The AGNT Project Report—Q2 2014

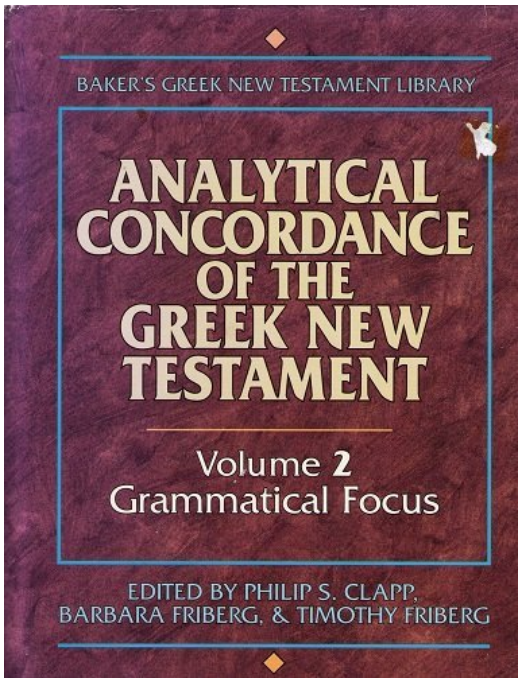
As a licensee or friend of AGNT or ANLEX, we would like to update you once a quarter about our continuing work to enhance and perfect these databases and about our plans for the future.

The Project. *The AGNT Project Report—Q3 2008* introduced the team, outlined ongoing tasks, and discussed potential tasks.

Typesetting the *Analytical Concordance of the Greek New Testament*

John J. Hughes

This article is a trip down memory lane for Timothy and Barbara Friberg and for me. It chronicles how the Fribergs' magnificent, two-volume, 4,879-page *Analytical Concordance of the Greek New Testament* (Grand Rapids; Baker, 1991; hereafter *ACOGNT*) came to be typeset.



I first became acquainted with Tim and Barbara while writing my book *Bits, Bytes, and Biblical Studies: A Resource Guide to the Use of Computers in Biblical and Classical Studies* (Grand Rapids, Zondervan, 1987), which contains a long chapter titled "Machine-Readable Texts & Text Archives." In that chapter (pp. 565–68) I discussed the Fribergs' *Analytical Greek New Testament* (Grand Rapids: Baker, 1981; *AGNT*). Since readers of this newsletter are already familiar with *AGNT*, it needs no explanation here. At the end of my discussion of *AGNT*, I outlined the Fribergs' plans for volume 1—Lexical Focus—and volume 2—Grammatical Focus—of *ACOGNT*. Little did I know at the time that Baker Book House's Allan Fisher would contact me in 1988 and that I would end up typesetting this massive work in 1990.

In addition to Tim and Barbara, many people had a hand in creating the electronic *ACOGNT* database, as well as the final, printed form of *ACOGNT*. All of them are recognized in *ACOGNT's* Acknowledgments. I mention a number of them below.

ACGONT, AGNT, and GENCORD

The electronic form of *ACOGNT* was created in 1981 by GENCORD, a concording program, at the University of Minnesota, from the AGNT database. (All subsequent references in this article to

AGNT are to the electronic database, not to the printed book.) Thus when working with the concordance files, I was working with AGNT in its concorded form. In order for *ACOGNT* to be typeset, 70-megabyte's worth of concordance files had to be transferred from tape to an enormous number of 5.25-inch, MS-DOS-formatted diskettes. This resulted in 336 files that contained 310,000 lines of text that I had to transfer from the diskettes to my computer's hard drive.

ACOGNT's printed form is based on the structure of its two sets of files: lexical and grammatical. In the lexical files, each item was listed in context according to its lemma and followed by its AGNT tag. The result was a key-word-in-context (KWIC) concordance. All instances of a word were grouped together, not listed in canonical order, and arranged by conjugation or inflection. In the grammatical files, each grammatical form was listed in context according to its AGNT tag. All instances of each form were grouped together, not listed in their canonical order, and arranged by conjugation or inflection. Thus, for example, every instance of verbs that are aorist active indicative third person singular were grouped together, regardless of the form of their lexical manifestation in the text. In both the lexical and the grammatical files, each line ended with the biblical reference, e.g., Luke 15.12, which is represented in the file as LK15.12.

