Windows Desktop and WinUI 3

For writing rich-client applications that run on Windows 10 and above, you can choose between the classic Windows Desktop APIs (Windows Forms and WPF) and WinUI 3. The Windows Desktop APIs are part of the .NET Desktop runtime, whereas WinUI 3 is part of the Windows App SDK (a separate download). The classic Windows Desktop APIs have existed since 2006 and enjoy terrific third party library support, as well as offering a wealth of answered questions on sites such as StackOverflow. WinUI 3 was released in 2022 and is intended for writing modern immersive applications that feature the latest Windows 10+ controls. It is a successor to the Universal Windows Platform (UWP)

ترجمه پاراگراف

Windows Desktop وWinUI 3
برای نوشتن اپلیکیشن‌های rich-clientکه روی ویندوز ۱۰ و بالاتر اجرا می‌شوند، می‌توانید بین Windows Desktop APIهای کلاسیک)یعنی Windows Forms و (WPF و WinUI 3یکی را انتخاب کنید.

Windows Desktop APIها بخشی از .NET Desktop Runtimeهستند، در حالی که WinUI 3 بخشی از Windows App SDKاست (که باید جداگانه دانلود شود).

Windows Desktop APIها از سال ۲۰۰۶ وجود داشته‌اند و از پشتیبانی بسیار خوب کتابخانه‌های شخص ثالث (third-party libraries) برخوردارند، و همچنین در سایت‌هایی مثل StackOverflow حجم زیادی سؤال و جواب برای آن‌ها وجود دارد.

WinUI 3در سال ۲۰۲۲ منتشر شد و برای نوشتن اپلیکیشن‌های مدرن و غنی طراحی شده است که شامل آخرین کنترل‌های ویندوز ۱۰ به بالا هستند.
WinUI 3جانشین (Universal Windows Platform) UWPمحسوب می‌شود.

🔷توضیحات تکمیلی و نکات ارائه

1. Windows Formsو) WPF کلاسیک:(
   - قدیمی‌تر (از ۲۰۰۶).
   - بسیار پایدار و امتحان‌شده.
   - کتابخانه‌های جانبی زیاد + جواب‌های بی‌شمار در انجمن‌ها.
   - مناسب برای اپلیکیشن‌های سازمانی یا پروژه‌هایی که پایداری مهم‌تر از ظاهر مدرن هست.
2.) WinUI 3جدید:(
   - عرضه در ۲۰۲۲.
   - بخشی از) Windows App SDK باید جدا دانلود شه.(
   - هدف: ساخت اپلیکیشن‌های مدرن، غنی و.immersive
   - ارائه‌ی آخرین کنترل‌ها وUI های مخصوص.+Windows 10
   - جانشین → UWP یعنی مسیر آینده‌ی مایکروسافت برای اپلیکیشن‌های ویندوز.
3. انتخاب بین این دو:
   - اگر پروژه‌ای داری که به پایداری و سازگاری نیاز داره.Forms/WPF →

- اگر پروژه‌ای داری که روی UIمدرن و استفاده از قابلیت‌های جدید ویندوز تمرکز داره → WinUI 3.

☞جمع‌بندی ارائه برای این بخش:
"روی ویندوز ۱۰ به بالا، برای ساخت اپ‌های دسکتاپ دو انتخاب داریمAPI :های کلاسیک Forms) و (WPFکه قدیمی ولی پایدار و پر از کتابخانه هستن، یا WinUI 3 که تازه و مدرن و جانشین UWP محسوب میشه. انتخاب بین این دو بستگی به این داره که دنبال ثبات باشیم یا دنبال UI مدرن".

MAUI

MAUI (Multi-platform App UI) is designed primarily for creating mobile apps for iOS and Android, although it can also be used for desktop apps that run on macOS and Windows via Mac Catalyst and WinUI 3. MAUI is an evolution of Xamarin and allows a single project to target multiple platforms

ترجمه پاراگراف

MAUI
(Multi-platform App UI) MAUIعمدتاً برای ساخت اپلیکیشن‌های موبایل برای iOSو Androidطراحی شده است، هرچند می‌تواند برای ساخت اپلیکیشن‌های دسکتاپ که روی macOSو Windowsاجرا می‌شوند (از طریق Mac Catalystو (WinUI 3نیز استفاده شود.

