


Designing a GUI Look

By *Sam Vulopas*

In determining a GUI look and feel for your applications, there's a lot to consider. How much or how little do you want it to look like Microsoft Windows? Like it or not, it's the standard. Here, Sam Vulopas, president of VLS Consulting, outlines the basic considerations and gives some design advice.

WINDOWS STANDARDS

Windows standards are anything but...

 Many current publications exist (Microsoft Press, Que, etc.) that define standards for GUI applications. You can derive most standards using Microsoft products as an example. Other companies have created their own specific looks that, while not completely Windows standard, have gained widespread acceptance. The common thread through all of the various looks is really the feel of a GUI application. In addition to the abilities inherent in Windows controls, this includes event-driven entry, multitasking and the ability to use a mouse.

DESIGN MEETINGS

Engineering will generally vote for a straightforward interface, high on function, low on aesthetic: the Windows equivalent of whitewashed apartment walls. Marketing will want an interface equivalent to a Monet painting: something that renders people awestruck in its presence.

Ideally, the design should fall somewhere in the middle. Is the interface boring? Is it cluttered and confusing? Keep in mind what would work best for your target market. Also, design meetings are a good place to practice your people skills and learn how to negotiate different opinions and ideas.

AESTHETICS

In general, common sense will dictate how the application should look. Keep the design current. It makes no sense to work for months or years on an application and have it look outdated when it's finally released. Here's a simple test:

Start two or three other current Windows applications on your system and do a side-by-side comparison on the interfaces. The overall looks should be similar. Try to keep them consistent without sacrificing too much individuality. If your application uses 100-by-100-pixel tool buttons and most others use 25-by-25 buttons, that should be a clue.

Generally, field literals (static text controls) should be right-justified and end with a colon (:). Entry controls should be left-justified, placed just to the right of the field literal and have the flags set to display with a three-dimensional lowered client edge.

When deciding on entry control widths, I use the following guidelines:

- Key, numeric and date entry controls are sized to the field length times the pixel width of the widest character allowed, plus any formatting characters (slashes, hyphens, etc.).
- All other entry controls are sized to the field length times the average character-pixel width.

Generally, because of proportional fonts, there will be more room available on a Window screen than a character-based screen.

Fields should be logically grouped and may be enclosed in a group box, if desired. The standard font should be MS Sans Serif, 8 point. The entire application should default to the current Window color settings.

When converting existing systems, the character-based screens will, for the most part, drive the look of the new interface as far as field placement is concerned. All fields on an 80-by-25 character-based screen will fit in the same location on an equivalent 640-by-480 pixel window, with room left over for toolbars, and menu and status bars.



Conversion Examples

Figure 1 shows the existing character-based Inventory Maintenance screen. Figure 2 shows the same maintenance function after being converted to a GUI interface. While there are differences in alignment, the entry fields are generally in the same position on the new window. Key field fonts are in boldface type.

Figure 1. Character-Based Inventory Screen. (Image used with the permission of Parasol Systems, Inc.) The toolbar contains rollover buttons for the following functions, from left to right: New, Save, Undo, First Record, Previous Record, Next Record, Last Record, Delete, File Inquiry, Field Inquiry, Entry Image (an application-specific standard), Help and a link to the company's Home Page.

The status bar segments contain microhelp, current company and other application-specified indicators.

Toolbars & Buttons

Tool buttons should exist for the most common or standard functions. Rollover buttons are becoming standard (the button seems to rise up from the window when the mouse pointer is moved on top of the button), as are movable toolbars. The "standard" tool button size is 25-by-25 pixels. You can find button images by the thousands on the Internet. A quick search of "images" with my browser returned approximately 400 possible sites, and the majority of the images were free.

Menu Bars

Menu bars, at a minimum, should perform the same functions available on tool buttons. Ideally, they should contain every possible option available for the window and should also include:

- Accelerator codes (underlined letters on window menu items) for each option.
- Hot Keys or shortcut keys (Ctrl+N = New, Ctrl+S = Save) for the most frequently used functions.

There are standard uses for many of the Hot Keys, so pay close attention to the Microsoft Windows standards.

