Right Resources Can Open Whole New Windows

By Dan Rask

These days, learning to do quality Windows application design and user interface design is becoming more and more critical. The software development landscape is littered with bankrupt companies that didn't take Windows seriously at first, or tried to graft their own models and standards onto it. Some companies may still be able to convince customers that UNIX terminals are more productive or less expensive for order entry and other tasks. But Business Basic software developers increasingly run into customers who say they need every application to have a Windows graphical interface. In this article, we want to point to resources we've used to acquire new ideas and skills that can make a first Windows application more successful.

Expanding the library is always a great start, but training seminars, Web sites, and newsgroups are just as critical to keep track of the constantly evolving technical landscape. Web sites from companies like Microsoft and BASIS hold a treasure chest of material. Windows has been popular for seven years now and a lot of the challenges to programming for Windows may have already been solved by someone working outside the BBx® world. Even if it seems that a BBx application has nothing in common with Excel or Sidekick, look again. Features like the Windows tab control have been used in some very creative and innovative implementations by different developers.

Increase Product Life and Market

The time has passed when individual application developers could effectively define the interface "standard" for their users. For example, many Business Basic application end users and vendors believe that backward compatibility for screens and keystrokes should define the interface for an application that brings forward thousands of character-based legacy users. But to require this and ignore Windows interface standards and the requirements of new users is dangerous at best. Quality design and a good sales and marketing program to explain the benefits of a new interface may result in increased sales and a longer-lived product in the end. Consider this: a Windows application, no matter how easy the development environment, takes considerably more effort than the equivalent character-based application. Given that the investment in a new Windows version will be greater than any previous release, why not use an interface that addresses the largest possible market and affords the longest product lifecycle?

Begin With Good Application Design

Good application design is the hardest thing to pick up from standard industry sources. Many of these sources focus on languages other than Business Basic or on high-level design issues that don't translate into Business Basic code implementation. But because a longer development cycle could be at stake, thorough application design education is worth serious consideration.

We have found the following books to be excellent resources in acquiring knowledge and information about Windows applications.

One of the contemporary classics of software development is *Code Complete*, by Steve McConnell. Although the book is not specific to Windows, Windows development is inherently more complex than character-based development and, therefore, it matters that much more to have a clear, complete development process. There's enough in this book about analysis, design, architecture, implementation, testing, and maintenance to construct a life cycle that fits any project.

The Essential Guide to User Interface Design, by Wilbert Galitz, and *About Face: The Essentials of User Interface Design*, by Alan Cooper, contain important background information that will probably contribute more to core interface design than the Microsoft interface guideline. They discuss design *values* and *quality* as much as what controls work best and how they should look. They're good to read in the early stages of a new product cycle and good to keep for reference at key points in the application design, i.e., "Should this screen be ported over as is, or should we convert it to multiple child windows?" One of the most important concepts they remind us of is that programming and interface design have very little to do with each other. This is true both in terms of skills and implementation. Because most business applications will continue to be done by developers who have to design and code, it's worth planning how to successfully wear the two distinctly different hats.

The Windows Interface Guidelines for Software Design (Microsoft Press) is included with many Microsoft development products and is also available separately. It's the specific tract on current accepted interface practice: not what should be, but what is. This book should be a constant reference during interface prototyping and design.

Jim Douglas's accompanying review, "Off the Shelf," covers some of these key resources in greater detail. What's most important is that all three are every bit as important in the programmer's library as the PRO/5® User's Guide Reference or a good ODBC text.

The final word, of course, is that the latest innovations aren't always in the guidelines. When in doubt, find out how Microsoft does it. The industry is divided on whether Microsoft is the great interface innovator or simply the world's most successful appropriator. Regardless, every programmer should be aware of its newest applications because it may become next year's standard, in both the published book and the customer's mind.

Beyond the Bookshelf

Fortunately, there's more help available than just a stack of technical books. It's possible to finish school and spend twenty years in business programming without taking another class or attending conferences. But many developer and IS-focused conferences have technical breakout sessions and training seminars that can concentrate a lot of learning into a few hours. Anyone lucky enough to live within commuter airline distance from New York, Boston, San Francisco, or Las Vegas can find several trade shows and conferences that will be valuable and affordable. A variety of industry publications also sponsor technical seminars.

Visit BASIS to Net Ideas

After the software design is complete, matters turn to what programmers need during implementation: quality reference documentation. The current Visual PRO/5™ manual set has a lot of reference information about Windows mnemonics and core SYSGUI programming. The *GUI Guide* is a compendium of the information found throughout the manual set.

BASIS has always provided ideas and guidance on good application design through its magazine and technical publications. *The BASIS Advantage* has devoted hundreds of pages to Windows and Windows-related programming. We have recently begun to put past *Advantage* issues on the Web site at http://www.basis.com/advantage/. The BASIS website also has a version of what used to be known as the Online Manual Project. Now called "The Programmer's Technical Archive," this section contains white papers and sample code. See http://www.basis.com/support/pta.html.

Upcoming BASIS releases will have documentation available earlier. The *Kilauea* project, the successor to Visual PRO/5 that is currently under development, already has an alpha version of the manual supplement available in HTML format. Interested developers can contact BASIS to become part of the Early Access software program to preview both the software and documentation.