

January, 2002

Editor's View

The Needs of the One *are* the Needs of the Many

Designing applications to accommodate users with disabilities results in better design for all

One of the design goals for VFP 7 was to enable us to create applications that can be used by people with disabilities. The biggest change on that front is support for the IAccessible interface that lets accessibility tools, like screen readers, work with VFP applications.

Why did Microsoft care about this? There are two big reasons. First, there are a lot of people out there with disabilities. More than 20% of the US population has at least one disability, and the number increases with age. Second, and probably the primary motivator, is that software purchased by the US government is now required to be accessible to people with disabilities. If the US government isn't the largest purchaser of software in the world, it's close.

Windows' Accessibility Features

In fact, Microsoft has been working on many fronts to make its software and applications created with its software accessible. Each new version of Windows includes more accessibility features.

Accessibility features fall into two broad classes: settings and tools. Settings allow a user to configure the computer as he or she needs it. For example, Windows includes settings for colors and font sizes, mouse pointers, keyboard response, and much, much more. Many of the accessibility-related settings are in the Accessibility applet in the Control Panel, but others that affect accessibility are scattered throughout other applets, such as Display Properties and Mouse.

Accessibility tools go farther. They allow a user to use Windows and applications in a non-standard way. For example, the Narrator tool that comes with recent versions of Windows reads text from other applications aloud. Magnifier uses part of the screen to enlarge the area around the mouse cursor. (You may have seen a speaker use it at a user group or conference session.) The On-Screen Keyboard provides an input device for users unable to manipulate a keyboard. Accessibility tools are found in the Accessibility group of Accessories.

There's another difference between accessibility settings and accessibility tools. You're pretty much limited to the settings that are built into Windows, but many other companies provide accessibility tools. In fact, Microsoft says clearly that the tools it provides are meant more for initial set-up, emergencies and to demonstrate possibilities than for everyday use.

There's an excellent section of the Microsoft website devoted to accessibility issues: www.microsoft.com/enable includes information about Windows accessibility, Microsoft's efforts in this area, third party accessibility tools, and more.

Accessibility and Design

As many of you know, I'm particularly interested in user interface design. So, when I saw VFP 7's improvements in the area of accessibility, I started thinking about how to design applications that take advantage of the improvements, but more importantly, are designed to accommodate users with disabilities. What I learned was interesting.

It turns out that, in most cases, considering users with disabilities in designing a user interface results in a better interface for all users. The same principles that guide accessible design apply to overall user interface design: consistency, flexibility, and listening to the user.

The most important issue in designing accessible applications is to stay out of the way of accessibility tools and settings. That means going with the flow—honoring the user's color and mouse pointer choices, using system dialogs rather than your own whenever possible, using text rather than bitmaps of text, and so forth.

Consistency within an application is also a key consideration. Determine the standard way that your application will work, in terms of menu structure, tab order in forms, and so forth, and then apply it throughout. Ideally, your application should work the way other Windows applications work, so that users can transfer learned skills. While all of these issues apply no matter who the intended user is, they're especially important for users who may have visual or motor impairments that make it hard for them to see large areas of the screen or easily move the pointer.

Along the same lines, it's important not to require a mouse, since some people either can't use one or find it extremely difficult. Every required action should be available from the keyboard. Clearly, we

can't insist on the reverse, that every action should be available via the mouse, since pointer devices don't provide alphanumeric input. However, you can assume that a user who can't use a keyboard at all will have access to a tool like the On-Screen Keyboard, and you should make all navigational and command actions available using the mouse.

Don't require the user to be able to hear. While using sound in an application can be a great enhancement (see my November article "Make Them Hear You"), you must make sounds visible, as well. One accessibility setting flashes the screen on system sounds, but if you're using sound to provide instructions, make sure the instructions are visible as well.

There's one issue we don't generally think about as part of the design of an application- documentation. But, for users to be able to learn your application, they need access to the documentation. The standard approaches may not work for users with disabilities. When possible, provide documentation in multiple formats—printed, online, audio, etc. If that's not an option, always choose the online format, since that's the easiest to convert to the others.

The Bottom Line

If you look back at the list of design principles above, you'll notice that most of them don't really address disability issues directly. They're simply good design principles, regardless of the audience. However, keeping users with disabilities in mind as you design should help you produce applications that make all your users happier.