When typeset, each line in *ACOGNT's* Lexical Focus volume was structured like this:

κρείττονός A-MGF-S ἐστιν VIPA--ZS **διαθήκης** N-GF-S μεσίτης N-NM-S, ἥτις APRNF-S HE08.06

I have bolded **διαθήκης** to indicate that this is the key form that is being concorded. In the printed concordance, instead of bold, a wide space appeared before the key form to make it easy to scan down a column of text. The example above, Hebrews 8.6, occurs in the midst of the sixteen occurrences of **διαθήκης** in the New Testament.

When typeset, each line in *ACOGNT's* Grammatical Focus volume was structured like this:

κρείττονός A-MGF-S ἐστιν VIPA--ZS διαθήκης **N-GF-S** μεσίτης N-NM-S, ἥτις APRNF-S HE08.06

I have bolded **N-GF-S** to indicate that N-GF-S is the key form that is being concorded. In the printed concordance, instead of bold, a wide space appeared before the key form to make it easy to scan down a column of text. This example, Hebrews 8.6, occurs in the midst of the hundreds occurrences of N-GF-S forms.

The lexical files were numbered so that when typeset in order, the printed lexical concordance was organized alphabetically by lemmas (citation forms). Within a lemma category, the organization is alphabetical by form. Likewise, the grammatical files were numbered so that when typeset in order, the printed grammatical concordance was organized by the seven major analytical divisions in this order: adjectives (and adverbs), conjunctions, determiners (or definite articles), nouns (and pronouns), prepositions, particles, and verbs. The next organizational division is alphabetical by individual grammatical tags. For more information on the organizational specifics of both volumes, I direct you to the Introduction in *ACOGNT*, from which I took the information in this paragraph.

GENCORD created files in which each line represented one instance of a lexical or a grammatical form in context, as described above, ending with the biblical reference. So that the files could be typeset, and to ensure as much uniformity of context as possible, line length was limited to 200 characters, including spaces, tags, inter-tag connectors, and biblical references. This maximum line length was determined by the width of the pages on which the concordance would be printed and by

the font sizes to be used. Since *ACOGNT* is a KWIC-type concordance, no line could wrap; each 200-character line in the files had to print as one line on paper.

As wonderful a job as GENCORD did in concording AGNT lexically and grammatically, it created a *huge* problem that had to be resolved at the proofreading stage. In order to restrict lines to no more than 200-characters, GENCORD simply chopped off as many characters as necessary at the beginning of lines, whether the ax fell in the middle of a word or in the middle of a tag. (The program could not chop the right-hand ends of lines, since this is where the biblical references are listed.) More about this later.

FinalWord II™

Because the concordance database—the lexical and the grammatical files—had to be manipulated and coded in certain ways (more on this below), conventional typesetting programs of the day—*PageMaker* and *Ventura Publisher*, for example—were not suitable. For example, they lacked internal programming languages for manipulating text, and their style-sheets were restricted to a fixed number of entries. Providentially, I had a good deal of experience in using a program—*FinalWord* II™ (FWII)—that was perfectly suited to *ACOGNT's* typesetting requirements. I had previously used FWII in 1987 to typeset *Bits, Bytes, and Biblical Studies* and other books, so I felt comfortable with it.

In the mid to late 1980s, FWII (later renamed *Sprint*, after the Borland Software Corporation purchased it in 1987) was one of the most powerful word processing programs available. The heart of FWII was a marriage between Carnegie Mellon University's text formatter SCRIBE and a powerful mainframe editor, originally developed at MIT and frequently used by programmers, called EMACS. Because of its open architecture, FWII could be highly customized by knowledgeable users. One could customize the editor, the formatter, and the printer drivers, as well as create character substitution tables, character translation tables, and much more. FWII provided users with an amazing degree of control over fonts, styles, and layout, thus making it well-suited for typesetting, though as a DOS program, it definitely was not WYSIWYG.

FWII included its own C-like programming language that allowed me to write little programs, which compiled into binary files, to manipulate the concordance database anyway I needed to. Without FWII's built-in programming language, I could not have typeset *ACOGNT*. Additionally, FWII shipped with a large number of printer drivers, including one for a Compugraphic® MCS™ 8400 digital typesetter, the device on which *ACOGNT* was typeset. I return to this topic below.

