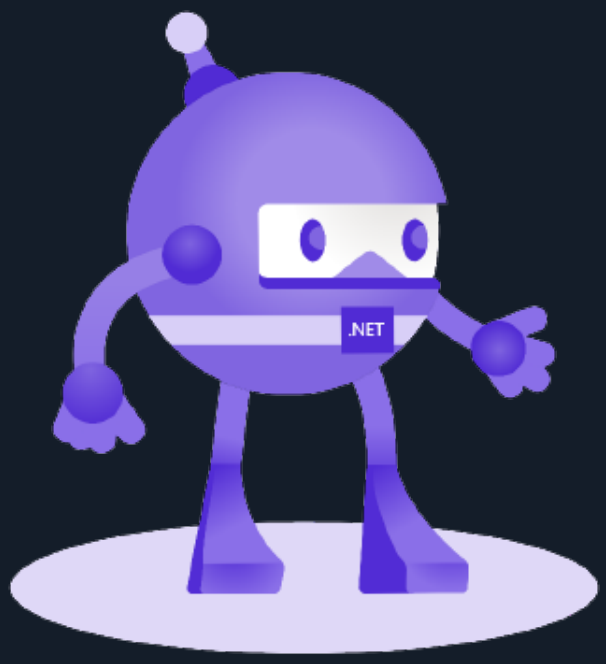
Services
Quality Redefined

GROW YOUR BUSINESS WITH .NET MAUI

TABLE OF CONTENT



ALL YOU NEED TO KNOW ABOUT .NET MAUI

Net Multi-program APP UI (MAUI) is a framework for developing modern, natively compiled Android, iOS, Windows, macOS, multi-platform apps using XAML and C# in a single codebase.

With the help of NET MAUI, you can develop a certain native application from a single codebase for iOS, Android, Windows, macOS, and Tizen supported by Samsung.

If you know to code and have used Xamarin Forms before, then you won't face many issues while using MAUI as it is easy to use. Instead of having different mobile, desktop, and OS projects, you can keep all your code in a single project. NET MAUI also offers to host Blazor in MAUI, with additional embedded web view control to make the razor component work on your targeted device natively. The single project and decoupled UI allow you to focus on one application rather than getting trapped in the need for various platforms.

A COMPLETE OVERVIEW OF .NET MAUI

.NET MAUI is considered one of the latest editions of Microsoft's .NET ecosystem. This blog is all you need if you are new to this industry and .NET MAUI. So, here you will get to know everything about this industry, mainly what .NET MAUI is and the reason behind its growth. Simply put, .NET MAUI is a UI framework that helps develop UI apps using .NET, C#, and XAML. This UI framework can be used to develop apps for macOS, Web, Windows, etc. It simply means that you can create a single project and later deploy it on various platforms so that it works on all the platforms, including Mac, Laptop, smartphone, etc.

Whenever the topic of the .NET ecosystem comes to mind, one thing that wonders me is Code Re-use. However, knowing how much an individual can do in just one .NET codebase is amazing.

We all have noticed many people switch their daily devices to stay productive. It is becoming very important that the apps that are in work, regardless of the devices our clients are in front of. This feature makes the web powerful, and there are many advantages to writing your application for the browser. But you need to keep one thing in mind when we do this, we usually get through our native platform performance and control offered to everyone by the OS vendors, making this app very uncommon to work at and look at.

