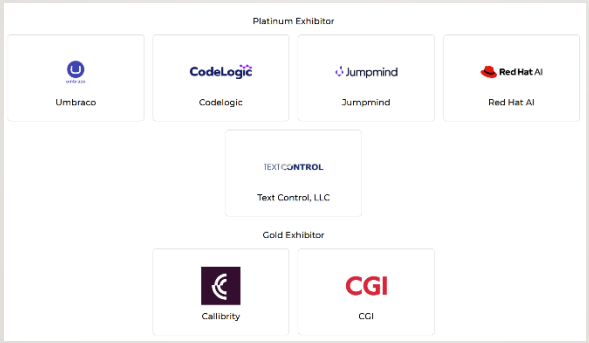
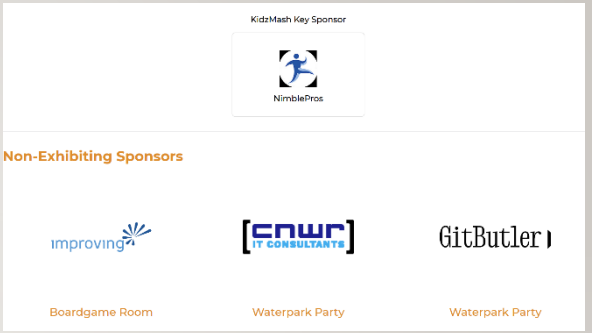
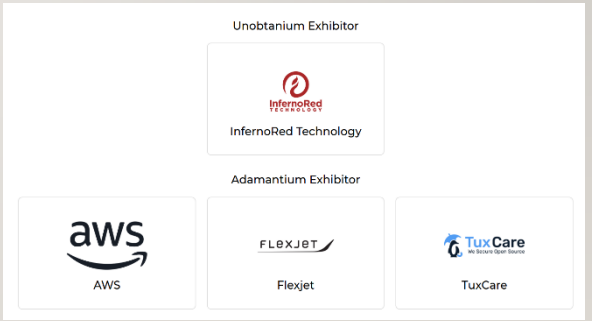
CREATE MULTI-PLATFORM APPS WITH .NET MAUI & THE MAUI COMMUNITY TOOLKIT

- **Alvin Ashcraft**
- **Sr. Content Developer, Microsoft**

THANK YOU CODEMASH SPONSORS!



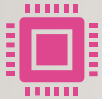




AGENDA

- About Me
- Introduction to .NET MAUI and Cross-Platform Apps
- Develop on Windows and macOS
- Rapid App Building with XAML, C# & Visual Studio
- Enhance Apps with the .NET MAUI Community Toolkit
- Simplify Your MVVM Architecture with the MVVM Toolkit
- Build Apps with C# Markup in VS Code: No XAML Required
- Style Apps with CSS
- Access Platform-Specific Features
- Foundry Local models on Windows & Android (time permitting)
- Q&A

ABOUT ME



Software industry for 30+ years

Started programming in 1995 (VB 3 & Microsoft Access)

Content developer at Microsoft since 2022

Morning Dew link blog

Daily links for .NET developers since 2007

Three books for Packt Publishing

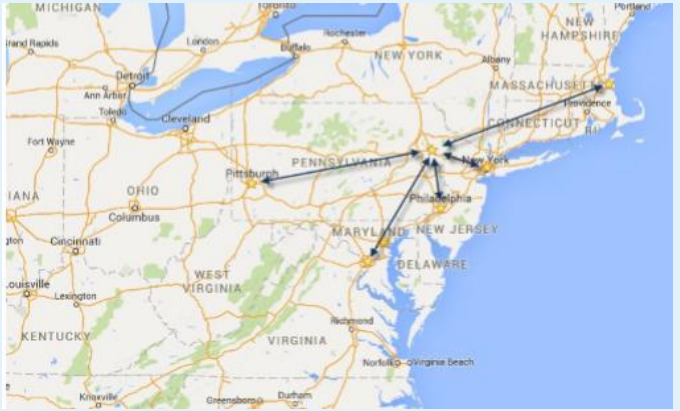
Learn WinUI 3 (two editions)

Parallel Programming and Concurrency with C# and .NET

TechBash dev conference organizer since 2016

techbash

October 13-16, 2026 | Pocono Manor, PA



- Great speakers with top content
- A fraction of the cost of the more crowded conferences
  - 3-day conference plus lodging for ~$1000
- Full-day deep dive preconference sessions available
- Easy travel from almost anywhere
- World-class keynotes
- In addition to the sessions, you get a great hallway Track, amazing food, attendee Welcome Reception, Game Night & more
- Family Day Friday - full day of kids' sessions, free for attendees' families
- Discounted Kalahari Resort rooms with water park access: stay, learn & play all week

https://techbash.com or **@techbash**

OVERVIEW OF .NET MAUI & SUPPORTED PLATFORMS

- Multi-platform Framework
  - .NET MAUI allows building apps for multiple platforms using a single shared codebase.
- Supported Platforms
  - .NET MAUI supports Windows & macOS on the desktop and the iOS & Android mobile operating systems enabling cross-device compatibility for most app users today.

