

Diglot Typesetting

by Charles Lehardy
The Summer Institute of Linguistics

This article describes T_EX macros that print Scripture in parallel-column diglot form. The program has been used to typeset the New Testament in the San Bartolomé Zoogocho dialect of Zapotec, a language spoken in southern Mexico.

Diglot means “two languages”. A diglot book contains a running translation of the primary text in some secondary language. In the context of our work with the indigenous peoples of Mexico, the primary text (the “idiom”) is a language such as Zapotec, and the secondary text (the “diglot”) is Spanish. Printing the two languages side-by-side serves a number of purposes. It gives them equal status, sometimes helping to settle questions about the “legitimacy” of one language or the other. It helps speakers of one of the languages make comparisons to the other. And it has the practical effect of producing shorter lines, which is important to those who are not skilled readers.

Traditionally, our parallel-column diglot versions of the New Testament were done largely by hand. The idiom and the Spanish were typeset in two separate runs. The pages were created by cutting the appropriate strips of the idiom and diglot copy and pasting them side by side. As might be expected, the process was slow and quite expensive.

Because of a growing interest in diglot New Testaments, we have created T_EX macros that page the two languages together automatically. What follows is a simplified description of how the **diglot.tex** program does its work.

The basic feature of T_EX being exploited by **diglot.tex** is its ability to read from and/or write to auxiliary files while typesetting. The Zoogocho Zapotec book of Matthew is contained in a single 325k-byte file. At the beginning of chapter one, the file contains the command `\df{svpmat1}`. This tells T_EX that the diglot material to be used with this portion of the idiom is found under the filename **svpmat1.tex**.

The `\df{diglot-filename}` command opens the auxiliary input channel `\diglotfile` and searches for **svpmat1.tex**. The Spanish Versión Popular edition of Matthew is a 200k-byte file. We’ve found that smaller files are easier to handle, so we use chapter-sized files for the diglot text, which for Matthew average a comfortable 7k-bytes each. A `\df` command placed in the idiom text at the beginning of each chapter calls up the appropriate diglot file as typesetting progresses. After `\diglotfile` is opened, verse processing begins.

Each time a new idiom verse is encountered, several things happen. First, the verse number is saved. Verses are usually marked by a single number, but sometimes the translator elects to create a synthesis of two or more verses, in which case the verses are marked by two numbers separated by a comma or hyphen (*e.g.*, 12–14). The verse command has the form `\iv m.n`, where *m* is the starting verse number and *n* is the (optional) ending verse number. *m* is compared to the current verse number in an effort to guard against missing or duplicated verses. If all is well, the `\getdiglot` routine is called.

The listing below is a simplified version of the `\getdiglot` routine. T_EX treats each verse in `\diglotfile` as a unit. `\getdiglot` pulls a verse from `\diglotfile`, increases the value of `\diglotverse` by one, makes an insert of the diglot verse, and compares the verse number of the idiom text with the diglot. If the idiom verse was a synthesis of several verses, the routine repeats until the verse counts for both the idiom and the diglot match. When `\getdiglot` has finished inserting the diglot material, the verse routine contributes the idiom text to the current page.

Pages are constructed using the routines called “marginal hacks” in the *T_EXbook* (pp. 415–416). The `\pagecontents` macro has been altered as shown below to place the diglot material

```

% Get a verse from the diglotfile and make an insertion of it
\def\getdiglot{\loop\read\diglotfile to \diglottext
  \ifeof\diglotfile\closein\diglotfile
  \else
    \advance\diglotverse by 1
    \insert\diglotins{\diglottext}
  \fi
  \ifnum\diglotverse < \lastverse
  \repeat}

```

on the page as an insertion, much like a footnote. T_EX really sees the page as a fairly narrow column of text with an extraordinarily large right margin into which we are dumping the contents of `\diglotins`.

```

% Redefine pagecontents to allow for diglotins
\def\pagecontents{
  \hrule width\pagewidth
  \global\bardepth = \ht\diglotins
  \rlap{\kern\hsize\tbar
    \vbox to 0pt{\vglue 6pt\unvbox\diglotins \vss}}
  \ifvoid\topins\else\unvbox\topins\fi
  \global\pagedim = \dp255 \unvbox255
  ...}

```

The new `\pagecontents` macro draws a horizontal line just below the header. It then stores the height of `\diglotins` which is used to draw the vertical line between the columns. Next, it kerns to the right `\hsize` (the width of the idiom text column) and prints the vertical line (using the `\tbar` macro) followed by the diglot text. After the diglot text is printed, `\pagecontents` continues as it would under normal conditions.

