PRESS RELEASE

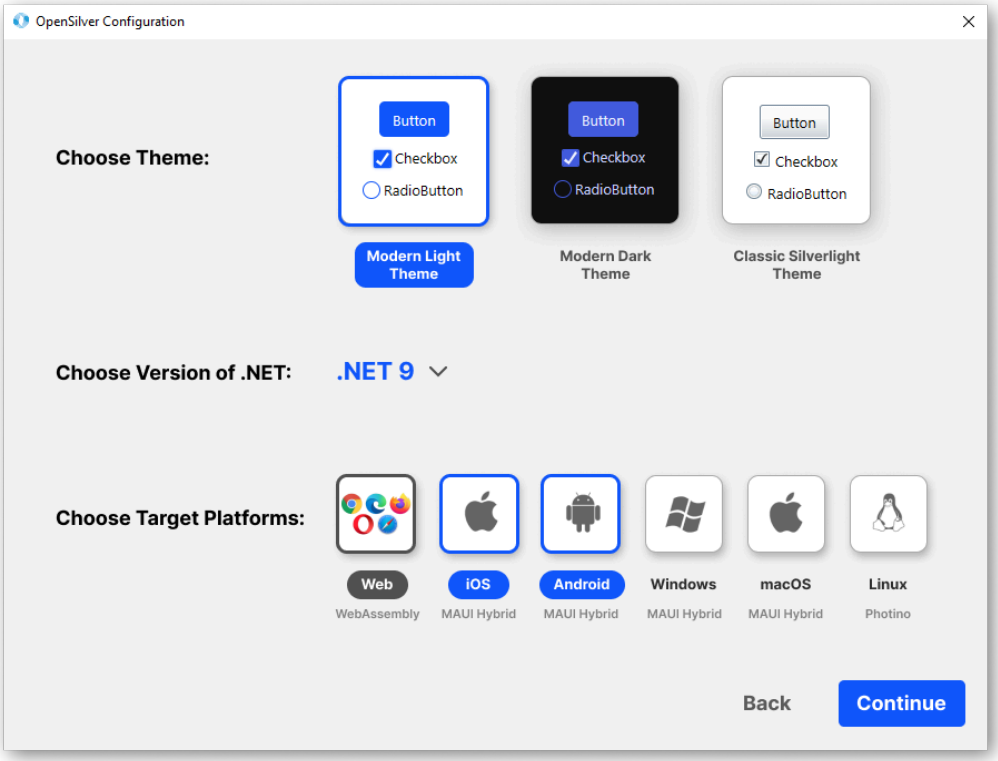
OpenSilver 3.2 Brings WPF Apps to iOS, Android, and Beyond via MAUI Hybrid

WPF Everywhere with OpenSilver 3.2: From Desktop to Web to Mobile

Paris, France – March 18, 2025 – Userware today announced OpenSilver 3.2, enabling developers to deploy WPF-compatible applications to iOS, Android, Windows, Mac, Linux, and web browsers from a single codebase using .NET MAUI Hybrid integration. While web support was available previously, this release extends reach to mobile platforms. Migration to OpenSilver's supported WPF subset is required, but preserves significant development investments while expanding platform reach.

Breaking the Platform Barrier: WPF Beyond Windows

For enterprise .NET teams with substantial WPF applications, mobile deployment has traditionally presented a difficult choice between complete rewrites or maintaining separate codebases. OpenSilver 3.2 takes the framework beyond browsers by leveraging .NET MAUI Hybrid to compile C# to native code while rendering UI through WebView.



OpenSilver 3.2 'New Project' dialog inside Visual Studio

"This release represents a fundamental shift in how developers can approach WPF modernization," said Giovanni Albani, CEO of Userware. "Instead of choosing between maintaining a desktop-only application or starting from scratch for mobile, teams can now extend their existing WPF investments across all major platforms with reasonable adaptation efforts."

The approach resembles Blazor Hybrid but maintains XAML as the UI language instead of switching to HTML. Applications can be deployed to multiple platforms by adding platform-specific projects to the solution. An ASP.NET project serves as the entry point for web browsers (a capability available in previous versions), while the new MAUI project serves as the entry point for iOS, Android, Windows, and Mac deployments. For Linux targeting, developers can add a Photino project as an additional entry point.