PART - 1

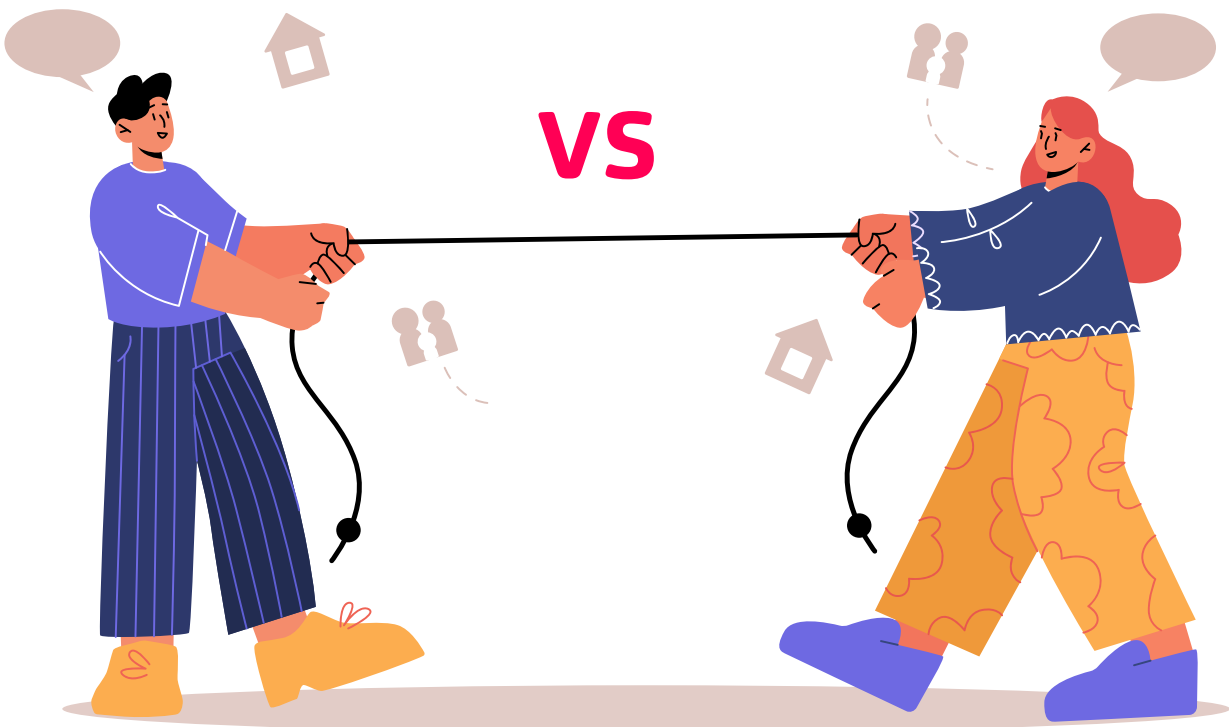
.NET 5 AND 6

.NET 5 AND 6

Microsoft's main aim with .NET 5 was to start unifying all the scattered technologies we know and love within. NET. Therefore, the main function was to bring all the .NET Core, Xamarin, and Mono into one unified ecosystem. With the help of.NET6, the system will keep moving forward with the development of .NET MAUI.

MAUI VS. XAMARIN

If you collect knowledge about NET MAUI, you must know that it is the next step or a better version of Xamarin. Forms. Some of the advantages of MAUI include.



ONE PLATFORM FOR VARIOUS PLATFORMS

With MAUI, you can manage various platforms in a single project. Simply put, there won't be multiple projects for every platform. If you are not new to this industry, then you might know that Xamarin uses different projects for different platforms. In MAUI, all the shared resource files get stored within a single project. A single application manifests the app that specifies the id, version, and title. A single cross-platform helps you to develop the entry point faster.

GRAPHICS APIS

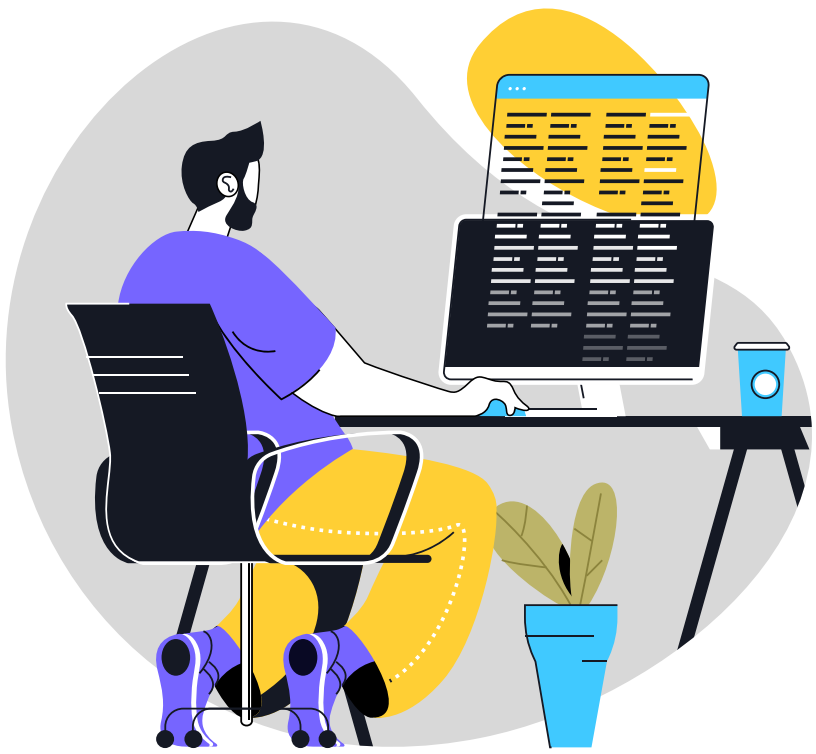
We all know that Xamarin has no direct API that can handle any drawing requirement. You will need to do it using renderers on the native side. On the other hand, NET MAUI doesn't promote native drawing. In NET MAUI, the cross-platfor graphics functionality offers a canvas for painting and drawing shapes.

