Interlinear User Guide

Stephen Palmstrom 30 January 2015

# Introduction

After documents have been converted from legacy fonts to Unicode, it is useful to compare the legacy and Unicode documents to see how successful the conversion has been. This can be done by looking at the documents side-by-side, but this can become difficult, particularly as there may be subtle differences between the fonts that may change the pagination. It is therefore useful to be able to see the two documents in one place, line-by-line. This program does this.

This document describes how to use Interlinear, a program that takes two Microsoft Word documents, generates segmented files (i.e. files with a fixed number of words per line) and copies them into an Excel spreadsheet that gives a line-by-line comparison of the two documents.

# Licensing and Copyright

The program was developed for MissionAssist, so it is copyright to that organisation, but distributed under the GNU General Public License (<http://www.gnu.org/licenses/gpl.txt>).

# Installation

Open the Interlinear folder in the zip file and execute setup.exe. Until the code can be digitally signed, Windows 8 may throw up security warnings that you will have to ignore.

# Execution

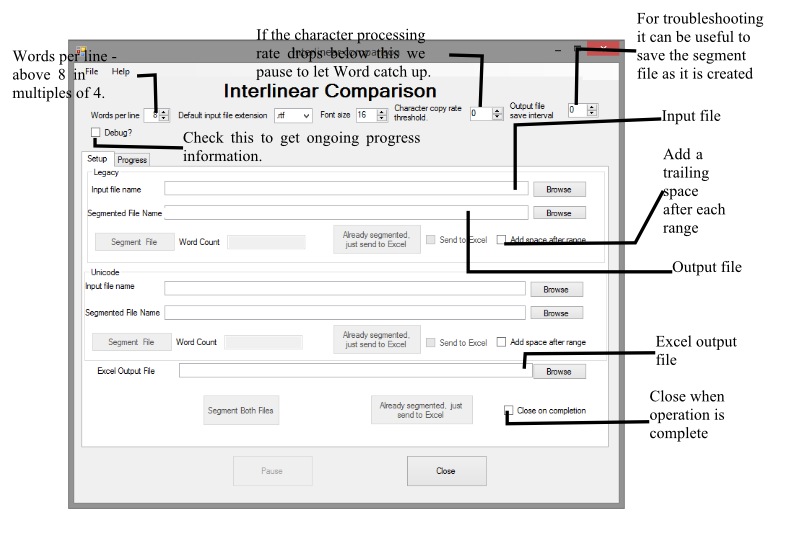
After installation, you will see this:

Figure 1 The program when it first opens.

Click the Browse buttons to insert the file names for legacy, Unicode and the Excel spreadsheet. The buttons allowing you to segment the files and create the Excel output will be enabled as you add the file names. The installation process is supposed to put this in your start menu. This means that the *Already segmented, just send to Excel* buttons and the *Send to Excel* tick boxes will only become enabled when you give the program an Excel file name to which to send the data and the former when the segmented files exist.

Once you have entered the file names you can click on the Segment buttons. If the Excel file name has been entered, the default is that the segmented data are sent to Excel. If you have all the file names completed, the Segment Both button will do segment and create the Excel file in one operation. The progress tab reports progress after you have started the process. Unfortunately, because the program is doing much copying and pasting, it runs very slowly at present, though we hope to improve performance in future.

The program now remembers the last folders you used for input and output.

# The Excel File

The Excel file consists of one worksheet with the legacy and Unicode text interleaved.

# A glimpse under the bonnet

This application creates hidden instances of Microsoft Word and Excel. If instances are already open, Interlinear will connect to them and remember that it has done so. It will make them invisible while it runs to improve performance, and make them visible again when it finishes. If the application {Word or Excel) is not running before you start, it should shut down when Interlinear exits, but in the event of a program crash, they may remain running and have to be shut down manually using Task Manager.

# Modus Operandum

This program moves through all the stories in Word and copies the text and corresponding fonts to a new document. It remembers any text boxes it finds for future use. It then goes through all the stories again and looks for shape ranges that have text. It copies the text and corresponding fonts for any text boxes not found in the first pass (by checking identity of text and font against the list it held from the first pass). This is because it did not find text boxes in headers and footers during the first pass. This is the same technique as used by the Character Counting program.

The program then cleans up the text, removing text boxes (there may not be any at this point in any case), converting tables to text (again this may not be necessary, but we did this before changing the copying technique), removing paragraph markers and replacing with spaces and also removing various control characters. It chops up the text into segments of words, the number selected by you, but the default is eight. It finally selects and copies and pastes the paragraphs into Excel, the legacy text in odd rows and Unicode text in even rows, line by line to preserve the various fonts that may change mid-word.

# A note about Symbols

If characters have been inserted using the Insert Symbol function, this fact is ‘hidden’ and they appear when viewed in Word to have the same font as the surrounding text. You can determine that they are inserted symbols by clicking on them, when the Insert Symbol dialogue box will appear. We are able to find and process them with this program by using the Range.get\_XML function to get the XML of the range and looking for w:sym nodes. If we find them, we then analyse the xml and insert the relevant symbols using the Selection.InsertSymbol function. Unfortunately, this slows the program.

# A note about file corruption

We have seen a Rich Text Format file where a paragraph was corrupted so when the program read it word by word it found itself in an infinite loop. This could be cured by saving it as a Word 2007+ file and, if necessary do an open+repair on it. The program now catches such issues and warns you to that effect.