the search and replace facility to create a readable version of the Suda lexicon.

./tlgu -B TLG4085.TXT tlg4085u.txt
A Unicode file with the text is created

Open the generated file with OOo:
File | Open | Filename: tlg4085u.txt, File Type: Text Encoded -- Press Open

The ASCII Filter Options window appears. Select the Unicode (UTF-8) character set and a proper Unicode font installed in your machine (e.g. Cardo). Press OK.

Replace angle brackets with expanded text
Lexicon terms are enclosed in <angle brackets>. The actual beta codes indicate the use of expanded text for emphasis. Select Edit | Find & Replace. The **Find & Replace** window appears.

In the **Search For** field, type the following expression: **<[^<>]*>** This means "find any characters between angle brackets, not including angle brackets".

In the **Replace With** window insert a single ampersand: **&** This means that we need to **add** formatting information (this case) or additional text to the text found. Press **Format...** and select the **Position** tab; select Spacing Expanded by 2.0 points. Press OK.

Check the **Regular Expressions** box and press **Replace All**.

You may now replace the angle brackets with nothings.

Repeat the above procedure for titles enclosed in {braces}. Write a macro...

Other useful information

In the "Execute" tab of the "Properties" window of my KDesktop Link to Application I have the following command (single line):
LC_CTYPE=el_GR.UTF-8 /whereitsat/OpenOffice.org1.1.x/soffice
The prefix, an environment variable, allows you to use the same program with different locales; in this case, hellenic Unicode (UTF-8).

I put my default locale and keyboard definitions in my **.profile**:
export LC_CTYPE=el_GR.UTF-8
setxkbmap us+el polytonic -option grp:ctrl_shift_toggle

This way multi-lingual text can be entered; keyboard layout switching is done by pressing Ctrl/Shift.

REFERENCES

There are several texts describing the internal representation of **PHI** and **TLG** text, ID data, citation data and index files. The originator of this format is the Packard Humanities Institute. The TLG is maintained by UCI – see **www.tlg.uci.edu** – where you may find the **TLG Beta Code Manual** and the **TLG Beta Code Quick Reference Guide**.

Unicode consortium publications pertaining to the codification of characters used in Hellenic literature, scientific and musical texts.

The OpenOffice suite (**www.openoffice.org**) includes a word processor that you can use to load, process and create new polytonic texts.

COPYRIGHT