The Windows Presentation Foundation Roadmap: 2015 and Beyond

Whitepaper

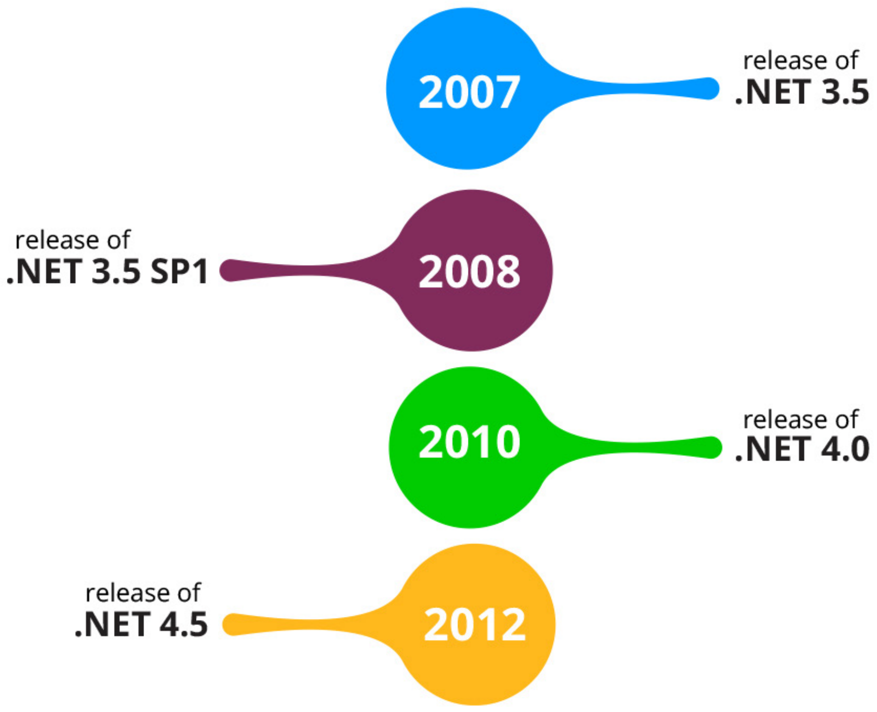
INFRAGISTICS

Introduction

When Microsoft first introduced Windows Presentation Foundation (WPF) in 2006, it represented a real step forward in application design and development. WPF allowed developers to create intuitive User Experiences (UX) for both standalone and browser based applications. It simplified the development of complex GUI applications, helped to separate UI and business logic, and provided a consistent programming model. Not long after WPF's first release, Infragistics released NetAdvantage for WPF 7.1 which provided developers with all the tools they needed to create more modern, mission critical line of business applications using WPF.

Since WPF was first launched as part of the .NET Framework 3.0 in 2006, it has seen four major revisions since then. First with the release of .NET 3.5 in 2007, then .NET 3.5 SP1 in 2008, followed by .NET 4.0 in 2010, with the last major update being .NET 4.5 in 2012.

Major Revisions to WPF



With each release of WPF, Infragistics was there supporting their customers and community with great new controls and enhancements. By releasing two to three NetAdvantage for WPF volume releases per year, Infragistics customers were always up to date with the latest and greatest that WPF had to offer. Given that the last major update to WPF was in late 2012, this extended period of silence led to confusion and doubt surrounding WPF's future. Although during this time Infragistics continued to show its dedication to WPF by continuing to release new controls and updates to their WPF product, companies looking to use WPF for their application development were concerned about using a technology that Microsoft appeared to have abandoned. If the platform were to receive no further improvements, regardless of the investment Infragistics was making, companies risked investing in a technology with no long term prospects.

WPF is used by many organizations to build applications and tools that support day-to-day business processes. Any interruption to these processes, or risks to the tools delivering them, causes real uncertainty, and so stakeholders had become nervous about using a technology which Microsoft appeared to have left behind. Major concerns centered on:

1. The technology underpinning WPF would not improve and no new features would be added. It would become difficult or even impossible to improve apps built with WPF over time and bring them in line with modern UI standards.

2. Support for the framework would gradually diminish. While Microsoft provides software support agreements for their tools and services, they do not cover every eventuality. Support is mainly extended to bugs, so less pressing issues such as poor performance would not usually be covered by this kind of agreement and consequently would not receive support.

Using WPF to build business critical applications might, therefore, appeared unwise in the long term. However, recent developments have given the WPF community cause for hope.

This Whitepaper summarizes the history of WPF, highlights the strengths of the platform and, most importantly, will analyze its future. It will also discuss the commitment Infragistics has made, and continues to make, to the WPF framework.

