Character Counter 4.0

Stephen Palmstrom 3 December 2013

# Purpose

Figure 1 Individual analysis

As part of checking the conversion of legacy fonts to Unicode, it is useful to obtain statistics for the numbers of different glyphs and their respective fonts. This program will analyse Microsoft Word documents including Rich Text Format and produce an Excel workbook of the various glyphs and their fonts. You can then compare the legacy and Unicode files to check that the conversion has worked satisfactorily. It will also analyse text documents (anything that does not have the .doc, .docx or .rtf extensions). It is possible to analyse several documents and obtain an aggregate count of the various glyphs and their fonts. You can either analyse one document at a time or do a bulk analysis.

# 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>).

# System Requirements

The program uses both Microsoft Word and Excel in its analysis, so needs both programs, and runs under Microsoft Windows (7 or above).

# Architecture and Troubleshooting

The program opens the Word or text documents you specify and goes through them character by character counting each one, but distinguishing their respective fonts. It gives you the option of counting the decomposed characters that make up individual glyphs or graphemes as units. It also checks, if the font is Unicode, if the glyphs are normalised, i.e. if they are represented as a single characters or the smallest number of characters possible and suggests improvements.

The program invokes hidden instances of Word and Excel. In the event of it crashing or being terminated abnormally, you will have to use Task Manager (Right-click on the taskbar and select Task Manager to run it) or its equivalent to terminate those instances. If you terminate one of them using Task Manager or because it becomes visible, the program may crash, though there is error trapping so if, for example, Word has been closed, the program will attempt to reopen it. Similarly for Excel. If the Excel workbook you want to write to is open, the program can’t delete it before writing, but it will let you close the workbook and try again.

Figure Bulk Analysis

You can also analyse text documents, i.e. documents without the .doc, .docx or .rtf extensions, but the program can’t determine the font with which to display the glyphs.

If you give it an Excel file name in the Aggregate stats file box, it will default to accumulate aggregate statistics on a number of files. If you use the bulk option you can select the files you want to analyse in a given folder.

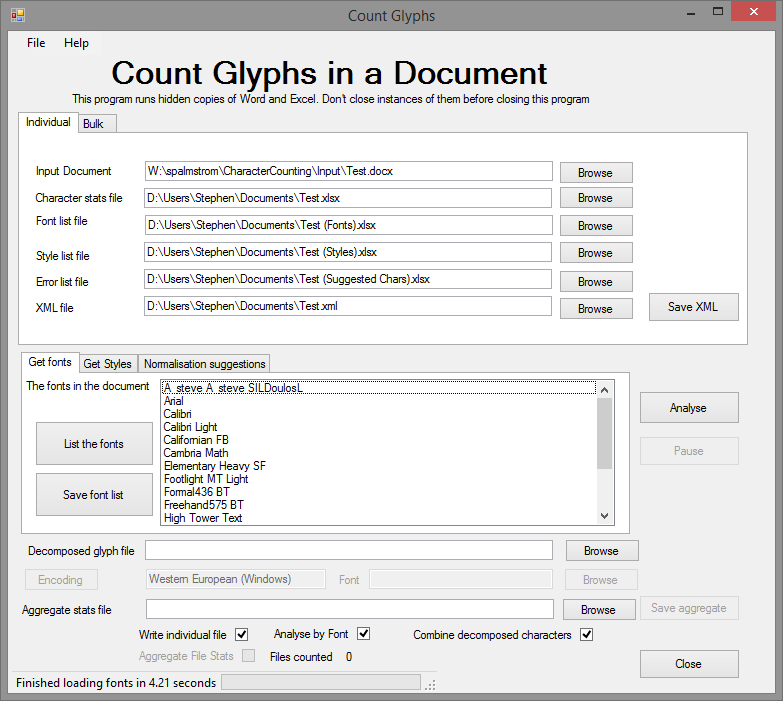
# Installation

Install the program by clicking Setup.exe in the CharacterCounter folder. A link to it should then appear in the Start Menu. The program will also run immediately after installation.