Scripts and Tagging

In the concordance files, all Greek was represented by uppercase ASCII letters and other ASCII characters. The files also used ASCII for tags, intertag connectors, biblical references, and number counts (the number of times a form occurs). These various non-Greek elements were hard coded like this—<DNMS>—to separate them from the surrounding Greek text, which looked like this: H)GA/PHSEN. The running text looked like this example from the beginning of John 3.16: *OU(/TWJ <AB> GA\R <CS> H)GA/PHSEN <VIAA--ZS> O(<DNMS> QEO\J <N-NM-S> TO\N <DAMS> KO/SMON <N-AM-S>. So anything between a < and a > was non-Greek. These various non-Greek elements had to be coded for typesetting. Each element—AGNT tag (including inter-tag connector), biblical reference, and number count—had to be given a distinctive tag so that I could control *how* it was typeset (font, size, etc.) and *where* on the page it was typeset (biblical

references, flushright; all number counts start at the same column position). With the exception of lemma entry words (see below), the Greek did not need to be explicitly coded, since Greek was the default script/font to be used in the concordance. However, the all-uppercase ASCII used to represent Greek had to be changed to lowercase Greek by creating a custom FWII character translation table. Instances where an initial letter should be capped, e.g., Moses, Paul, were indicated in the concordance files by an *, e.g., *PAU=LOJ, and these, too, were accounted for in the translation table.

I wrote three FWII programs that coded

- Biblical references like this: @vrb<MT05.25>
- Number counts like this: @num<3>
- And AGNT tags like this: @ii<VIAA--ZS>

Additionally, the lexical focus concordance files positioned each lemma *entry* at column 70, regardless of whether the first character was alphabetic or diacritical, followed by five spaces and then the number of occurrences of all forms found under that lemma. Each lemma entry was followed by one or more subsets of like forms that related to that lemma. In each of these, the first alphabetic character of the keyword began at column 90 (for the lexical-focus files). The grammatical focus concordance files positioned the keywords—the grammatical tags—at column 92, with a bracket in column 91, and a preceding space or connector in column 90. (The information in this paragraph is taken from a letter that Tim Friberg wrote in July 1988 and that Allan Fisher subsequently sent me.)

Lemma entries and their keyword subsets were typeset so that each entry word and each keyword began at the same column position; they were vertically aligned. Thus I had to tag the lemma entries (which began at column 70 in the files) and the lemma's keywords (which began at column 90).

I wrote two FWII programs that coded

- Keywords like this: @}W)NOMA/SQH (lexical focus) and like this @}@ii<A--GF-S> (grammatical focus)
- And lemma entries like this: @ewb<OI)KTIRMO/J, OU=, O(>

Keywords in the lexical focus files only needed the @} code because they only needed to be "pushed" to the column number where they should begin, which is what the value assigned to @} resulted in. Entry words, although Greek, needed start and stop delimiters—the < and the >—so that I could typeset them in bold. Thus the @ewb code included these definitional components: Greek font, bold, begin at a certain column.

In all, I wrote nine FWII programs, which were run in this order:

- Coding keywords.
- Coding entry words.
- Coding verse references.
- Coding subtotal numbers.

- Removing dollar signs and question marks. This clean-up routine was required because the concordance files (in keeping with their derivation from AGNT) used $ to indicate bold, e.g., Old Testament quotations. Question marks indicated right parentheses. For some reason I cannot remember, these had to be deleted.
- Removing extra spaces. This clean-up routine was required to ensure that only single spaces occurred in the typeset copy.
- Coding for @Hinge & @Nohinge. These two FWII commands caused certain text elements, e.g., subtotal numbers, to be typeset on a line of their own.
- Inserting "springs"—@>. This FWII "wide-break" command pushed the text to its right against the margins or the next tab stop. Springs were inserted at the beginning of every one of the 310,000 lines to prevent the lines from beginning flushleft, thus resulting in the ragged-left look of the printed *ACOGNT*.
- Coding TAGs with @ii<TAG>.