Migration Pathway vs. Traditional Approaches

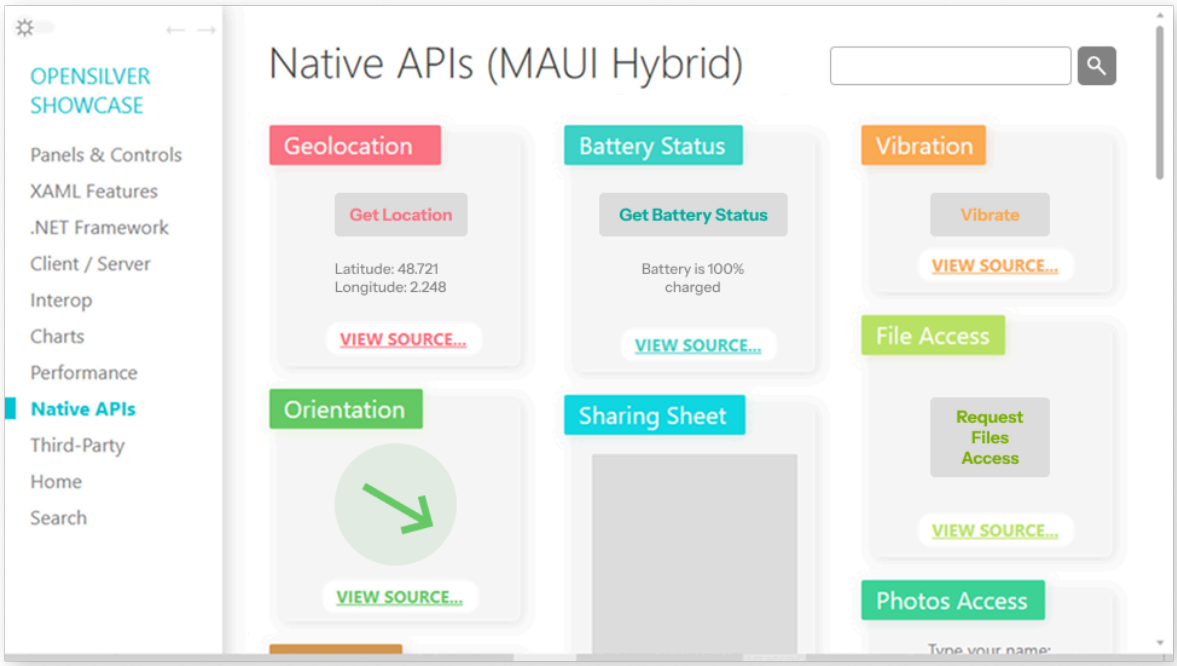
Deployment Target	Traditional Approach	OpenSilver 3.2 Method
Web	Complete UI rewrite	Adapt to OpenSilver's WPF subset
Mobile (iOS/Android)	Platform-specific implementations	Add MAUI Hybrid entry points
Desktop (Windows/Mac)	Multiple UI frameworks	Same codebase, different entry points

Porting applications to OpenSilver begins with migrating to the framework's supported subset of XAML and C#. Migration complexity scales with application size and feature usage—from hours for small projects to a few months for enterprise-scale applications with millions of lines of code.

"What's compelling about OpenSilver's approach is how it preserves development investments," explained Darshin Vyas, VP of Sales at Userware. "We've seen clients maintain a single codebase with platform-specific functionality isolated to conditional statements. It's about adapting to OpenSilver's supported WPF subset rather than completely rewriting your application."

Proof in Production

To demonstrate the capabilities, Userware has released two open-source sample applications. The first is a full-featured task management app in production that includes native Calendar integration and is available on App Store, Google Play, and other platforms. The second is the updated "OpenSilver Showcase" app, which contains numerous small samples demonstrating native API integration including sensors, vibration, location, flashlight, notifications, file access, and share sheet functionality. Both applications are available on GitHub for developers to explore.



Native API access samples in the OpenSilver Showcase app

The primary adaptation required involves front-end architecture. Due to security limitations of web and mobile platforms that restrict direct database access and unrestricted file I/O, these operations need to be moved to backend services. OpenSilver supports multiple service integration types, including REST, SOAP, WCF, and RIA Services, alongside select third-party libraries such as Telerik. The framework's ecosystem is expanding rapidly, and users can also leverage JavaScript-based alternatives via C#/JS interop for components not yet natively supported.

Most business logic and XAML design can be preserved, though not all WPF features are fully implemented yet. The framework ensures forward compatibility—everything OpenSilver supports works in WPF, even if full WPF feature parity remains in progress.

The WebView Advantage vs. Alternatives

While multiple approaches exist for cross-platform development, OpenSilver offers unique advantages for teams with WPF expertise:

- **vs. Complete Rewrite:** Preserves existing code investment and specialized business logic
- **vs. Blazor:** Maintains XAML/C# development model and leverages existing WPF skills
- **vs. Native MAUI**: Delivers consistent UI across platforms with less platform-specific code
- **vs. Hybrid Frameworks:** Provides deeper .NET integration and familiar tooling

Unlike alternative frameworks that render UI using native controls, OpenSilver uses a WebView-based approach that prioritizes consistency over platform-specific optimizations.