BENEFITS OF A SINGLE CODEBASE

- Reduced maintenance
  - A single codebase simplifies updates and bug fixes, lowering the time and resources needed for maintenance.
- Faster development
  - Developers can build and deploy features faster by writing code once for all platforms.
- Consistent user experience
  - Using a unified codebase ensures users enjoy the same experience across devices and operating systems while still providing native performance and look & feel.

MAUI VS. OTHER X-PLAT FRAMEWORKS

- Language
  - Ideal choice for .NET developers. Frameworks like Flutter and React Native require other technical expertise.
- Enhanced Performance
  - Enables faster and more efficient app execution compared to PWAs or React Native.
- Community and Ecosystem
  - Puts Microsoft in your corner. 3rd party libraries & controls are available but not required for building rich, performant apps. The .NET MAUI Community Toolkit and .NET Community Toolkit offer additional libraries supported by the open-source community.

IDES AND TOOLS

- Primary IDEs
  - Visual Studio on Windows and VS Code on Windows and macOS serve as the main development environments for .NET MAUI development.
- Supporting Tools
  - VS Code, GitHub Copilot, and CLI tools complement the primary IDEs by supporting flexible and lightweight development workflows. .NET MAUI Community Toolkit offers open-source controls and helpers for your apps.
- Additional Options
  - JetBrains Rider supports .NET MAUI development on Windows and macOS.

DEBUG & BUILD ACROSS PLATFORMS

- Debugging Tools Usage
  - Use debugging tools like XAML Hot Reload, Live Visual Tree, and Visual Studio's Binding Failures window to identify and fix issues effectively during development across platforms.
- Simulators and Emulators
  - Simulators and emulators for Android and iOS replicate different platforms, enabling real-time testing without physical devices. Connect to a Mac from Visual Studio on Windows to debug iOS apps from your favorite IDE.

PRODUCTIVITY TIPS

- Hot Reload Feature
  - XAML Hot Reload & .NET Hot Reload enable instant code and markup changes, speeding up development & reducing testing time.
- Use of Templates
  - Templates provide reusable structures, enhancing consistency & saving time coding.
- Extensions for Quality
  - Visual Studio & VS Code extensions add functionality & help maintain high code quality across platforms.

BUILD UIS WITH XAML OR C#

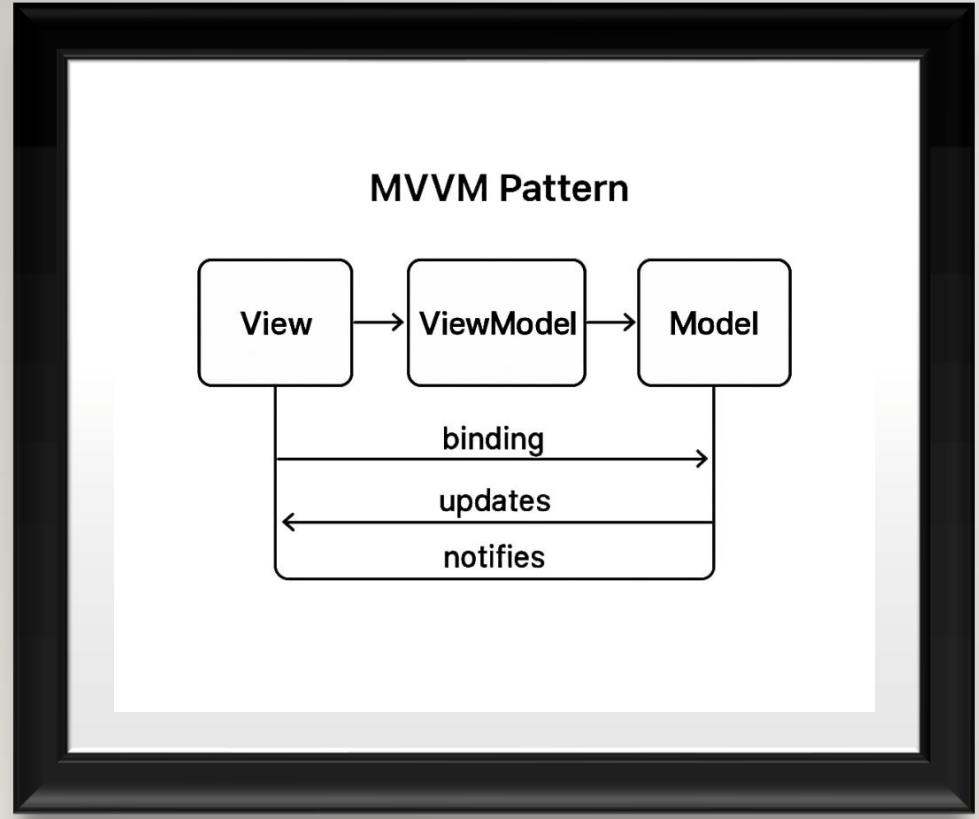
- XAML markup for clear & declarative UI design when building responsive UIs.
- Code behind in C#
- Separation of concerns with MVVM