Here is what one of those programs looked like. I'll spare you an explanation!

```
message "\nCoding keywords." $
   While(10 csearch)
     (if(isend) break
       (-5 c)
          if!(istoken) (toeol f c)
          if(istoken) (r to iswhite) if(current=64) (toeol f c))
      else (r tosol 90 c) r to (iswhite || match "[^//&:\\\\]")
         insert "@}" toeol (f c)) $
            r toend
   while(125 csearch) (f c if((current=62) || (current=64)
         ||(current=10) || (current=32) || match "[123456789]")
           (r c del r c del)
      else (f c))
         r toend
bell message "\nKeywords coded." $
```

In FWII's Default.FWM file—a library of macros that was compiled each time FWII was started—I created one macro called DOITALL that consisted of the nine programs listed above, as well as an additional program called FILES, whose contents looked like this:

- find "d:\\friberg\\gf146" 0->fill newline toend f c insert " XXXXX" r toend doitall if (modf) (write "*") close
- find "d:\\friberg\\gf147" 0->fill newline toend f c insert " XXXXX" r toend doitall if (modf) (write "*") close
- find "d:\\friberg\\gf148" 0->fill newline toend f c insert " XXXXX" r toend doitall if (modf) (write "*") close
- Etc., etc.

In essence, FILES contained a list of the files that DOITALL was to be run on, one file at a time. I believe I would list 15 files, run DOITALL, and then list 15 more files. The "XXXX" that was inserted at the end of each file (later to be removed) signaled to me that the DOITALL macro had been run on that file. A note I located in some old papers that I had saved from 1988 said that DOITALL "will code 250K of data in under 12 minutes!" It didn't take much CPU power to

impress us back then. At that rate, it would have taken at least 56 hours to code all the concordance files, which totaled about 70 megabytes altogether. It actually took considerably longer.

To automate things even more, I modified one of FWII's menus (remember, this was a highly customizable program) by adding "DOITALL" to its list of items. By popping up the menu, I could select DOITALL, and the nine programs I wrote would be run on the files specified in the FILES section of the overall macro.

After the *ACOGNT* files had been properly tagged, in order to ensure complete uniformity of style and layout when printed, I defined all the necessary parameters in a custom MAKE file—a type of style-sheet file whose values FWII implemented globally when formatting files for printing. To use this file, I inserted this command—@make(newgreek.mak) at the beginning of each of the concordance files I typeset, along with this command— @device(sgreek02). That command called a file that specified which device to print to and that included various settings and specifications for the Greek characters to be used. Those were the only two commands I had to insert into the tagged concordance files in order to typeset them. Among other definitions and settings, the newgreek.mak file contained the font information for these entries:

- **@vrb** (biblical references): @form(vrb() = "@notct<@string(vrbTitle=text)@tab(150)@foox{@eval}@hinge>")
- **@num** (number counts): @define(num, font times, spacing .5, witheach "@Hinge@tab(90)@eval")
- **@ii** (AGNT tags): @define(ii, font times.small.italic, notct)
- **@ewb** (lemma entries): @define(ewb, font symbol.bold, ifnotfound, overstruck, spacing .5, afterexit "@nohinge") and @form(ewb() = "@string(ewbTitle=text)@tab(70) @SB{@eval}@nohinge")

The Compugraphic® MCS™ 8400 Digital Typesetter

FinalWord II™ was the product of a Cambridge, Massachusetts, company called Mark of the Unicorn (MOTU). In the late 1980s, FWII was the only word processing program that included a printer driver for a digital typesetter, the Compugraphic® MCS™ 8400. The *MCS* in "MCS 8400" stands for Modular Composition System. This was one of the first composition systems to use a 16-bit processor, the Intel 8086. The system came with its own WYSIWYG display, software, and a choice of three typesetting devices: 8600, 8400, and 8200. Normally, an MCS typesetting device (e.g., the 8400) would have been driven by its own dedicated Compugraphic terminal, not by an MS-DOS computer running FWII.

In about 1985, when I began to write *Bits, Bytes, and Biblical Studies* using FWII, I realized that FWII included a printer driver for the MCS 8400, and I also realized that I had a friend who owned one. (More about this below.) With Zondervan's (my publisher's) blessings, I decided to typeset the book myself. This sounded like it should be a fairly simple task. How different could outputting text to a digital typesetter be from outputting text to a laser printer? A lot, a whole lot, it turned out! And not just *different* but a whole lot more *difficult*.

The first, and one of the most maddening problems I faced, was to get an IBM-PC with FWII on it to "talk" to the typesetter, to communicate with it. In a normal situation, for example printing to a laser printer, you cable the printer to the PC, using a serial or a parallel cable, and off you go.