A Brief History of Windows Presentation Foundation

Summary of the Technology

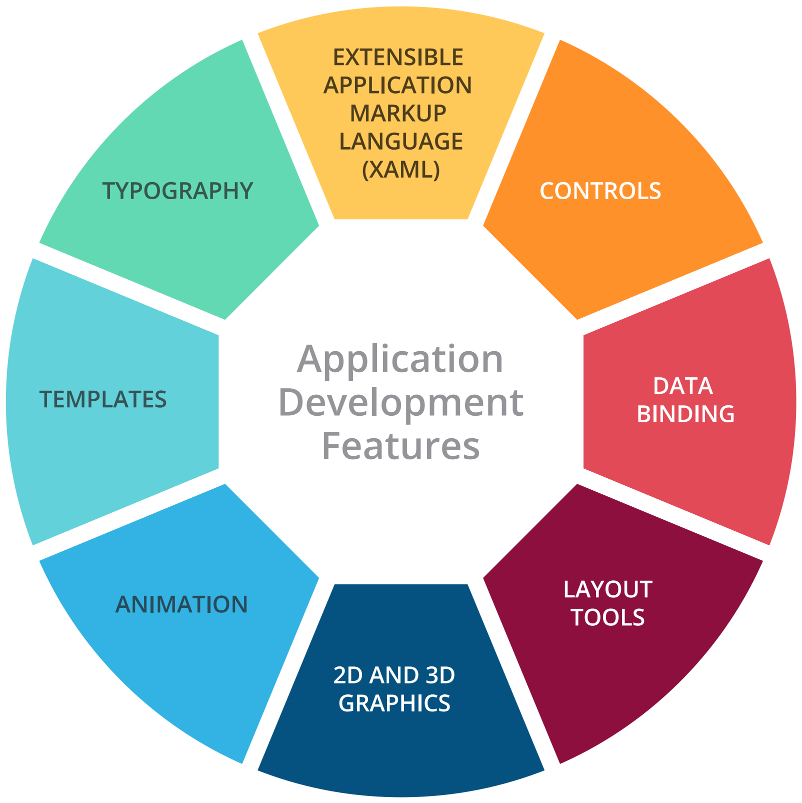
Windows Presentation Foundation (WPF) is a pillar of Microsoft's .NET Framework and was first released as part of version 3.0 in 2006. WPF is a graphical subsystem and renders User Interfaces using DirectX. Prior to WPF, graphics would be rendered with the Windows Graphics Device Interface (GDI). WPF was in part designed take advantage of modern graphics hardware that was starting to appear at this time.

WPF lets developers separate the building of an application's UI from its functional architecture, it also provides a standard programming model with a comprehensive set of application development features. Put simply, WPF aids in the development of visually stunning applications.

WPF is widely seen as an improvement on WinForms, an older platform used for rapid application development, and widely known for its battleship gray UIs. While WinForms continues to be popular for developers building event-driven Windows desktop applications, it supports limited customization and flexibility. WPF is a more modern technology and is less restricting in terms of what can be created.

For instance, a dialog box built for an app in WinForms would typically conform to the standard shape and style of a regular Windows dialog box. It is hard to create anything more complex. In WPF on the other hand, the developer has far greater control when choosing the color, shape, and means of interacting with the dialog box. They can still easily create 'standard' interactions, but WPF also supports much more customized interfaces and design elements.

As users expect more from interfaces and general usability, developers can combine the features of WPF, with the power and functionality of Infragistics WPF controls to create a greater scope for designing engaging and inspiring applications.



Common Features and Tools

There are numerous reasons why a developer would choose to build an app with WPF:

- Allows the developer to separate view concerns and business logic through data binding. This makes it easier to develop, maintain, test and extend applications. This architecture allows the use of design patterns like Model View controller (MVC) or ModelViewViewModel (MVVM) in developments.
- WPF applications make use of C# and Visual Basic, which have had widespread adoption for enterprise application development. This ubiquity means they are well supported on the web, with many samples, forums and active communities
- UI is defined with XAML, a powerful markup language used to express animation, graphics, data-binding and multi-media. Using XAML significantly reduces the amount of code developers need to write when they employ these techniques in their applications.
- WPF allows users to easily create UIs which adapt to different screen shapes, sizes and resolutions
- WPF itself has a large and active online community and third party support (not least from ourselves)

Microsoft's Apparent Distancing from WPF

Between 2011 and late 2014, Microsoft posted no new content on the WPF blog, nor on any of their dedicated developer resources. Over the same period there were no significant updates or new releases for WPF. This lack of communication from Microsoft left many developers in doubt as to whether the platform was in the process of being abandoned. As new technologies like Windows Universal Apps and Xamarin have emerged, the community feared that WPF was no longer a priority for Microsoft.