MAUIتکامل‌یافته‌ی Xamarinاست و این امکان را می‌دهد که با استفاده از یک پروژه‌ی واحد، چندین پلتفرم را هدف قرار دهید.

🔷توضیحات تکمیلی و نکات ارائه

1. هدف اصلی:MAUI
   - تمرکز روی توسعه‌ی Cross-Platform Mobile Apps (iOSو.(Android
   - پشتیبانی از دسکتاپ هم به عنوان مزیت اضافه.
2. ارتباط با:Xamarin
   - MAUIنسخه‌ی جدید و پیشرفته‌ی Xamarin.Forms محسوب میشه.
   - مایکروسافت MAUI رو جایگزین Xamarin کرده تا توسعه یکپارچه‌تر و مدرن‌تر باشه.
3. یک پروژه برای چند پلتفرم:
   - به جای داشتن چند پروژه جدا برایiOS ، Androidو → Desktop فقط یک پروژه داری.
   - مدیریت و نگهداری راحت‌تر میشه.
   - کدهای UI و logic رو میشه مشترک نوشت.
4. پلتفرم‌های پشتیبانی‌شده:
   - Mobile: iOS, Android
   -) Desktop: Windowsبا (WinUI 3 و) macOS با(Mac Catalyst
5. چرا مهمه؟

- یکپارچگی → برنامه‌نویس با C# و XAML می‌تونه هم موبایل بنویسه هم دسکتاپ.
- صرفه‌جویی در زمان و هزینه‌ی توسعه.

☞جمع‌بندی ارائه برای این بخش:
MAUI"تکامل‌یافته‌ی Xamarin هست و بهت اجازه میده با یک پروژه، اپلیکیشن رو برایiOS ، Androidو حتی دسکتاپ‌های Windows و macOS بسازی. این یعنی یک کدبیس برای چند پلتفرم، که توسعه و نگهداری رو خیلی ساده‌تر می‌کنه".

For cross-platform desktop applications, a third-party library called Avalonia offers an alternative to MAUI. Avalonia also runs on Linux and is architecturally simpler than MAUI (as it operates without the Catalyst/WinUI indirection layer). Ava lonia has an API similar to WPF, and it also offers a commer cial add-on called XPF that provides almost complete WPF compatibility

ترجمه پاراگراف

برای اپلیکیشن‌های دسکتاپ چندسکویی (cross-platform desktop applications) یک کتابخانه‌ی شخص ثالث به نام Avaloniaجایگزینی برای MAUIارائه می‌دهد.

Avaloniaروی Linuxهم اجرا می‌شود و از نظر معماری ساده‌تر از MAUI است (چون بدون لایه‌های واسط Catalyst/WinUIکار می‌کند.(

Avaloniaیک API شبیه به WPFدارد و همچنین یک افزودنی تجاری به نام XPFارائه می‌دهد که تقریباً سازگاری کامل با WPF را فراهم می‌کند.

🔻توضیحات تکمیلی و نکات ارائه

1. Avaloniaبه‌عنوان جایگزین:MAUI
   - تمرکز روی) Desktop Cross-Platformویندوز، مک، لینوکس.(
   - تفاوت اصلی MAUI :موبایل + دسکتاپ رو هدف گرفته؛ Avalonia بیشتر روی دسکتاپه.
2. مزیت‌ها نسبت به:MAUI
   - سادگی معماری :نیازی به واسطه‌های WinUI یا Catalyst نداره → مستقیم‌تر کار می‌کنه.
   - پشتیبانی از :Linux چیزی که MAUI به‌طور رسمی پوشش نمیده.
3. شباهت به:WPF
   - APIخیلی نزدیک به WPF → یادگیری راحت برای کسانی که تجربه‌ی WPF دارن.
   - کدهای WPF با کمترین تغییر قابل استفاده در Avalonia هستن.
4. XPF (Add-on تجاری:(
   - افزونه‌ای پولی که تقریباً سازگاری کامل با WPF میده.
   - این یعنی پروژه‌های WPF قدیمی می‌تونن خیلی راحت به Avalonia منتقل بشن.