# Execution

Click on the icon in the start menu to execute. By default, it comes up for analysing files one at a time, but clicking on the Bulk tab will allow you to analyse several files at once.

Browse to select the Word document you wish to analyse. The Excel output file will default to a file of the same name in the Documents folder. The program remembers the last folder you used, so once you have done one analysis, subsequent analyses in the program run will default to the folders you used in the last analysis.

The *List the fonts* button will be enabled once the program has an input and output file. This will list the fonts the Word thinks it has in the document. Clicking on *Save font list* will save the list in an Excel workbook.

The *Analyse* button will be also enabled once the program has input and output files. Once you have clicked it, the *Pause* button will be enabled and the Close button disabled. Clicking *Pause* will give you the option to close whilst doing the analysis. Progress is shown at the bottom of the window together with an estimate of the time to completion calculated by multiplying the average clock ticks per character by the remaining characters. This means that the estimated time to completion may even increase if the program encounters many fonts in the middle of the document. The calculation does not take into account the writing of the Excel file at the end of the analysis. If you uncheck the Analyse by Font checkbox the program will run faster, but it will not attempt to analyse the glyphs by font.

Figure 3 Listing the fonts in a document.

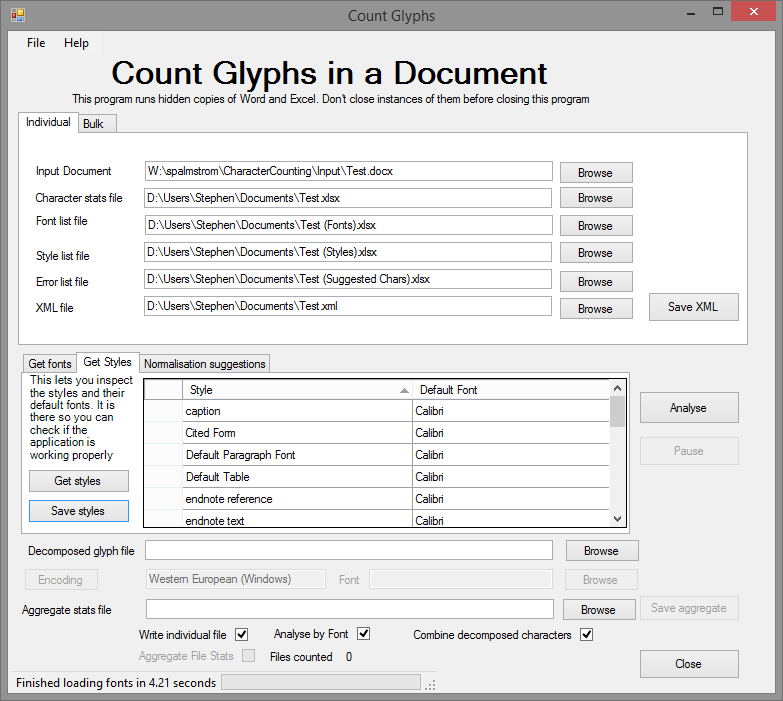


Figure 4 Getting the styles.

Clicking on the *Get Styles* tab will list the styles in use and their default fonts. It will also list some notional styles used by the program to remember the default styles in the Word document. The function is provided in case you need to troubleshoot the analysis.