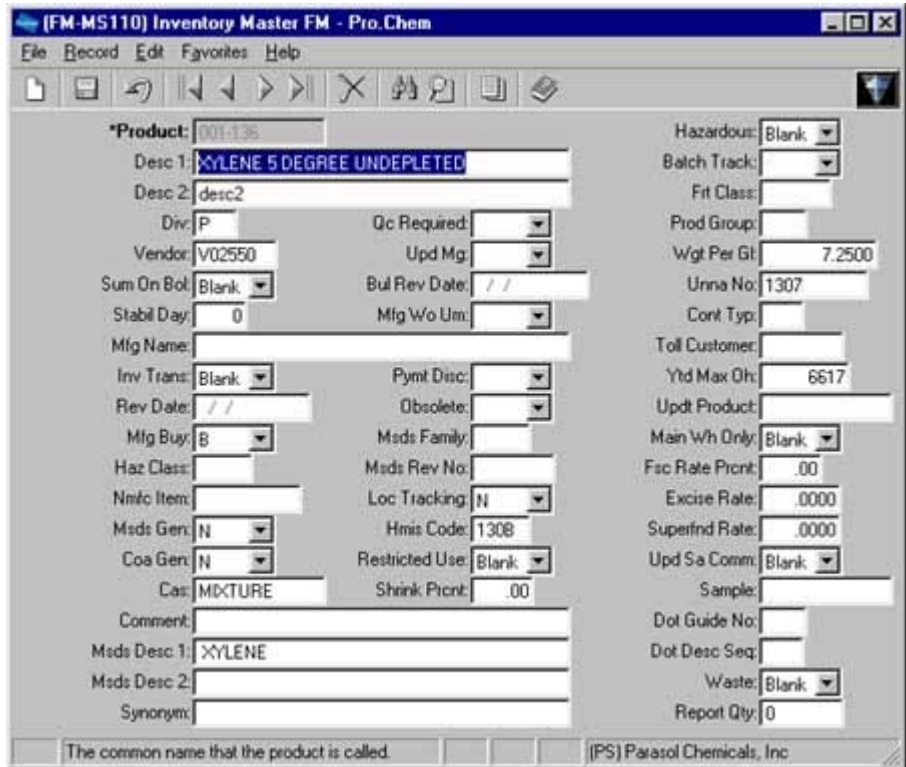


Figure 2. Inventory Screen Converted To GUI. (Image used with the permission of Parasol Systems, Inc.)

Status Bar

A wealth of information can be placed in the status bar. Present information that will be useful to the end user. At a minimum, field microhelp should be displayed. Other useful information about the current field could include field length, required or optional, inquiry available, etc.

Application Menu System

This one's a toss up. Menu system designs vary from one application to the next. Pick one that you like and go with it.

Splash Screen

The splash screen is the point of entry for your application and is basically a two- to three-second display of the company and software advertisement.

Note to Marketing: Here's your chance to pull out all the stops and create your Monet. Engineers won't get too distraught about this because there's no engineering change involved; they simply display the bitmap you give them.

USER ERGONOMICS

Consistency

Consistency is the user's best friend, promoting comfort and ease of use. The concept is simple; teach someone how to use one standard maintenance window, and they know how to use all of them. Ditto for Header/Detail, etc. Decide on the look for each of the interface types:

- Standard Entry/Maintenance

- Header/Detail Entry/Maintenance
- File Inquiries
- Batch Processing (includes reports and updates)

Inevitably, there will be a few non-standard windows created for functions with no previous Microsoft Windows example. It's important to remember the overall look must be consistent even though the various interface types will have different tool and menu options.

Keystrokes

Contrary to popular belief, a new, correctly designed, Windows-standard application will not need more keystrokes than an equivalent character-based application. A converted character-based application, with a few modifications, can be reduced to one or two more keystrokes per function.

Certain key differences are obvious. versus , for example (don't you just hate that!). Resist the urge to use the SETOPTS bit that reports the key as a . This causes engineering problems when a button with an default (an OK button) is placed on the window. What does mean? The next field? The default button?

Navigation

The tab order, i.e. field entry order, should follow the field positions on the window. The important point is the user must be able to navigate around the entire window and perform all available functions without a mouse.

These are just some ideas and guidelines to follow when creating or migrating your product from character to graphical user interfaces. If you stick with the Windows standard or close to it, you probably won't go wrong. Using existing applications and volumes of design guides as examples, your software can have a true Windows look.