Despite trying several different cables, I was unsuccessful in getting the PC to talk to the 8400. I contacted MOTU and was able to get in touch with Bill Spitzak. Bill had studied computer science at MIT and was one of MOTU's software engineers, who, along with several other software engineers, had written FWII. I explained what I was trying to do—typeset a book using FWII, a Compugraphic 8400, and FWII's driver for the latter. I also explained my problem. Bill was intrigued and jumped in with both feet. After some trial and error, and after testing some things on the Compugraphic typesetter used by *The Tech*, MIT's newspaper, Bill figured out the problem. I ended up having to use a custom-made, null-modem, serial cable in "semi-slave" mode! The subsequently revised instructions for using FWII's 8400 driver included this statement: "On the IBM end, you must set the com chip strangely: MODE COM1:9600,e,8,1,p (8 data bits, even parity), MODE LPT1:=COM1, Select PRN: output when you run SPINST." That was somewhat strange to me in 1987; it sounds *very* strange now!

Now that we had the PC talking to the 8400, I thought all was well. Not so! The driver for the 8400 "sort of" worked, when I first tried it, but not properly. After working through a series of trial-and-error experiments with Bill at the helm, he finally figured out all the settings in the printer driver's variables that we needed to change and tweak to get the driver to work properly. We worked together so much on this that MOTU ended up putting this statement in a subsequent version of its printer driver file for the 8400: "The CompuGraphic Modular Composition System MCS8400. Thanks to John J. Hughes for making this work for real." Bill's name should have been substituted for mine; I simply implemented his suggestions until we arrived at a solution.

Just for fun, here is a portion of the resulting printer-driver definition for the Compugraphic 8400 that was included with *Sprint*, FWII's successor. *Pst* (proportional spacing table) specifies character widths; *tct* refers to the character translation table. Each font—regular, italic, bold, bold-italic—in a font family, e.g., Times, has to have its own pst table. Fonts can share character translation tables.

```
printer CompuGraphic,root
printer CompuGraphic.MCS-8400,page ^M,
;; The CompuGraphic Modular Composition System MCS8400 Phototypesetter.
;; Please read the comments in COMPUGRA.SPL for assistance, this printer will
;; not work unless you edit this file to match your font setup!
        hpi 1296,;; 1/18 point
        vpi 576,;; 1/8 point
        init %10z\200\200\1, reset \201\200\1,
        down \211%7>128|c%0g%<c, mvm 8064,
        fwd \213%1g%8>128|c%1g%c,
        back \213%1g%8>128|c%1g%c,
        ff \211\201%z\161\211\201%z,
        cr \213\200%z,so %c,
        shadow 6,size 24,scale 1
attr italic, on \217\200\50, off \217\200%z ;; 10 degree slant
font Times,
        on \231\200\14\207%8>128+c%0g%c,tct cc500,pst Times
pst Times,
t 15,   z 24,   T 36,   Z 36,   _ 54,   ; 18,   / 18,
o 27,   d 27,   O 39,   D 42,   . 18,   : 18,   ' 18,
h 27,   b 27,   H 42,   B 36,   , 18,   $ 27,   ` 18,
n 27,   s 21,   N 42,   S 30,   0 27,   ? 24,   = 48,
m 42,   y 27,   M 51,   Y 39,   1 27,   * 27,   @ 48,
l 15,   f 18,   L 36,   F 33,   2 27,   & 42,   { 27,
r 21,   x 27,   R 36,   X 39,   3 27,   ) 18,   } 27,
```

```
g 27,  a 24,  G 39,  A 39,  4 27,  ( 18,  | 18,
i 15,  w 39,  I 18,  W 51,  5 27,  ! 18,  " 27,
p 27,  j 15,  P 33,  J 21,  6 27,  - 18,  # 27,
c 24,  u 27,  C 39,  U 42,  7 27,  % 48,  SP 12,
v 27,  q 27,  V 39,  Q 39,  8 27,  ] 18,  Õ 48,
e 24,  k 27,  E 36,  K 39,  9 27,  [ 18,  + 48
tct cc500,
a 21,  n 4,   A 47,  N 30,  _ 53,  ; 66,  / 102, +,
b 16,  o 2,   B 42,  O 28,  . 54,  : 67,  ' 97,  <,
c 11,  p 10,  C 37,  P 36,  , 55,  $ 68,  ` 98,  >,
d 15,  q 25,  D 41,  Q 51,  0 56,  ? 71,  =,    \\,
e 13,  r 7,   E 39,  R 33,  1 57,  * 73,  @,    \^,
f 19,  s 17,  F 45,  S 43,  2 58,  & 84,  {,    \~,
g 8,   t 1,   G 34,  T 27,  3 59,  ) 85,  },
h 3,   u 24,  H 29,  U 50,  4 60,  ( 86,  |,
i 9,   v 12,  I 35,  V 38,  5 61,  ! 87,  ",
j 23,  w 22,  J 49,  W 48,  6 62,  - 89,  #,
k 26,  x 20,  K 52,  X 46,  7 63,  % 91,  SP,
l 6,   y 18,  L 32,  Y 44,  8 64,  ] 93,
m 5,   z 14,  M 31,  Z 40,  9 65,  [ 94
```

