Diglot setup instructions.

(this assumes you know how to make the normal paratext2 side of things work)

Make a diglot - friendly .usfm file (or files)

this can be by hand or by using the `diglot_merge.pl` perl script.
Useful commands: `\lefttext` `\righttext` `\nolefttext` `\norighttext` This snippet (from `history.usfm` in the examples/diglot directory) shows some of them in action.

```
\lefttext
\s The birth of diglot
\p
\norighttext
\p
\v 4 David took ptxplus and tweaked it in various places and so diglot came into
   being, and it lived in dark obscurity for many years, sometimes crafting text,
   sometimes creating strange things.
\p
\righttext
\p
\v 4 And David took ptxplus and did bend it in diverse places to make diglot.
   Diglot dwelt in darkness and created beauty or disaster depending upon many
   factors.
```

Notice that the `\norighttext` means that the `\v 4`s should line up, with a space in the right for the section heading.

1 [1]This is the begining of the story of diglot. [2]First Knuth made TEX, then Kew made XETEX from TEX. [3]XETEX had a child called paratext2, and paratext2 had a child called ptxplus.

The birth of diglot

[4]David took ptxplus and tweaked it in various places and so diglot came into being, and it lived in dark obscurity for many years, sometimes crafting text, sometimes creating strange things.

[5]Then Mark and Martin created of a beautiful thing called ptxprint using Something Completely Different called python, and asked David to possibly make the diglot code fit into it. And David had time, and patched some bugs in diglot and also in ptxprint. And so it was that diglot ventured out into the world.

1 [1]Hereby beginneth the account of the nation of diglot. [2]Knuth begat TEX, Kew took TEX unto himself and begat XETEX. [3]XETEX begat paratext2, paratext2 begat ptxplus.

[4]And David took ptxplus and did bend it in diverse places to make diglot. Diglot dwelt in darkness and created beauty or disaster depending upon many factors.

[5]And then did Mark and Martin come together and used Something Completely Different, named python, and crafted the wondrous ptxprint. And David was asked if diglot would wed ptxprint. And David found the precious treasure called time, and trimmed some strangenesses from diglot, and even from ptxprint also. And behold, the two became one and marched forth into the world.

Setting this up by hand is possible, but rather hard. The easier option is to use the `diglot_merge.pl` program. You might want something like:

```
diglot_merge.pl -s -p -L merge.log -o merged.usfm file1.usfm file2.usfm
```

Those options say:

- -s Spit sections into their own sections
- -p Use paragraph-by-paragraph matching
- -L Logging goes to merge.log
- -o Output goes to merged.usfm
- The contents of file1.usfm and file2.usfm go on the left and right respectively.

You can also select a range using `-R chapter:verse-chaper:verse` It might work, but it's been a while since it was extensively tested.

Set up the stylesheet options.

Any given marker `\thing` applies to both sides unless it is overridden; marker `\thingL` only applys to left column, `\thingR` only applys to the right column. Ptxplus's addition `\BaseLine` property should actually work, so you can have different baselines if you need them for your fonts, as well as different faces.

Make sure that you set `\diglottrue`
If you plan to use the `\thingL` format, you must load the style sheets with `\diglottrue` set, even if you then use monoglot text first.

- `\def\DiglotLeftFraction{0.45}` % In case texts align better with unequal columns
- `\def\DiglotRightFraction{0.55}`
  You probably want to ensure that the fractions add up to 1! These fractions will probably need tweaking, based on the relative wordiness of the two translations, the font, and so on.

I'm not sure what else is needed, other than running xetex on your master file. There are more details in the documentation directory.