.NET 6 SUPPORT

With MAUI, you can manage various platforms in a single project. Simply put, there won't be multiple projects for every platform. If you are not new to this industry, then you might know that Xamarin uses different projects for different platforms. In MAUI, all the shared resource files get stored within a single project. A single application manifests the app that specifies the id, version, and title. A single cross-platform helps you to develop the entry point faster.

SLIM RENDERERS

Almost everyone knows that custom renderers are used in Xamarin App. Although no custom renderers are available, an individual can still use slim renderers. Using that will streamline the development experience and will make your app lightweight.



ONE PLATFORM TO MANAGE ALL THE ASSETS

You can maintain all your resources like images, splash screen, CSS, fonts, or natural assets from a commonplace. Whereas on the other hand Xamarin. Forms use multiple projects for those.

MULTITARGET TO ARRANGE PLATFORM-SPECIFIC CODE

.NET MAUI projects usually consist of a platform folder, where every child folder represents a different platform. So now, you can easily maintain different platform specific codes.

HELP FOR THE (MVU) MODEL VIEW UPDATE PATTERN

As we all are aware that Xamarin Forms offer help to both Reactive UI patterns and Model view patterns. However, .NET MAUI offers help to Blazor development pattern and model view update. In addition, this pattern offers a unified method to create a cross-platform native front end through a single code base.

HELP FOR BLAZOR

We all know that Blazor is one of the best adaptive models (programming) for creating a web application. .NET will help and extend Blazor's target market to add on native desktop applications that rely on web-based rendering. It is said that the Blazor desktop is going to be structured exactly like the Electron works

UNIFICATION OF LIBRARIES

.NET MAUI comes with a wide option of unification of critical libraries. It offers several additional benefits by attaching the Xamarin essential library with the .NET MAUI so that you can easily access the device capabilities like device photos, sensors, and some other services that are used regularly like secure storage and authentication.

COMPLETE HELP TO THE HOT RELOADS

You can manage and modify the source code and XAML while the application is turned on. You can also see the result of your changes without rebuilding stopping, or starting your application.

PART - 2

WHY CHOOSE .NET MAUI?

ACCESSIBILITY FIRST

One of the main benefits of using the UI is the inherited accessibility support that it offers to its users. It helps make the process easy to create high apps with high accessibility with semantic services.

- Properties include hint, heading level, and description.
- Screen reader
- Focus
- Automatic properties

YOU CAN EASILY CREATE A NEW PLATFORM

To protect you from hitting a wall .NET MAUI is architected. Moreover, Android draws a line below every text filled; most of the time, developers want to remove that underline. With the help of .NET MAUI, you can customize all the entries in your project in just a few clicks with the line of coding.

FILE SCOPED NAMESPACE AND GLOBAL USING STATEMENT

We all know that .NET MAUI uses the C# 10 features available in .NET 6, including the file scoped namespace feature and global using statement. These features are great for eliminating clutter from your file.

ONE PROJECT FOR VARIOUS PLATFORMS

In the new .NET MAUI projects, platforms are usually placed in a folder that will give special attention to your application. Within the resource folder of your project, you can manage your image, app icon, and splash screen styling in one place. NET MAUI will do the necessary to reduce them for the platform's unique needs.

