XEROX **Internal Memo**

To BCPL Programmers

From Richard Sauvain
CSIA
WRC/ISL
W128 - 8*222*5298

Subject XM Display Stream Package

Date February 22, 1980

Filed on: [ERIE]<AltoDocs>XMDStream.press

This memo describes a BCPL-callable XM display stream package for the Alto. The package allows you to keep display bitmaps in the upper memory banks of an XM Alto, thus making more space in bank zero available for programs, virtual memory buffers, and the like. It is designed to be as compatible as possible with the standard OS display stream package (see the *Alto Operating System Reference Manual*, pp. 57-59). Notable differences are: a different stream creation call, use of a strike font instead of a .al font, and a few un-implemented features.

Where to get it : BR files for the package and several utilities and accoutrements can be found in dump file form in [ERIE]<Alto>XMDStream.dm. Load this file with FTP to get the package onto your disk. Documentation (this memo) is in [ERIE]<AltoDocs>XMDStream.press, while source files are collected in [ERIE]<Alto>XMDStreamSources.dm.

Differences from the OS display stream package : most of the procedure names are the same as those in the OS version. Hence when you load the XMDS package, references to the OS display package automatically get satisfied by XMDS routines - which work just like the OS versions except:

1. There is no *CreateDisplayStream*(). Instead, you call:

sPtr **= XCreateDisplayStream**(dsPlace,BitmapPlace,wWidth,BitmapSize,FontPlace,Bank)

dsPlace is the address of a 60 word block of memory supplied by the caller in bank zero, to hold the displaystream structure used by this package.

BitMapPlace is the word address (must be even) of the first word of the memory area to be used for the display bitmap

wWidth is the display line width, in Alto screen units divided by 16 (wWidth of 38 is full width).

BitmapSize is the size in words of the first word of the memory area reserved for the display bitmap.

FontPlace is the word address (must be even) of the first word of the area into which a strike font has been read. This must be in the same bank as the display bitmap. See the description of XReadFile below for a way to read the font in.

Bank is the memory bank in which the bitmap and font reside. Banks are numbered 0,1,2 & 3. Not all XM Altos have banks 2 & 3 - hence bank 1 is the usual choice.

XCreateDisplayStream returns as value a display stream pointer, which is used just like the result of *CreateDisplayStream*. I.e. you can *Gets* or *Puts* to it, use *Wss*, set its margins, etc.

2. **SetLinePos** will work to lines that have not been written on; the only reason that it can fail is if line number requested is greater than number of lines in the window.

3. **InvertLine** ignores the second argument (lpos); inverts entire window.

4. **CharWidth** can't handle a font pointer as the first argument as can the OS version. Results are unpredictable if a font pointer is in fact given.

5. **EraseBits** cannot invert (third argument a -1), will call Swat if you try.

6. **SetFont** and **Scroll** are dummys which call Swat; you cannot change the font after the stream has been created.

7. **GetFont** and **GetHeight** are not present in this package. If you reference them, you will get the OS display stream versions!

Operating Environment

In order to use this package, you must do some setup work that is not required when using the OS display stream package. I suggest you look at the beginning of XMDSDriver.bcpl for examples while reading the following.

1. The IFS microcode must be loaded (in order to get BitBlt microcode available). This requires that you do a *LoadRam(IfsRamImage*) before calling any of the XMDS routines.

2. It's a good idea to call *CountBanks()* to verify that the Alto you're running on has the bank you anticipate using , and to clean up parity in the XM banks. This procedure returns the number of banks available: 1 thru 4 (zero if not an XM alto).

3. You have to manually allocate your upper memory bank, i.e. assign non-overlapping areas for a font, all your display bitmaps, and anything else you need to put there.

4. A strike font must be read into the memory bank you are using for bitmap storage. Strike fonts may be found by sniffing around your local file server looking for files whose names end in ".strike". A generally useful one, StSerif.strike, is included in XMDStream.dm. A utility to read such a file into an XM bank is available in XReadFile.br. Call *XReadFile(filename,bank,wordaddress*) to read in the strike font.

5. The bank register for the display word task must be set to indicate the bank you're using. This register is at location 177751b. Set the 'normal' bank , not the alternate (see p. 9 of the Alto Hardware Manual for more details on this). Before finishing, set it back to zero.

6. If you expect to use Swat to debug, it's a good idea to attach a Swat context proc that will switch the display word task bank register back to bank 0 when you enter Swat and back to whatever you're using when you exit. If you don't, the display looks mighty funny whenever you Swat. Such a proc is available in XMDSUtilsA, and is called *SwatProc*. You set *@lvSwatContextProc* to the name of your procedure.

Loading

Your load command should include the modules XMDStream, and XMDSUtilsA (and if you use *CountBanks*, XMUtils). All of these modules are included in XMDStream.dm. You will also need IFSmc and LoadRam from the <Alto> directory.

Disclaimer:

This is an augmentation of a package originally written by Peter Deutsch for DLisp. I've been using it in one fairly complex application for some months with no problems, and am willing to answer questions. I cannot guarantee that it's well checked out or that maintenance support will be available.