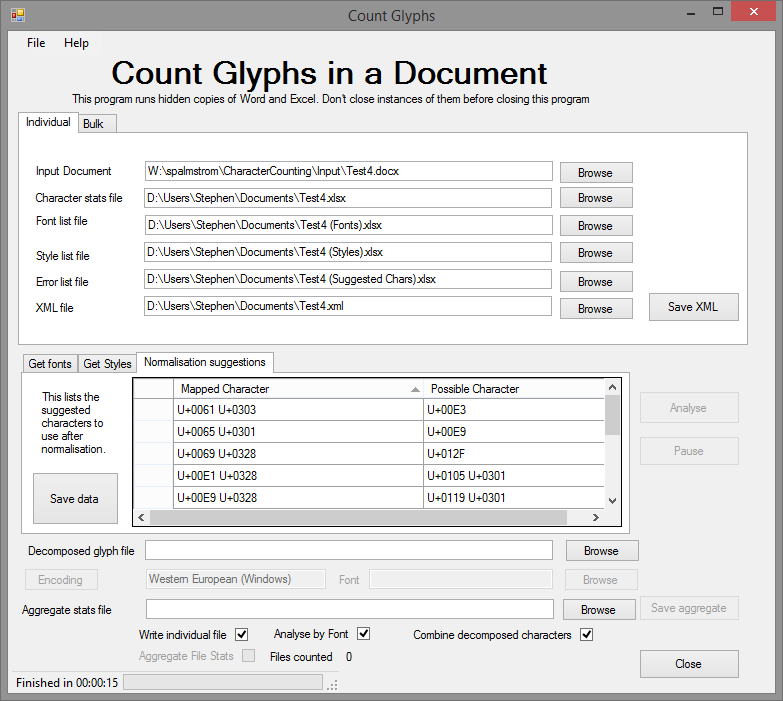
Clicking on *Analyse* will do the analysis using the options you have chosen. If the program is counting decomposed characters as one, it will also flag up any normalisation anomalies it finds. You have the option of saving them to an Excel workbook so you can modify the mapping if you so wish.

Figure 5 After analysis, showing normalisation suggestions.

# Output from one at a time analysis.

This is an example of the first few rows of the Excel statistics worksheet when Analyse by Font is checked and you analysing one file at a time:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Font** | **Dec** | **MS Hex** | **USV** | **Glyph** | **Count** |
| Calibri | 2 | 2 | U+0002 |  | 2 |
| Arial | 9 | 9 | U+0009 |  | 2 |
| Napa Heavy SF | 12 | C | U+000C |  | 1 |
| Arial | 13 | D | U+000D |  | 3 |
| Calibri | 13 | D | U+000D |  | 9 |
| Elementary Heavy SF | 13 | D | U+000D |  | 1 |
| Footlight MT Light | 13 | D | U+000D |  | 1 |
| Formal436 BT | 13 | D | U+000D |  | 2 |
| Freehand575 BT | 13 | D | U+000D |  | 1 |
| High Tower Text | 13 | D | U+000D |  | 1 |
| Horizon BT | 13 | D | U+000D |  | 1 |
| Napa Heavy SF | 13 | D | U+000D |  | 3 |
| Symbol | 13 | D | U+000D |  | 2 |
| Tennessee Heavy SF | 13 | D | U+000D |  | 4 |
| Times New Roman | 13 | D | U+000D |  | 2 |
| Troutkings BTN | 13 | D | U+000D |  | 1 |
| Napa Heavy SF | 14 | E | U+000E |  | 1 |
| Napa Heavy SF | 30 | 1E | U+001E | ‑ | 2 |
| Napa Heavy SF | 31 | 1F | U+001F | ­ | 1 |
| Arial | 32 | 20 | U+0020 |  | 19 |
| Calibri | 32 | 20 | U+0020 |  | 48 |
| Elementary Heavy SF | 32 | 20 | U+0020 |  | 9 |
| Footlight MT Light | 32 | 20 | U+0020 |  | 4 |
| Formal436 BT | 32 | 20 | U+0020 |  | 4 |
| Freehand575 BT | 32 | 20 | U+0020 |  | 11 |
| High Tower Text | 32 | 20 | U+0020 |  | 3 |
| Horizon BT | 32 | 20 | U+0020 |  | 7 |
| Napa Heavy SF | 32 | 20 | U+0020 |  | 19 |
| Tennessee Heavy SF | 32 | 20 | U+0020 |  | 11 |
| Times New Roman | 32 | 20 | U+0020 |  | 20 |
| Troutkings BTN | 32 | 20 | U+0020 |  | 11 |
| Calibri | 33 | 21 | U+0021 | ! | 1 |
| Calibri | 36 | 24 | U+0024 | $ | 1 |
| Calibri | 37 | 25 | U+0025 | % | 1 |
| Calibri | 38 | 26 | U+0026 | & | 1 |
| Calibri | 40 | 28 | U+0028 | ( | 1 |
| Calibri | 41 | 29 | U+0029 | ) | 1 |
| Calibri | 42 | 2A | U+002A | \* | 1 |
| Calibri | 43 | 2B | U+002B | + | 1 |
| Napa Heavy SF | 44 | 2C | U+002C | , | 2 |
| Napa Heavy SF | 45 | 2D | U+002D | - | 2 |
| Arial | 46 | 2E | U+002E | . | 2 |
| Calibri | 46 | 2E | U+002E | . | 6 |
| Freehand575 BT | 46 | 2E | U+002E | . | 1 |
| Napa Heavy SF | 46 | 2E | U+002E | . | 1 |
| Times New Roman | 46 | 2E | U+002E | . | 1 |
| Troutkings BTN | 46 | 2E | U+002E | . | 1 |
| Footlight MT Light | 48 | 30 | U+0030 | 0 | 1 |
| Footlight MT Light | 49 | 31 | U+0031 | 1 | 1 |
| Horizon BT | 49 | 31 | U+0031 | 1 | 1 |