Because of the narrow columns used, line-breaks can become difficult. We help the program along by allowing it to stretch the white space on a line a bit more than usual. We often avoid hyphenation and justification because whole words and ragged margins are a help to beginning readers. The Zoogocho Zapotec New Testament is an example of this style. Some languages, however, have much longer words and are impossible to typeset without hyphenation. Had hyphenation been required, it would have been necessary to hyphenate the idiom and diglot texts simultaneously using two sets of rules. Unfortunately, T_EX was not designed to hyphenate two different languages at the same time. T_EX's hyphenation rules (for English) could be changed to work correctly for Zapotec, but then the Spanish text would also be hyphenated using Zapotec rules.

The easiest solution seems to be to combine the use of discretionary hyphens with penalties that make hyphenation somewhat limited. Before typesetting, we run both texts through programs that insert discretionary hyphens in each word according to the rules of that particular language. In any word containing a discretionary hyphen, T_EX will suspend its hyphenation rules and break the word where the user has indicated. This works, but we are looking for other solutions.

The sample shown here is a page from the Zoogocho Zapotec Matthew. Each column has attributes that can be altered independently of the other. These include: column width, typesize, typestyle, leading, hyphenation (on or off) and justification (on or off).

yodao' čhegaquen'. Na' leca besyebande' len xtižen' na' gose' lježe':

—¿Ga jasede' yeļa' sina'? na' čnacxečen' chac chone' yeļa' guac ca'? **55** Nombia'chone', le naque' xliņ ben' chonšagüe' yag. Na' xne'ena' lie' María, na' Jacobo, José, Simón na' Judas zjanaque' bene' biše'. **56** Na' lecze bene' zane' ca' nite' lažcho nga. ¿Nacxe chaquen' cho'e diža' ca' na' chacte' chone' yeļa' guac?

57 Na' dan' gosone' xbab nac Jesús en' con to bene' gualaž čhegaque', bi gosaclače' yesejle'e če'. Na' gož Jesús en' legaque':

—Yogo'ze bene' chonļa'ane' bene' choe' xtiža' Diosen', perw bene' gualaž če' na' bene' lo' yo'lo če' bi chesonļa'ane' le'.

58 Na' to čopze yeļa' guac ben Jesús en' Nazareten' dan' bitw gosejle'e če'.

Quingan' goc gosote' Juan ben' bzoa bene' nis

(Mr. 6.14-29; Lc. 9.7-9)

14 Na' ca tiempen' Herodes ben' naque' gobernador če distritw Galilean', bende' diža' ca nac yogo'lo' dan' chon Jesús en'. **2** Nach gože' xmose' ca':

—Bengan' da' Juanna', ben' bzoa bene' nis. Ba bebane' ladjo bene' guat ca', da'nana' chaque' chone' yeļa' guac.

3-4 Herodesen' gwne' ca' dan' bene' mandadw gosote' da' Juanna'. Quingan' goc: Herodesen' beque'e Herodías ca xo'ole' la'czla' Herodías naque' xo'ole bene' biše' Felipe. Na' Juanna' gože' Herodesen':

—Bi cheyala' soalen no'ol če bene' bišon'.

55 Pues este es el hijo del carpintero, y su madre es María. Es hermano de Jacobo, José, Simón y Judas,

56 y sus hermanas también viven aquí entre nosotros. ¿De dónde, pues, sabe todo esto?

57 Por eso no quisieron hacerle caso. Pero Jesús les dijo:

—Todos aprecian a un profeta, menos los de su propia tierra y los de su casa.

58 Y no hizo muchos milagros allí, porque ellos no creían en él.

La muerte de Juan el Bautista

(Mr. 6.14-29; Lc. 9.7-9)

14 1 En ese tiempo Herodes, el que gobernaba en Galilea, tuvo noticias de Jesús,

2 y dijo a los que estaban con él:

—Ese es Juan el Bautista, que ha sido resucitado de la muerte. Por eso tiene este poder milagroso.

3 Es que Herodes había hecho detener a Juan y llevarlo atado a la cárcel. Lo hizo por causa de Herodías, que era esposa de su hermano Felipe.

4 Juan había dicho a Herodes:

—No debes tenerla como tu mujer.