☞جمع‌بندی ارائه برای این بخش:
Avalonia"یک گزینه‌ی قوی برای ساخت اپلیکیشن‌های دسکتاپ چندسکویی محسوب میشه. مزیت اصلیش پشتیبانی از لینوکس و سادگی معماریه. چون API مشابه WPF داره، مهاجرت برای توسعه‌دهندگان ویندوزی راحت‌تره، و با افزونه‌ی XPF میشه تقریباً سازگاری کامل با WPF داشت".

.NET Framework

.NET Framework is Microsoft's original Windows-only runtime for writing web and rich-client applications that run (only) on Windows desktop/server. No major new releases are planned, although Microsoft will continue to support and maintain the current 4.8 release due to the wealth of existing applications. With the .NET Framework, the CLR/BCL is integrated with the application layer. Applications written in .NET Framework can be recompiled under .NET 8, although they usually require some modification. Some features of .NET Frame work are not present in .NET 8 (and vice versa). .NET Framework is preinstalled with Windows and is automatically patched via Windows Update. When you target .NET Framework 4.8, you can use the features of C# 7.3 and earlier. (You can override this by specifying a newer language version in the project file—this unlocks all of the latest language features except for those that require support from a newer runtime.

ترجمه پاراگراف

.NET Framework
.NET Framework.، اولین Runtime اختصاصی ویندوز از طرف مایکروسافت است که برای نوشتن اپلیکیشن‌های وب و اپلیکیشن‌های rich-client طراحی شده و فقط روی دسکتاپ/سرور ویندوز اجرا می‌شود. هیچ نسخه‌ی بزرگ جدیدی برای آن برنامه‌ریزی نشده است، هرچند مایکروسافت به دلیل وجود حجم زیادی از اپلیکیشن‌های قدیمی همچنان از نسخه‌ی فعلی یعنی 4.8پشتیبانی و نگهداری می‌کند.

درNET Framework. ، CLR/BCLبا لایه‌ی اپلیکیشن یکپارچه است. اپلیکیشن‌هایی که با NET. Frameworkنوشته شده‌اند را می‌توان تحت NET 8. دوباره کامپایل کرد، هرچند معمولاً نیاز به اصلاحاتی دارند. برخی قابلیت‌های NET Framework. در NET 8. وجود ندارند (و برعکس).

NET Framework.به‌طور پیش‌فرض همراه ویندوز نصب می‌شود و به‌صورت خودکار از طریق Windows Updateوصله (patch) می‌شود. وقتی پروژه‌ای را روی NET Framework 4.8. هدف قرار دهید، می‌توانید از قابلیت‌های زبان C# نسخه ۷,۳ و قبل‌تر استفاده کنید) .می‌توانید این محدودیت را با مشخص کردن نسخه‌ی جدیدتر زبان در فایل پروژه کنار بزنید—این کار همه‌ی قابلیت‌های جدید زبان را فعال می‌کند به جز آن‌هایی که به پشتیبانی یک Runtime جدیدتر نیاز دارند(.

🔻توضیحات تکمیلی و نکات ارائه

1. ماهیت:NET Framework.
   - اولین Runtime مایکروسافت، فقط ویندوزی.

- هدف: اپلیکیشن‌های وب (ASP.NET) و rich-client (WinForms, WPF).
- الان به عنوان فناوری) Legacyقدیمی اما پایدار) محسوب میشه.

2. وضعیت توسعه:
   - روی نسخه‌ی 4.8متوقف شده.
   - آپدیت جدید بزرگی براش نخواهد آمد.
   - ولی همچنان پشتیبانی بلندمدت و وصله‌های امنیتی دریافت می‌کنه.
3. یکپارچگی:(Integration)
   - CLR + BCL + Application Layer → همه در یک پکیج.
   - تفاوت با .NET 8. که ساختارش لایه‌بندی و ماژولارتره.
4. سازگاری با:.NET 8
   - کدهای .NET Framework → اغلب نیاز به تغییر برای اجرای صحیح روی.NET 8.
   - دلیلAPI :های متفاوت و حذف/اضافه شدن امکانات.
5. مزیت بزرگ:
   - Preinstalledروی ویندوز.
   - همیشه از طریق Windows Updateبه‌روزرسانی میشه → کاربر هیچ کاری لازم نداره.
6. محدودیت زبان:#C
   - به طور پیش‌فرض فقط تا .C# 7.3
   - میشه نسخه‌ی جدیدتر زبان رو در فایل پروژه مشخص کرد، ولی ویژگی‌هایی که نیاز به Runtimeجدیدتر دارن کار نمی‌کنن.