- Create UI in code with C# Markup
- MVU pattern for robust apps
- No XAML learning curve

Demo: .NET MAUI projects in Visual Studio 2026

TOOLKIT OVERVIEW

- Reusable UI Components
  - Microsoft's Community Toolkits provide reusable components that speed up UI development and maintain consistency.
- Functionality Enhancements
  - Extensions, converters & animations add new features that extend the capabilities of existing UI elements.
- Improved Responsiveness
  - Behaviors improve app responsiveness by managing user interactions smoothly.
- Demo
  - Toolkit docs on Microsoft Learn
  - MAUI projects with Toolkit features



MVVM IN MAUI APPS

- Separation of Concerns
  - MVVM separates UI from business logic for cleaner & modular code design.
- Improved Code Organization
  - Using MVVM enhances code maintainability and organization in .NET MAUI applications.
- Easier Testing
  - MVVM facilitates unit testing by isolating business logic from UI elements.

REDUCE BOILERPLATE WITH MVVM TOOLKIT

Minimizing Repetitive Code

Base classes designed to reduce repetitive coding tasks efficiently.

Streamlining Development

Helpers in the toolkit streamline the development process, enhancing productivity & maintainability.

BUILD SCALABLE APPS

- Data Binding
  - Use data binding to synchronize UI and data models efficiently, enhancing app responsiveness and maintainability.
- Command Patterns
  - Apply command patterns to encapsulate requests as objects, facilitating flexible and reusable code.
- Dependency Injection
  - Use dependency injection to manage component dependencies, promoting modular and testable code architecture.
- Demo – Use the MVVM Toolkit in Your Project

C# MARKUP

Programmatic UI Creation

Create and modify user interface elements using C# markup programmatically.

Improved Code Reuse

Enhance code reuse by allowing modular UI components to be managed efficiently.

Reduced Context Switching

Using C# markup reduces context switching between design and code environments for developers.

RECOMMENDED SCENARIOS

- Dynamic Layouts
  - Get flexible, dynamic layouts that adjust automatically based on data and user interactions.
- Data-Driven UI Generation
  - UI elements can be generated dynamically from data sources, improving responsiveness and customization.
- Demo – Create a Project w/ C# Markup

REUSE WEB SKILLS WITH CSS IN MAUI

Leverage Web Development Skills

Utilize your existing web dev knowledge to efficiently style MAUI applications.

Save Development Time

Reusing web assets reduces styling time, accelerating app development & deployment.

Maintain Brand Consistency

Consistent styling across web & MAUI apps ensures strong & uniform brand identity.

USING CSS IN MAUI PROJECTS

- Including CSS Files
  - Learn the process to add CSS files to a MAUI project to manage styles separately from code and XAML.
- Applying Styles to Controls
  - Apply CSS styles effectively to UI controls for a clear separation between design & app logic.
- Demo – CSS in .NET MAUI projects

DEVICE FEATURE ABSTRACTION

Unified API Access

Unified APIs simplify device feature access across multiple device platforms.

Sensors Integration

APIs enable consistent access to device sensors like accelerometers & gyroscopes.

Camera and Storage

Support for camera use & file management is streamlined via unified APIs.

Demo – View the Device Platform docs

ADDITIONAL RESOURCES

- Documentation Access
  - .NET MAUI docs home: https://learn.microsoft.com/dotnet/maui/
  - MAUI Toolkit docs: https://learn.microsoft.com/dotnet/communitytoolkit/maui/
- Tutorials and Guides
  - Get started in docs: https://learn.microsoft.com/dotnet/maui/get-started/first-app
  - .NET MAUI learning path: https://learn.microsoft.com/training/paths/build-apps-with-dotnet-maui/
- Community Groups
  - .NET MAUI Community Standup: https://www.youtube.com/watch?v=SeRGpBR3-3M&list=PLdo4fOcmZ0oX-sL7AFmygVw2A37Hbp8ZS
- Sample Projects
  - Access sample projects to practice and apply new skills in real-world scenarios: https://github.com/dotnet/maui-samples

FOUNDRY LOCAL MODELS IN WINDOWS, MACOS & ANDROID APPS

- Command line install on Windows & macOS
  - `winget install Microsoft.FoundryLocal`
  - `brew tap microsoft/foundrylocal && brew install foundrylocal`
- Install Foundry Local app on Android
- Use Microsoft.AI.Foundry.Local nuget package in .NET MAUI
- Use cached models for offline scenarios

Q&A AND THANK YOU!

- Questions about .NET MAUI & the Community Toolkits?
- General Cross-Platform Dev Questions?
- Feedback →
- Follow up later:
  - alashcraft@gmail.com
  - Find me on BlueSky and LinkedIn
- Thank you!