The first column is missing if Analyse by Font is not checked and there is only one value for each code point. Unchecking Analyse by Font speeds the process a bit. The default paragraph font is used to display the characters when you don’t check Analyse by Font. The MetaData worksheet shows the version of the program and the input file used to generate the statistics.

If you choose to count decomposed characters as one, all the codes will appear in the one line, e.g.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Times New Roman | 100 821 | 64 335 | U+0064 U+0335 | d̵ | 1 |

# Composed and decomposed characters

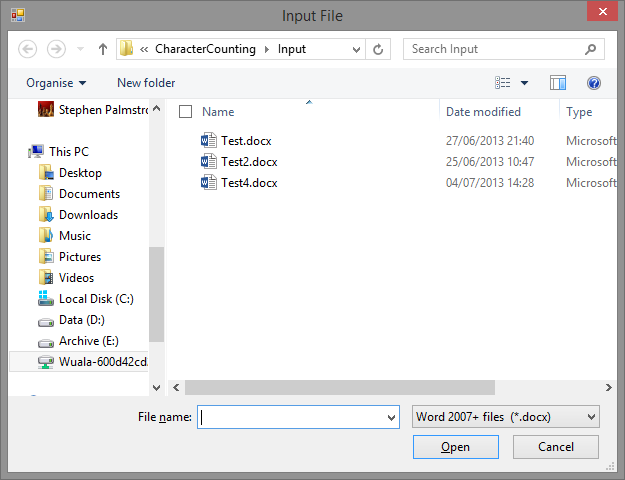
|  |
| --- |
| Glyphs |
| r̃ |
| ṣ̌ |
| š̯ |
| ʊ̀ |
| į̈́ |
| į̈ |
| ɨ́ |
| ɨ̨ |
| ɨ̨́ |

There are times when a given glyph or grapheme needs more than one Unicode or legacy character to represent it, for example to add an accent, circumflex etc. These would be defined on a font by font basis for legacy fonts, whilst Unicode has a whole list of combining characters to perform that function. Because they are standard in Unicode, standard routines can be made available to handle them, whilst they are non-standard in the legacy fonts. We handle those by allowing you to provide an Excel workbook containing the combining characters and their fonts in cells A2 downwards. The program creates a regular expression that matches any character preceding any of those glyphs, or a range of characters. A sample workbook is distributed with this program. Thus, it will find any characters associated with the suspected combining character. This may help you in mapping legacy character combinations to Unicode combined characters. This feature only makes sense if you are analysing by font. Unicode fonts are handled using standard Windows routines. We also check to see if decomposed characters can be made up of composed characters, again using standard routines and list any for which that is true. You can save the list as an Excel workbook for future reference – see the Help menu.