USE BLAZERS FOR MOBILE AND DESKTOP

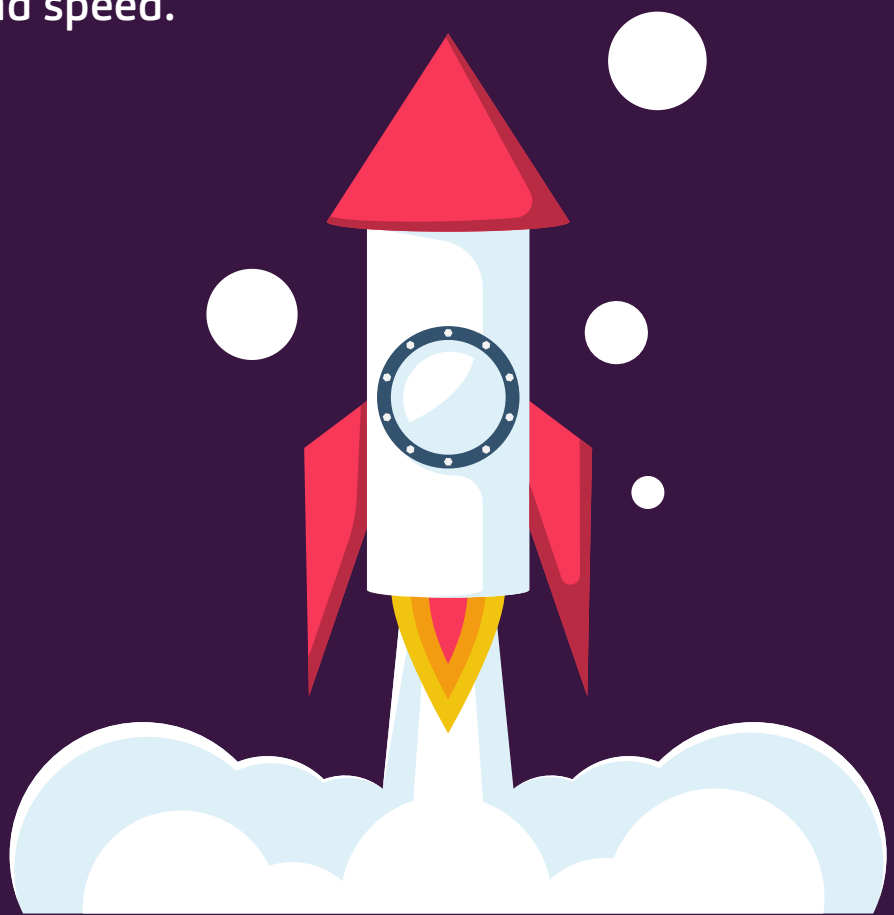
NET MAUI is also a great choice for web developers who are thinking of running web code in a few native client apps. Anyone can use their skills in web development to build a cross-platform application for mobile, web, and desktop. It will also add your Blazor components to the device. In the entire process, no web assembly is needed. Your blazer component will get compiled and executed in the .NET process. Moreover, they aren't limited to the platform and can they add any platform features like Bluetooth, sensor, notification, etc.

SPEED OPTIMISM

Users can experience a reduced app size and a good improvement in performance after the .NET 6 transition. The best part is that the team eliminated some parts of the Java<-> C# call cycle. After this elimination, no further creation of C# objects from Java is seen. The sample application of the .NET podcast is also created with .NET MAUI.

- It started with 1299ms (about 1 and a half seconds)
- It came up till 814ms.
- The report shows that it has shown a 37% improvement.

You will find the native application on the left-hand side and the .NET MAUI on the right side. Whereas the native app takes half a second to load, the .NET MAUI adds 1 second in JIT complication. Users may experience a 2x faster loading time when the AOT complication gets enabled. The higher the file size, the faster your system will load only when the AOT is enabled. In simple words, choosing AOT will help you maintain a balance between size and speed.



MIGRATING FROM XAMARIN APP

The manual way

If you plan to migrate from a Xamarin app, there is no need to rewrite the old Xamarin. Forms apps to shift to the .NET MAUI. You can still use the single-project features without attaching all your Xamarin. Forms project into one platform.

- Transfer the project from .NET Framework to .NET SDK style.
- Update all the namespaces.
- Update all the incompatible NuGet packages.
- Find out the breaking API changes.

Upgrade assistant

This tool is very famous and is developed by Microsoft. This tool was not for .NET MAUI, but an individual can use it for migrating their Xamarin. Forms projects.

GET STARTED .NET MAUI

If you want .NET MAUI to start on your windows, then update or install the visual studio preview and make sure you choose workload as “.NET Multi-platform App UI development."