☞جمع‌بندی ارائه برای این بخش:
.NET Framework."اولین Runtime ویندوزی مایکروسافته که الان روی نسخه‌ی ٤,٨ ثابت شده و دیگه توسعه‌ی بزرگی براش در کار نیست. هنوز به خاطر پروژه‌های قدیمی پشتیبانی میشه و همراه ویندوز نصبه. اما محدودیتش اینه که فقط ویندوز رو پوشش میده و نهایتاً تا C# 7.3 رو به طور پیش‌فرض ساپورت می‌کنه. برای مهاجرت به .NET 8. هم معمولاً نیاز به تغییرات در کد هست".

The word ".NET" has long been used as an umbrella term for any technology that includes the word ".NET" (.NET Frame work, .NET Core, .NET Standard, and so on). This means that Microsoft's renaming of .NET Core to .NET has created an unfortunate ambiguity. In this book, we'll refer to the new .NET as .NET 5+ when an ambiguity arises. And to refer to .NET Core and its successors, we'll use the phrase ".NET Core and .NET 5+." To add to the confusion, .NET (5+) is a framework, yet it's very different from the .NET Framework. Hence, we'll use the term runtime in preference to framework, where possible.

ترجمه پاراگراف

واژه‌ی ".NET"مدت‌ها به‌عنوان یک اصطلاح کلی (umbrella term) برای هر فناوری‌ای که شامل ".NET" باشد استفاده شده است) مثلFramework .NET. ، .NET Core.، .NET Standard.و غیره.(

این موضوع باعث شده تغییر نام .NET Core.به .NET.از طرف مایکروسافت، نوعی ابهام ناخوشایند ایجاد کند. در این کتاب، هر جا که ابهام پیش بیاید، به .NET. جدید با عبارت .NET 5+.اشاره خواهیم کرد. و برای اشاره به .NET Core. و جانشینان آن، از عبارت ".NET Core and .NET 5+"استفاده می‌کنیم.

برای پیچیده‌تر شدن موضوع، (+5) NET.یک Frameworkاست، اما تفاوت زیادی با .NET Framework دارد. بنابراین ما ترجیح می‌دهیم تا جای ممکن از اصطلاح Runtimeبه جای Frameworkاستفاده کنیم.

🔻توضیحات تکمیلی و نکات ارائه

1. .NET.به‌عنوان یک اصطلاح کلی:
   - از اول ".NET" مثل یک برچسب کلی روی محصولات مختلف مایکروسافت بوده:
     - .NET Framework
     - .NET Core
     - .NET Standard
     - .NET 5, 6, 7, 8
2. ابهام نامگذاری:
   - وقتی مایکروسافت ".NET Core" رو به ".NET" تغییر داد) از .NET 5 به بعد(، ابهام ایجاد شد:
     - آیا منظور .NET Framework هست؟
     - یا منظور همون Core/5+ هست؟
3. روش کتاب برای رفع ابهام:
   - = ".NET 5+"اشاره به نسخه‌های جدید (از ۵ به بعد).
   - = ".NET 5+ .NET Core and"وقتی می‌خواهیم به کل این شاخه اشاره کنیم.
4. چرا Runtime به جایFramework ؟
   - چون ".NET 5+" یک Framework هست، اما شباهتی به ".NET Framework" قدیمی نداره.
   - برای جلوگیری از سردرگمی، بهتره بگیم → Runtimeدقیق‌تر و شفاف‌تر.

👉جمع‌بندی ارائه برای این بخش:
"اسم .NET. همیشه به شکل کلی برای محصولات مختلف استفاده شده. وقتی مایکروسافت Core رو حذف کرد و اسمش رو گذاشتNET. ، یه ابهام بزرگ درست شد. برای همین ما تو این کتاب می‌گیم .NET 5+تا منظورمون واضح باشه. و چون .NET 5+ خیلی با .NET Framework. فرق داره، ترجیح می‌دیم بهش Runtime بگیم تا".Framework