

Building Vista Style Interfaces with Classic Windows Forms

Previously impossible UI is now reality with Telerik RadControls for WinForms.

For more than twenty years, the foundations that have underpinned Windows® development have remained stable. From the original GDI introduced with Windows 1.0 in 1985 to the latest GDI+ refresh in 2005, a generation of users and developers has become accustomed to the relatively simple interfaces that can be built for Windows. In the rapidly evolving world of software, few frameworks have proven their value and withstood the test of time like GDI.

As developers, this stability has allowed us to amass a wealth of resources and expertise that enable us to build powerful Windows applications with relative ease. The applications may not be the most visually stunning (especially if the use of less accessible styling methods, such as using bitmaps for buttons, is avoided), but they reliably get the job done. We have learned to accept the limits of Windows Forms like:

- No overlapping objects
- Limited transparency support
- Limited object transformations
- Difficult styling and control customization

“Good enough” is not enough anymore

Times are changing, though. With the release of Windows Vista® earlier this year, users are beginning to expect applications that match the new “clear and confident” interfaces of their operating system. Traditional GDI-based experiences that have been “good enough” for years now look strangely out of place and dated. As developers, we must rise to meet these expectations or watch as our applications get lost in a sea of visually impressive interfaces.

There are several ways developers can respond to this changing environment:

1. Do nothing, deny reality

The easiest response is to deny the importance of these new user experiences and insist that the current Windows Forms styling is still good enough. Burying your head in the sand won’t change the fact that your application looks dull,

though, and your users will quickly perceive your application as hard to use and behind the times.

2. Be an early WPF adopter

Alternatively, developers can embrace Microsoft’s new Windows development platform, WPF. WPF will eventually mature and replace Windows Forms-based Windows development, but that time is still years away¹. Adopting WPF today means wasting years of Windows Forms experience and starting over with a whole new development platform- a platform that still doesn’t officially have a visual IDE. Add to that the very specific .Net Framework version required to run WPF applications and you’re facing some serious hurdles to clear if you choose to adopt the infant WPF platform.

3. Deliver stunning visuals with Window Forms

Fortunately, there is a better alternative that blends the visual styling of Vista with the familiar Windows Forms development environment:

Telerik RadControls for WinForms

With RadControls for WinForms, you can leverage all of your existing Windows Forms knowledge to build applications that deliver the visually stunning experiences native to Vista. Build modern applications that run on all versions of Windows- from 2000 to Vista- without using the WPF runtime.

Telerik Presentation Framework

The Telerik Presentation Framework (TPF), like Windows Forms, runs on GDI+, but it uses advanced techniques to unlock the full power of GDI. A simple API and one of a kind design tools give developers easy access to these techniques without requiring any advanced knowledge of GDI. Without TPF, developers would need a strong grasp of the GDI+ APIs to achieve similar visual effects in Windows Forms.

Telerik’s Presentation Framework is a highly efficient and powerful visual rendering engine inspired by WPF best

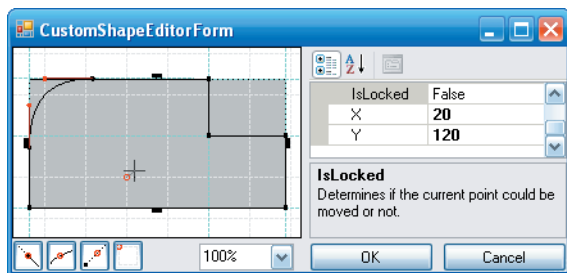
¹ According to Adam Nathans in Windows Presentation Foundation Unleashed, Sams, Dec. 2006

practices. It bridges the gap between Windows Forms and WPF, enabling you to deliver previously impossible visualizations with unique characteristics:

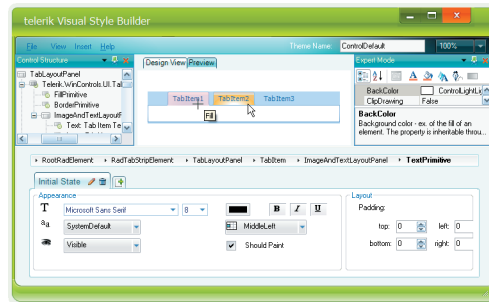
- **Scaling, zooming, and rotation** – You can manually achieve these effects with GDI+, but by default they are unaware of their surroundings. If a form is scaled, your custom GDI element does not change, resulting in awkward- often unusable- user interfaces. TPF is aware of its surroundings, though, and automatically handles these situations by intelligently updating elements and delivering better user experiences.
- **Alpha-blending and transparency support** – While visually blending two objects is possible with GDI+, it is impossible to control the transparency of nested Windows controls. RadControls solves the problem, delivering transparency through alpha-blending and transparency of nested items. Easily control the transparency of any RadItem via an Opacity property that affects text, border, fill, and even SubItems.



- **Animations** – Unique to Telerik's Presentation Framework, the RadControls animation engine allows any property of type number, color, size, or rectangle on any TPF control to be animated. Combined with Flash®-like easings, TPF can deliver smooth animations and transitions between control states that look much like Vista.
- **Shapes – the Shape Editor** allows you to easily draw any custom shape that you can imagine and apply that shape to any RadControl UI element. It even supports Bezier curves for complete control over your objects.



- **Application level skinning and themes** – Unlike WPF, the RadControls for WinForms includes a powerful Visual Style Builder tool to help you maximize the visual impact of your applications. Run from within Visual Studio® or as a standalone utility, the Visual Style Builder allows developers and designers to completely customize control appearance and behavior without writing any code. Customizations are saved in CSS-like XML files for easy re-use throughout a project.



- **Unlimited UI element nesting** – Put a RadButton in a RadMenu in a RadListBox. With the flexible and primitives-based TPF rendering engine, you can combine controls in ways previously unthinkable in Windows Forms.

The RadControls suite includes over 20 unique tools that take full advantage of these powerful UI features. From the Office 2007 inspired RadRibbonBar to mainstays like RadTreeView, RadGridView, RadMenu, RadPanelBar, and RadListBox to simple UI controls like RadButton, RadTextBox, and RadScrollBar, the suite includes everything you need to quickly build Vista-like applications.

In no time you'll have the slick Vista interface you never thought possible implemented in your Windows Forms application running on Windows XP.

The choice is clear. The world is changing and your users expect modern, visually stunning interfaces. You can deliver these experiences today, without the need to learn anything new, by simply adding the Telerik RadControls for WinForms to your next project.

The future has arrived: Vista-style controls for Windows Forms.

With the RadControls for WinForms you get most of the benefits of WPF and Vista in the familiar Windows Forms environment. You get a toolbox full of TPF controls, including the brand new RadGrid for WinForms. You get powerfully easy configuration tools like the Visual Style Builder and Shape Editor. You get to stop wishing you could build visually stunning applications and you get to start building them.

Test drive RadControls for WinForms today for free and see for yourself how easy it is to give your application the Vista-like interface you know it needs.

For more information and your free trial visit:

www.telerik.com/winforms