PART - 3

WHAT DOES .NET MAUI OFFER?

WHAT DOES .NET MAUI OFFER?

NET MAUI offers a wide collection of controls that individuals can use to initiate action, display data, display collection, pick data, and many more. Moreover, some of the collection of that control includes:

- A wide layout engine for fulfilling the purpose of designing pages.
- Numerous page types for creating rich navigation types that include drawers.
- An indirect help for data binding for a more maintainable and elegant development pattern.
- It permits you to customize the handlers to enhance the procedure in which the UI is presented.
- Cross-platform APIs for getting access to native device features. These APIs allow the apps to access features like the accelerometer, battery, GPS, and network status. You can browse the internet to get more cross-platform API features.
- A separate single project system that uses the feature of multi-targeting to target iOS, Android, macOS, and windows.
- We all know about .NET hot reload; it allows you to modify both managed source data and XAML, which the app is in process. It can also go through the modification without modifying the app.

CROSS-PLATFORM APIS

We all know that .NET MAUI offers a cross-platform API for all native device features. A few examples of functionality offered by .NET MAUI for accessing features include:

- Access to various sensors, including compass, gyroscope, accelerometer, etc.
- Permission to check the device's network connectivity and make changes if required.
- Copy and paste various images and texts to the system clipboard between apps.
- Choose multiple or single files from the device.
- Secured data stored as value/key pairs.
- Usage of text-to-speech engines to understand the text from the device.

HOW IS .NET MAUI DIFFERENT FROM XAMARIN?

We all are now familiar with both the terms .NET MAUI and Xamarin. forms, especially when you plan to develop cross-platform apps. Before we start, all of us need to know what Xamarin. Forms are.

Forms: Overview

Xamarin. Forms are also used to create a native cross-platform app. In short, it is a plain layer of abstraction that allows interaction of the given code with the underlying platform code of iOS, Windows, and Android.

Using this, cross-platform app development can save a lot of time during the development process by penning down all their business process logic in one language. As per the notes, 90% of the codes needed for the development of an app is given across the platforms, permitting you to achieve native performance.

Common feature

We all know that .NET MAUI is a better version of Xamarin. forms that's why most of their features are the same. You can get everything in both versions. .NET MAUI and Xamarin have the same layouts, controls, shells, templates, gestures, and cross-platform APIs. One basic difference is that you will find all of these with a different name in .NET MAUI.

DIFFERENCE

Platform architecture

There are not as such many major differences in the platform architecture. You may find that .NET MAUI is broken down with .NET 6.0 so they can be a part of the One.NET journey.

Platform structure

Using Xamarin. Developer forms may get irritated working with various projects targeting different platforms simultaneously, keeping fonts, images, and platform-related codes, adding various dependencies, and resolving them if they are considered NuGet packages. Getting involved in all these issues, developers are now considering moving to .NET MAUI.

We all know that .NET MAUI is a very simple project, and mainly its entire system works on a multitarget-based structure. A .NET MAUI app consists of a platform folder, having many subfolders representing various platforms such as iOS Mac-catalyst, windows, and androids to target platform-specific code that helps the app to start on the platform.

