Usage notes for CLIPMETA

Hippocrates Sendoukas
University of Southern California
Dept. of Finance & Bus. Economics
March 15, 1993

Introduction

clipmeta.exe is a small utility that takes a metafile from the system clipboard and saves it to a disk file. This is helpful because most programs can send a metafile to the clipboard (using the "Cut" or "Copy" commands from the "Edit" submenu), but do not save it to a disk file. The main motivation for writing this program was to save pictures produced by any Windows program, so they can be used with my DVI driver. I tested it with several popular programs (Excel, PowerPoint, Word for Windows, 1-2-3 for Windows, Toolbook and Paintbrush) and it works fine.

Operation

You can invoke the program as:

clipmeta [-q] [-p] [filename]

where the items is square brackets are optional. If you do not supply any parameters, the program displays some information about the "natural" size of the metafile in the clipboard, and then displays a "Save As" dialog box that lets you select the directory and filename to save the metafile. If you specify the "-q" (quiet) switch, the program will not display the initial information about the metafile. If you specify the "-p" (plain) switch, the produced metafile will be in plain format instead of placeable format. If you supply a filename, the program will skip the "Save As" dialog box and will use the specified filename as the destination for the metafile.

The optional command line parameters are not anachronisms even for graphical user interfaces: suppose for example that you have a chart in Microsoft Excel and you want to save it to a metafile. Suppose that the name of the chart is "graph.xlc". The standard steps to produce the metafile are:

1. Select the entire chart by clicking the mouse.
2. Select the "Copy" command from the "Edit" submenu.
3. Minimize Excel so you can see the Program Manager.
4. Run the "clipmeta" utility.
5. Press the OK button when you see the size of the metafile.
6. Select a directory and enter the name "graph.wmf" in the "Save As" dialog box.
7. Restore Excel to continue.

These steps are not that difficult, but you do not want to do them frequently. After all, the whole point of a graphical user interface is to make your life easier. For this reason, you can write an Excel macro that carries all these steps automatically. I use the following macro:

```
A1      Export_Chart (a)
A2      =get.window(1)
A3      =if(find(".XLC",A2)>0,goto(A5))
A4      =goto(A9)
A5      =window.restore()
A6      =select("Chart")
A7      =copy()
A8      =exec("clipmeta -p -q"&substitute(A2,".XLC",".WMF"),1)
```

```
A9      =return()
```

The first line (A1) is the name of the macro and specifies that it can be invoked with the Control-A key. The second line (A2) gets the filename of the current window (it should be "GRAPH.XLC"). The next two lines (A3 and A4) make sure that the name has the extension "XLC". This check is useful, if you try to run the macro either from a non-chart window, or if the chart has not been already named. The next line (A5) ensures that the chart window is not maximized or minimized (you will get better results if it is at its natural size). The next two lines (A6 and A7) select the chart and copy it to the clipboard. The following line (A8) substitutes the extension "WMF" instead of "XLC" (the filename will become "GRAPH.WMF") and executes the command "clipmeta -p -q GRAPH.WMF". The final line (A9) terminates the macro.

If you run the macro from the chart window, all the above steps will be done automatically. You can put the macro in the global macro sheet of Excel, so it is always available. Furthermore, you can put a new button in the graph toolbar, and assign the macro to that button; in this way, you can do the whole operation just by clicking on that button.

You can tailor the macro to your own preferences, and you can do similar things on any application that supports macros. My main objective was to maximize the user's convenience. You can also put the program in any of the Program Manager groups for quick access when you cannot automate the entire process.

Caveats

The standard extension for windows metafiles is "wmf"; you should not use any other extension for compatibility purposes. Aldus developed an extended metafile format (called a "placeable" metafile), which circumvents a limitation of standard metafiles (it includes information about the natural size of the picture). Unfortunately, they chose the same extension (wmf) as the one for standard metafiles, and this can be the source of some confusion. Most applications expect a file with a "wmf" extension to be in the placeable format, while other applications (eg. Ventura) expect the plain format. If the metafile contains text, you will get much better results by using the scalable fonts introduced in Windows 3.1. Finally, keep in mind that unless you have the placeable metafile filters (distributed with MS-Word, PageMaker, Toolbook, etc.), you should use the standard (plain) metafile format; the DVI driver can read this format without any external support. Some programs do not specify the size of the metafile when they pass it to the clipboard. In that case, clipmeta cannot use the placeable format; it will save the metafile in the standard (plain) format. Metafiles produced by Simulink (Matlab for Windows) cannot be properly scaled, because it uses certain instructions reserved for the program inporting the metafile. Apart from this, its metafiles have a black background with white drawings which renders them more or less useless for printing. Mathematica metafiles can be very demanding on Windows; in some cases Windows runs out of resources (without any indication) and parts of the graphic may be wrong or missing. The only workaround that I know is to simplify the graphic: even if you add memory in your system, the amount of resources stays the same because of Windows limitations.