While WPF would certainly continue to receive support, there was real concern that it would become a 'legacy' technology. Developers were afraid that they would be using a platform without a future.

However, on November 12, 2014 the Microsoft WPF Team broke their silence with a blog post, and confirmed that they had in fact never stopped working on the platform. The announcement promised a number of new releases and updates inspired by requests made on various developer forums. Since then, the team has released a number of posts on the WPF blog about their continuing work, and have also made a number of Channel 9 video appearances discussing major new features being develop for WPF such as 'WPF Local'.

.NET, Open Source and WPF

As a company, Microsoft is undergoing considerable change at present in terms of strategy and direction. Part of this new approach has been the release of a large swathe of .NET to the open source community at the end of 2014. By making previously protected source code public, the corporation indicated a move to greater openness. However, while WPF is a component of the .NET Framework there appear to be no plans to release it as open source as yet.

According to Soma Somasegar, Corporate Vice President of the Developer Division at Microsoft, preparations for releasing .NET code as open source had been underway for around three years. Releases have emerged every few weeks since the end of 2014 and have been placed on GitHub.

Major releases so far include:
- .NET compiler platform ('Roslyn')
- .NET Core 5
- ASP.NET 5
- .NET Base Class libraries
- .NET Web, Data and API frameworks

For over ten years the Mono Project has built .NET compatible tools to support developers wishing to build cross-platform .NET applications. While Mono allowed developers to use components of .NET when building apps for Mac iOS, Android or Linux, the lack of original source code from Microsoft inhibited the process. With the move to open source, Microsoft is now firmly embracing cross platform support for .NET. The Mono project continues and they will be contributing to the .NET efforts and bringing in the best bits to Mono.

Soma Somasegar characterized the thinking behind this move to open sourceas such:

“We have to meet developers where they are as opposed to saying, 'Hey, you come to where we are.'”

There is a general move within Microsoft to become more open and transparent with developers. Nonetheless, there are a number of strategic decisions which informed this choice too. When announcing the move in a November 2014 blog , Immo Landwerth (Program Manager on the BCL team at Microsoft) was clear about the reasons behind this new approach:

- The way developers work has changed: no longer do they build for one platform; instead they typically try and build across platforms – for Mac, iOS, Android and Linux
- Consumers and enterprises access IT from a much wider variety of media than ever before. Making the code behind .NET available for professionals building apps for smart phones, tablets and other operating systems expands Microsoft's reach
- Releasing .NET to the open source community has potential benefits for Microsoft's own projects. Microsoft now have access to contributions from many more developers.

The release of .NET Core as open source is symbolic of a wider move within Microsoft's strategy. It stems from a recognition that cross platform development is increasingly common and it simply makes more sense to make apps available on different device types and operating systems. It is also a recognition that app stores symbolize a more open playing field, where developers anywhere in the world can contribute to and build their own applications. Allowing access to Microsoft's source code will help these individuals contribute much more easily.

Despite these changes, the desktop still has a future and while the world is changing rapidly, desktop applications are still incredibly important in enterprise IT (for example, it is currently just not feasible to write or even read lengthy reports on a tablet or mobile phone screen).

To quote Jay Schmelzer (Microsoft's Director of Program Management):

"As a company we've said that we believe the direction for client side development on a Windows device is more WinRT, the Store app, the modern app. But we acknowledge there's a gap there right now, you can't accomplish everything you want with that stack. For a desktop application which is keyboard and mouse oriented, WPF is the place to go."

Through this statement, Schmelzer confirms that unless the desktop ever completely disappears, there will always be a need for apps which function on the hardware. While Microsoft want to encourage Universal App development, the kind of enterprise desktop tools developers build often simply do not need to be made available on smartphones and tablets. WPF, therefore, has a long future ahead of it.

The WPF Roadmap

After almost three years of silence, the announcement of a new WPF roadmap provides real assurance to enterprises and developers alike. For those doubting whether their applications are built on 'legacy' technology, the confirmation that WPF has a future provides real peace of mind and confidence going forward with their productions.

The release of the WPF roadmap, and Infragistics continued commitment to WPF, means businesses can commit to the technology and expect updates and investment going forward.

A Raft of Improvements

When the WPF roadmap was released, Microsoft explained that they had been listening to requests and comments from the community and undertaking interviews with developers.