The program will count the combinations it recognises as single characters.

|  |  |
| --- | --- |
| **Mapped Character** | **Possible Character** |
| U+0061 U+0303 | U+00E3 |
| U+0065 U+0301 | U+00E9 |
| U+0069 U+0328 | U+012F |
| U+00E1 U+0328 | U+0105 U+0301 |

# Analysing several files at a time (bulk)

To do this, click on the Bulk tab. Browse or otherwise chose the input folder. Click the Select Files button to get a list of the files in the input folder. Select the files you want to analyse and click Open. Chose the folder where you want the character statistics workbooks to go. As with individual file analysis, you can get a list of the fonts in *all* the files in one go, and similarly the styles. Clicking Analyse will produce a workbook for each file. If you give it an aggregate statistics output file, the program will also generate the statistics in a single workbook.

# Aggregate statistics output

The Statistics worksheet has output like this, where you can see the file name is recorded, too:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Filename** | **Font** | **Dec** | **MS Hex** | **USV** | **Glyph** | **Count** |
| Test4.docx | Times New Roman | 261 | 105 | U+0105 | ą | 1 |
| Test4.docx | Times New Roman | 269 | 10D | U+010D | č | 2 |
| Test4.docx | Times New Roman | 269 803 | 10D 323 | U+010D U+0323 | č̣ | 3 |
| Test4.docx | Times New Roman | 281 | 119 | U+0119 | ę | 4 |
| Test.docx | Arial | 303 769 | 12F 301 | U+012F U+0301 | į́ | 7 |
| Test4.docx | Times New Roman | 331 | 14B | U+014B | ŋ | 3 |
| Test4.docx | Times New Roman | 353 | 161 | U+0161 | š | 4 |

# A note on regular expressions as used to analyse combining characters.

A regular expression is a very powerful, if complex, way of analysing and manipulating text. In this case, we create an expression of this form:

(.[<character combination1>|<character combination 2>|...|<Character combination n>])

This means: Match any character followed by any of the characters in the list supplied. You can include a range of characters in that list by indicating a range with a hyphen, e.g. 0-9 means match any character in the range 0 to 9. The supplied workbook contains a formula that prefixes an escape character (\) to characters that are significant for regular expressions.

# ‘Under the bonnet’

Whilst working on getting statistics on characters used in Word, we discovered that it was not possible readily to discover characters inserted using *Insert Symbol*. Instead, they were being returned as a left bracket ‘(‘ or U+0028. However, in newer versions of Word (2003 onwards) it is possible to get an XML (eXtensible Mark-up Language) representation of the document, which exposes the symbols to programs. So, we analyse the XML representation of the document to get the statistics on the text and symbols therein. It allows us to get the fonts and styles used very easily.

This does mean, however, that we do not handle characters below U+0020, i.e. control characters in the same way. This is because they are represented in a different way in XML. This program attempts to count them as if they had the value that appears in Word, so, for example, a tab is counted as U+0009 and a paragraph marker as U+000D.

We do the counting in the body of the document, which also holds headers, footers, footnotes and end notes. There are some Stories in Word that it doesn’t cover, like footnote continuations that don’t seem to be used in ‘real’ documents, though the program could be updated to do so if there were a felt need.

|  |  |  |
| --- | --- | --- |
| **Object** | **Character representation** | **Comment** |
| Footnote mark | U+0002 | ^f in Word when searching |
| Endnote mark | U+0002 | ^e in Word when searching |
| Separator | U+0003 | Seems only to be in special Stories |
| Continuation separator | U+0004 | Seems only to be in special Stories |
| Tab | U+0009 | ^t in Word when searching |
| Column break | U+000£ | ^n in Word when searching |
| No break hyphen | U+001E | ^~ in Word when searching |
| Soft or optional hyphen | U+001F | ^- in Word when searching |
|  |  |  |

**Warning** – the reliability of the counting and font specification for characters below U+0020 is less than for characters above it as we have had to deduce how Word handles these.