Had Bill not been able to solve those two crucial problems—getting a PC running FWII to talk to an MCS 8400 and getting FWII's 8400 printer driver to work properly—I could not have typeset *Bits, Bytes, and Biblical Studies* in 1987 or *ACOGNT* in 1990, several years after I first contacted Bill, who reenters the story below under "Typesetting the Concordance."

Proofreading Galleys—A Herculean Work!

Perhaps the most daunting challenge was proofreading the pre-typeset, laser-printer proofs. As I mentioned above, to limit lines to no more than 200 characters, GENCORD indiscriminately chopped off the beginning of longer lines, resulting in word and tag fragments being left there. This meant that I had to print a version of the concordance before typesetting it so that these "chopped lines" could be found, marked, and the fixed in the master files. Neva Miller oversaw this Herculean task. Not only did Neva help to proofread, she also oversaw eight other proofreaders (Dave Commons, Bill Griffin, Robert Merz, Henry Nall, Rod Smith, Frank Gordon Stockin, Kenneth Swank, and Terry Wade) who, like herself, were highly proficient in Greek.

Although *ACOGNT* is just under 5,000 pages, I recollect that we ended up dealing with something like 15,000 pages of laser-printer proofs. I believe that we divided the nine proofreaders into three groups of three. Then we had each group proofread a complete version of the proofs, thus ensuring that every one of the 5,000 pages was proofed three times.

The proofreaders were instructed to mark "chopped lines" with Magic Markers, as I remember. If a page did not have any chopped lines, the proofreader could discard that page. Only pages that had been marked were to be mailed back to me. Once I received the pages, my dear wife, Claire, made the required changes to the master files.

Without the proofreading work that Neva oversaw, *ACOGNT* could not have been printed, since it would have contained a considerable number of "chopped-line" errors. Once Neva's work and Claire's work had been completed, I was almost, but not quite, in a position to begin typesetting *ACOGNT*.

Typesetting the Concordance

The Compugraphic® MCS™ 8400 typesetter that I used to typeset *ACOGNT* belonged to two friends—Evelyn and Keith Barnes—who used it in their local printing business. They kindly allowed me to use the machine at night for a reasonable fee. Although I had been using their typesetter since 1987, when I tried to typeset some sample concordance files on it, using a PostScript Greek font, I faced a big problem: the 8400 had no idea what the proper character widths were for the font because it was not a Compugraphic font. Heretofore, I had used fonts purchased from Compugraphic, so I had not faced this particular problem.

If my memory is correct, at this point Bill Spitzak was in Los Angeles. Somehow—probably from some kind person at MOTU—I obtained his phone number and called for help! Once again, he graciously offered it.

The Compugraphic fonts I had been using included their own character-width tables that were uploaded to the typesetter each time I used it. The character-width definitions in the non-Compugraphic PostScript Greek font—essentially, Adobe's Symbol font, with a few custom tweaks—that I was trying to use differed significantly from those that FWII assigned to those characters, those that the typesetter was trying to use. The solution to the problem was to get the 8400 to "tell us," using its own units of measure, what the character widths were for each character in the Greek font. Under Bill's guidance, I was able to use FWII's PostScript "width-table dumper" (PST.PSC) to work properly with the 8400. PST.PSC was a utility that dumped character-width values for a specified PostScript font from a printer where the font is located to a file from which the width values could then be incorporated into an FWII PST table (proportional spacing table/character-width table). Once I had the character-width values that the 8400 assigned to the characters in the Greek font, with Bill's help I was able to construct the required PST table. There were a few other font-related problems that had to be addressed in order to begin typesetting *ACGONT* on the 8400, but, as I remember, they were relatively minor.