The blog post announced plans for the future, as well as publicizing a number of fixes to the platform, including:

- Multi-image cursor file support in System.Windows.Input.Cursor
- Support for transparent child windows.
- Improved double tap gesture recognition by using common threshold distance from registry.
- Improved text selection through double tapping the WPF TextBox control.
- Improved reliability of stylus input for the WPF ComboBox control.

Major Milestones

Looking forward, Microsoft plans to improve WPF in four major areas as follows

Performance
Tooling
Directx Interoperability
Supporting Modern Hardware

Performance

A recurrent complaint about WPF has been that it can be frustratingly slow. Microsoft have therefore announced their plans to improve performance in the following areas:

- Startup
- Scrolling
- Virtualization performance of ItemsControls

These improvements will allow developers to write applications that run more efficiently and are more responsive to end-user interactions. This will make developing a WPF application more appealing to many and should result in an increase in the number of developers using the platform. Infragistics will also leverage these performance improvements in their WPF controls to give their customers the competitive advantage needed to meet the demanding requirements and expectations of modern applications.

DirectX Interoperability

The Roadmap includes updates and improvements to WPF's interoperability with the various DirectX APIs. Despite being a platform for creating rich UIs, calling DirectX APIs is often surprisingly complicated. Furthermore, much of DirectX has advanced in recent years and WPF has not kept up. Microsoft is addressing this in its coming updates.

As desktop applications become ever more complicated and need to be able to operate with modern video, streaming, and 3D images, WPF will need to work more effectively with DirectX. The roadmap promises this will happen.

Supporting Modern Hardware

Since WPF was launched in 2006, the way users interact with IT has changed dramatically; enterprises and individuals interact with a much wider range of hardware at work. Although it is true that smartphone and tablet sales are growing, desktops and laptops remain the primary means of carrying out 'heavy weight' and intensive tasks such as roles which involve analyzing Excel data across multiple screens.

Desktop technology continues to improve and Microsoft recognize the importance of ensuring the WPF platform is best prepared for advances in the hardware. The Roadmap provides developers with the confidence that they will be able to design apps which make use of new technologies such as touch screen and high density (4k) displays.

Tooling

When the WPF team began their research with developers to find out ways of improving the platform, the need for new tools was consistently reported as a top request. As a result, the team will be refining and advancing tools for:

- Visual diagnostics – tools to help debug apps in a live visual tree
- Timeline tools – to help developers troubleshoot problems with their builds and improve performance
- Blend improvements – Blend for Visual Studio 2015 will help developers make sleeker, smoother UIs

Infragistics will be utilizing these tools to help improve the quality of their WPF controls, and provide better design time experiences for their customers using Visual Studio 2015.

The Infragistics Roadmap

While Microsoft has been quiet about the future of WPF up until recently, Infragistics has never stopped investing in the WPF platform since it started back in 2007. Since 2012, Infragistics has released over a dozen new WPF controls, multiple frameworks, and 100's of new features to existing WPF controls. Recent releases of Infragistics for WPF included brand new controls and features to help support customers creating modern Office inspired applications such as Microsoft Outlook, Excel, Word, and Project. The developer experience has also been improved with an increased emphasis on updating their WPF controls with improved MVVM and data binding support. New WPF sample applications have been created to provide developers with guidance and best practices for creating enterprise line of business applications with WPF and Prism.

Unlike most companies, the Infragistics roadmap for their WPF controls are greatly influenced by their customers. This is a refreshing quality that a lot of companies have lost touch with over the years. Here are a few items that Infragistics customers have been asking for recently.

- PDF Library and Viewer
- New controls, such as the highly requested 3D Surface Chart and Busy Indicator.
- Property Grid improvements
- Spreadsheet improvements
- Rich Text Editor improvements
- Theme improvements
- Improved Visual Studio design time experiences
- Improved WPF sample browser

Since the Infragistics WPF roadmap is customer driven, it is also constantly changing depending on customer demand at any given point. Infragistics is dedicated to providing their customers with the tools and support they need to be successful developing WPF applications

Committing to WPF with Confidence

For almost three years Microsoft was quiet about WPF and its future. As a result the community, businesses and developers alike were concerned that the platform had no future. However, we now know that Microsoft was undergoing a period of considerable internal development. Not only was it attempting to change computing with Windows 8 and the Modern UI, but it was looking at Universal Apps and opening sourcing its .NET Framework.

With a clearer picture starting to emerge on those big themes, Microsoft has now finally reconfirmed their commitment to improving and developing the WPF platform. Their now regularly updated blog feed, the roadmap announcement, and increased openness has instantly provided organizations that use WPF with the reassurance and the security they need.