"For enterprise and line-of-business applications, the ability to maintain pixel-perfect UI consistency across platforms often outweighs the benefits of platform-native controls," said Albani. "Our customers value giving users the same experience whether they're on a desktop browser, an iPad, or an Android phone. The migration effort to adapt to OpenSilver's WPF subset is offset by the significant time saved not having to maintain separate UIs for each platform."

This browser-first approach accelerates development by avoiding platform-specific UI implementations and, by leveraging web technologies, enables packaging into native applications for a wide range of platforms, including targets like Linux via Photino. The C# logic compiles to native code while MAUI Hybrid provides access to platform APIs needed for app store distribution. An additional benefit is the ability to incorporate web ecosystem components via C#/JavaScript interop.

Future-Proofing Development Investments

OpenSilver 3.2 serves both legacy migration needs and new development projects. Microsoft's Build 2024 endorsement of WPF for desktop development aligns with OpenSilver's strategy of extending that productivity model to additional platforms.

"We're seeing increasing interest not just in migration scenarios but also from teams starting new projects who value XAML's productivity and want cross-platform reach without learning multiple UI frameworks," said Vyas. "OpenSilver includes the first-ever XAML designer for VS Code, plus one for Visual Studio - both featuring unique drag-and-drop WYSIWYG capabilities with AI functionality. These designers can also be experienced online at XAML.io without installation. Coupled with support for C#, VB.NET, and F#, this makes the platform especially appealing for .NET teams willing to work within OpenSilver's supported WPF subset."

Rather than positioning OpenSilver as merely a legacy migration tool, Userware emphasizes its role in creating sustainable development strategies. "Organizations can adopt a 'develop once, deploy everywhere' approach that significantly increases their agility," notes Albani.

Additional WPF Enhancements in 3.2

Beyond the headline MAUI Hybrid integration, OpenSilver 3.2 includes numerous WPF compatibility improvements, including:

- Full support for right-to-left layouts
- WPF-like event bubbling system
- Advanced WPF animations
- Enhanced VirtualizingScrollPanel performance
- Scroll inertia for improved mobile experience

- And many other WPF-aligned enhancements

A detailed list of changes and screenshots of the new features can be found on the official blog announcement at https://opensilver.net/announcements/3-2/

Future Roadmap

Having completed full Silverlight support (which was itself a cross-platform WPF subset with a similar "WPF Everywhere" vision), development focus has now shifted to expanding WPF compatibility. Future releases will enhance third-party component support and improve XAML.io, the framework's online designer currently in preview that can be accessed without installation. Integration with Blazor components and 3D UIs via XRSharp.io are in development, alongside an as-yet-unrevealed feature that Userware expects will significantly accelerate development and deployment times, scheduled for late 2025.

Getting Started

OpenSilver 3.2 is available immediately at opensilver.net/download, as well as through the Visual Studio and VS Code Marketplaces for convenient installation and updates. The release includes integration extensions (VSIX) for both Visual Studio and VS Code, along with comprehensive documentation for adding MAUI Hybrid entry points to existing projects.

Two sample applications are available on GitHub: a task management app demonstrating production-level capabilities on mobile, web, and desktop, and a showcase app highlighting native API integration.

Userware will host a launch webinar on April 8, 2025, at 11 AM ET featuring a live demonstration of cross-platform development, migration strategies, and Q&A with the development team. Register at http://opensilver.net/webinar-for-v3-2

—

About OpenSilver

OpenSilver is an open-source platform that enables .NET developers to build cross-platform applications using familiar C# and XAML. Originally created as a modern reimplementation of Microsoft Silverlight, it has evolved into "WPF Everywhere" – extending beyond Silverlight's scope to bring Windows Presentation Foundation capabilities to virtually any platform.

The platform compiles to WebAssembly and JavaScript, allowing applications to run natively in all modern browsers without plugins while supporting deployment to mobile and desktop environments. OpenSilver empowers developers to:

- Modernize existing WPF, Silverlight, LightSwitch, and Windows Forms applications
- Create new business applications with a single codebase for web, mobile, and desktop
- Develop with C#, VB.NET, or F# using familiar XAML-based UI design
- Leverage the AI-assisted WYSIWYG designer in Visual Studio and VS Code

For developers, OpenSilver offers the productivity of WPF with the reach of web technologies, eliminating the need to learn multiple UI frameworks while enabling deployment across virtually any device or operating system.

For more information, visit https://opensilver.net

—

About Userware

Userware

Founded in 2007, Userware specializes in application modernization and cross-platform development. The company maintains OpenSilver as an open-source project while offering migration services for WPF, Silverlight, and LightSwitch applications. Userware also develops XAML.io for browser-based design and XRSharp.io for 3D experiences, alongside VR/AR solutions and AI integration services.

For migration services, visit: https://opensilver.net/contact

Userware can be found at https://userware.dev
—

Press Contact:

Vasil Buraliev
Media Relations at Userware
Email: vasil.buraliev@userware.dev