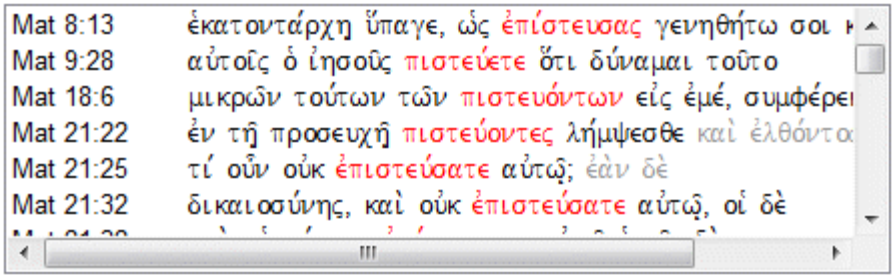
Each time I wanted to use the 8400, I had to upload the Greek PostScript font and all the other fonts used to typeset *ACOGNT* from a disk that I will call the "working font disk" (WFD). The WFD (which I believe was an 8-inch floppy!) contained the width value of each character in each font that was to be used. To upload the fonts on the WFD, I had to start the 8400, place the WFD disk in its drive, close the drive door, and press "reset" and then "load" on the typesetter's control panel, making sure that the control panel read 00.08 before and after the operation was performed. Failure to follow that procedure meant that the text would print randomly all over the page, because the typesetter would use random numbers for the character widths.

The Compugraphic® MCS™ 8400 digital typesetter was a *phototypesetter*; it output its results to film. In this context, *film* refers to a roll of photosensitive paper. When typesetting, the 8400 would pull the film from its original light-proof canister to a second, light-proof canister. After typesetting several dozen pages, I would cut the film and insert the canister with the exposed film into the developer, a machine that would pull the exposed film through two or three baths of chemicals to develop it. After a few minutes, the developed film would come scrolling out of the developer, a built-in fan gently blowing warm air on it. As soon as the film was completely dry, and after making sure that my hands were clean and dry, I used a paper cutter to cut the scroll into separate pages. Technology has progressed a long way since those days. Today, typesetting programs like Adobe's *InDesign* create PDF files, from which books are printed directly.

I do not remember how fast the 8400 was, but it was considered fast for its day. I believe it could set between 150 and 250 lines per minute, but I suspect that the very small font sizes we used in *ACOGNT* and its long, dense lines resulted in slower per-line speeds. Wikipedia says that "the 8400 was able to set type in point sizes between 5 and 120 point in 1/2-point increments. It was extremely fast and was one of the first output systems (the other was also a Compugraphic machine, the 8600) that was able to create camera-ready output with a maximum width of 12 inches" (https://en.wikipedia.org/wiki/Phototypesetting).

ACOGNT and Current Bible Software

Modern Bible software, such as *BibleWorks*, has, by and large, made printed concordances unnecessary but not undesirable. Software like *BibleWorks* that includes the Fribergs' AGNT database allows users to perform lexical and grammatical searches and to display the results in KWIC fashion like this (although the concorded forms in this example are not vertically aligned):



It is encouraging to know that AGNT, long the standard in its field, has been adopted by almost all the major Bible software companies, and that many of them also use the database that underlies the Fribergs' *Analytical Lexicon of the Greek New Testament* (Trafford Publishing, 2006)—*ANLEX*. These electronic implementations of *AGNT* and *ANLEX* have greatly expanded the number of people who can benefit from the Fribergs' work.

As always, we remain open to developing AGNT and ANLEX in ways that are most useful to the needs of students and readers of God's Word.

Thank you for your continued support of *The AGNT Project*, for faithfully marketing the AGNT and ANLEX databases, and for making these state-of-the-art tools for studying the Greek New Testament available to students, scholars, pastors, translators, and laymen worldwide.

John Hughes
Agent for *The AGNT Project*
johnhughes@centurytel.net
Phone: 406.862.7289
FAX: 406.862.0917