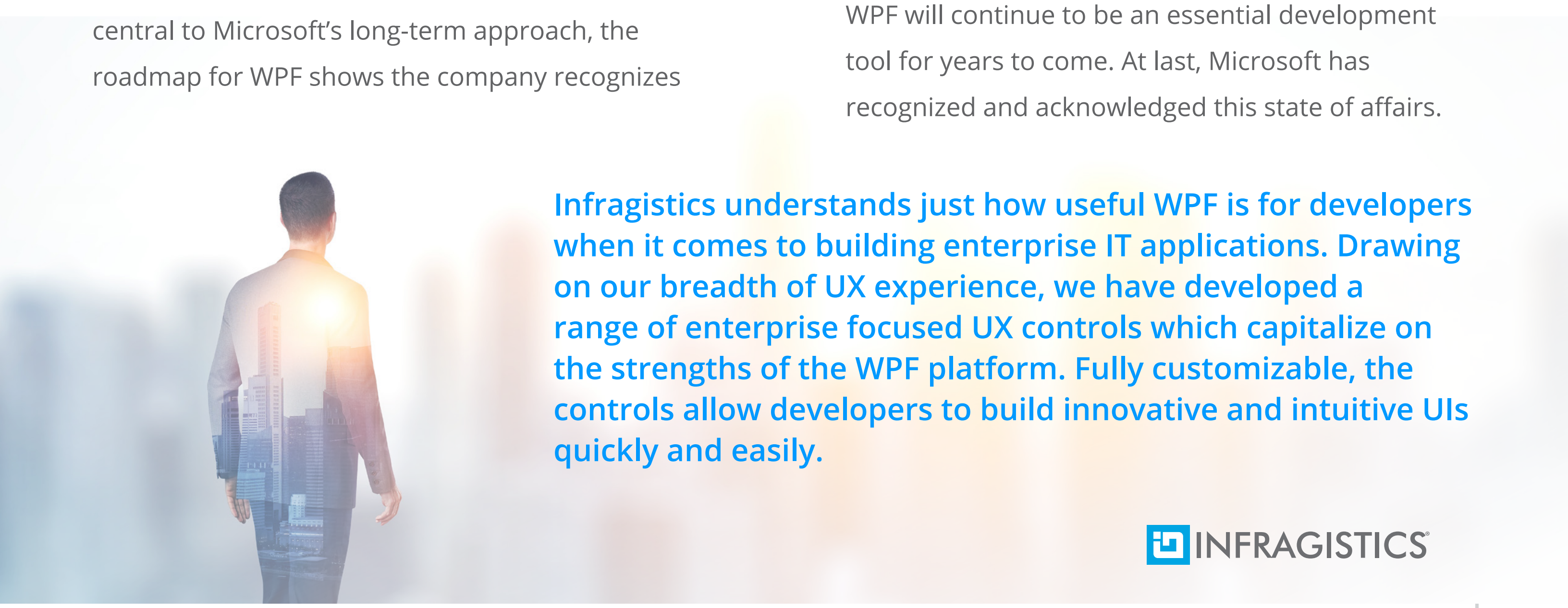
Microsoft has committed to a new approach to building applications – Universal apps. This strategy has come about as a response to the explosion in new devices in recent years, and the repositioning of Windows' role. While Universal apps are clearly central to Microsoft's long-term approach, the roadmap for WPF shows the company recognizes that the world has not moved on from the traditional desktop just yet.

The WPF Roadmap confirms Microsoft's dedication to improving the platform in the short and long term and they have committed to improving:

- Performance
- DirectX interoperability
- Supporting for modern hardware
- Tooling

Infragistics will also continue to provide developers with the tools they need to create a variety of WPF applications, ranging from Office inspired solutions, to financial dashboards and indicators, to data intensive ERP and engineering systems.

The WPF Roadmap - and Infragistics commitment to creating new controls and features to support WPF development - provides organizations with the confidence that their WPF applications will be supported in the coming months and years. Although for many, the desktop is no longer the dominant means of interacting with IT, it remains a major player and will continue to exist in its current form for the foreseeable future. This being the case, WPF will continue to be an essential development tool for years to come. At last, Microsoft has recognized and acknowledged this state of affairs.



Infragistics understands just how useful WPF is for developers when it comes to building enterprise IT applications. Drawing on our breadth of UX experience, we have developed a range of enterprise focused UX controls which capitalize on the strengths of the WPF platform. Fully customizable, the controls allow developers to build innovative and intuitive UIs quickly and easily.

INFRAGISTICS