PLX 1.3

PLX Programming Notes:
I've received several inquiries about the methods in PLX, so I thought a few words on the methods might be helpful.

Fonts Unit
The FU gathers together the methods and variables used to enumerate fonts in PLX, and tries to put an object interface on them. To use it, you would create an instance of PFont, get a DC for your output device (screen or printer), and call TFont.Enumerate. Your pFont object now contains a collection of TFontItem records available via the TFont.At method (as well as the horizontal & vertical resolution in LogPixX and LogPixY). Each TFontItem record contains a

TLogFont structure,
FontType variable, and
Sizes collection of available sizes (in device units)

for each typeface reported by the API function EnumFonts(). In general you would select the type face you wish to use, manipulate the TLogFont fields as needed, realize the font with a CreateFontIndirect call, and call GetTextMetrics to check on what the Windows font manager has **really** done.

The Sizes collection contains the available sizes only for raster fonts - vector fonts will have only one entry. You can use the FontType field to determine which beast you have at hand. This is not some bizarre whim of mine - it's the way EnumFonts() sees fit to answer.

Please note that TFont does not create fonts - it merely enumerates and stores information about available fonts in a given DC. You have to actually realize the font in your program.

PODButton - Owner Draw Button
This ownerdraw button object is defined in the WOPlus unit. It handles most of the overhead of drawing and maintaining itself. To use it, create an instance in your host window - the init syntax requires only one addditional piece of info over a regular button: the name of the bitmap resource to display on its face. Then create a response method for the wm_DrawItem message (that Windows sends to the parent of every owner-draw item) that decodes the object type and instance, and calls the object's DrawItem method. Otherwise, the button behaves mostly like a regular TButton.

TSText - Sculpted Text field
Behaves more or less like a regular static text field - except that it is set in front of a borderless window with hilites to simulate a raised or recessed area. Its Init method requires a st_Recessed or st_Raised flag, and a style flag composed of regular Windows dt_xxxxxx style constants. You can have left, right or centered text among others.

Printer1 Unit
Contains various subclasses of Bob Galivan's TPrinter unit(along with some new methods) to provide support for margins, changing fonts,

printing a header and/or footer, getting a quick DC, and changing printers. The printing model is fairly simple:the page has margins, responds to various fonts (but only one per line, pls!), a header and/or footer band, and a body of text. Your program feeds lines of text to the printer object, who has the responsibility of keeping track of things and deciding when to print the footer, eject the page, reinitialize the current font, and print the header. Basically, you override the DoHeader and DoFooter formal methods in your own code if you wish to have a footer or header. The SetFont method lets you change fonts, and returns the previous font so that you can delete it. The footer provision is the most problematic: in order to provide space for the footer text, TPrinter1 executes the DoFooter routine at the top of each page in a trial mode. Your DoFooter routine must be able to respond to this request by printing as many lines as it will actually use later on (and not play tricks with the print position). You are free to switch fonts in the DoFooter method, as the calculation routine saves the printer state before making the test and then restores it. Not ideal, and you can ignore the DoFooter altogether if you like.

The ChgPrinter method puts up a list box with the defined output devices, and lets you make one of these the current output device. You can't set the connections, or 'activate' an inactive device - you need Control panel for this. But, if you have two or more active devices, you can quickly switch between them.

Good Luck

Doug Overmyer
10/8/91