Supported version and platform

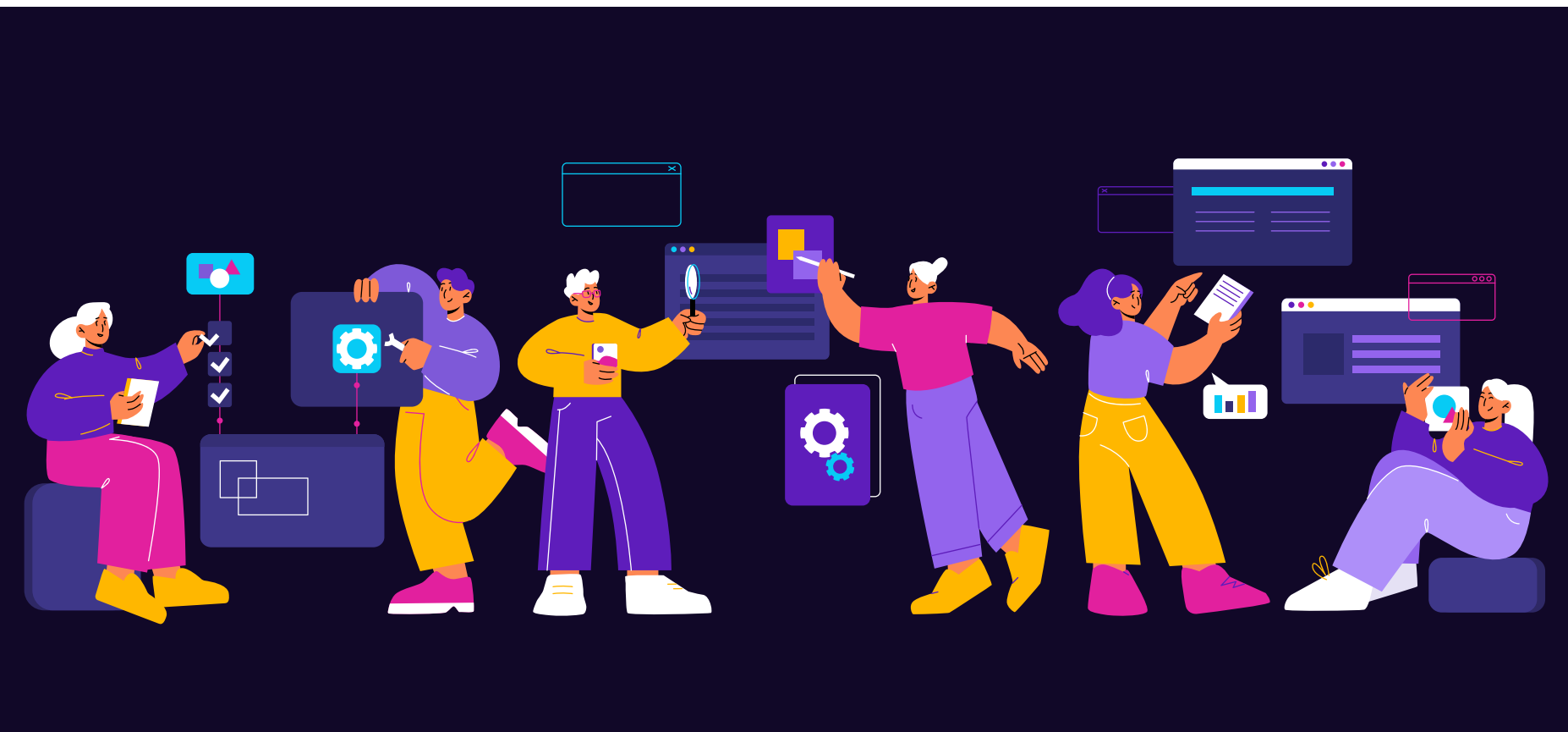
The major difference in platform support between Xamarin and .NET MAUI is their support system. According to the report, Xamarin supports UWP, and .NET MAUI supports WinUI.

Configuring resources and services

In this field, .NET MAUI apps are different from others; they are bootstrapped with the help of .NET Generic Host. So, if you are thinking of adding any services, fonts, or third-party libraries, you can do it directly from a single place.

Accessibilities

Especially in Xamarin, users need to use the automatic properties and the APIs to offer proper accessibility to readers of texts.



PART - 4

.NET CLI
.NET 6

.NET CLI

One main thing that everyone needs to know is .NET MAUI works under .NET CLI. The CLI is a cross-platform toolchain for building, developing, publishing, and running .NET apps. This tool offers a seamless run and builds experience.

HANDLER ARCHITECTURE AND RENDERER

We all know that Xamarin's renderer is used to make the controls. When the developer is interested in customizing a control's UI, they must choose a custom renderer. This renderer is usually very heavy in terms of app size and performance.

On the other hand, .NET MAUI takes the help of handler architecture that is very loosely tied with the native assembly. Using a native platform, the user may experience a lightweight app with great performance.

.NET 6

.NET MAUI is later broken into .NET 6, whereas Xamarin. Forms are themselves a .NET framework. Due to the breakdown of .NET 6, users can use the given .NET 6 and C#10 features in .NET MAUI:

- Nullable disable/enable and their reference types
- Record structs
- Global using directives
- Source generators

RESOURCE MAINTENANCE

Some report claims that .NET MAUI has taken over Xamarin in terms of various resources, especially in terms of photos. You don't have to maintain a set of photos for the device-specific or platform needs. A single photo is enough to meet all the device and platform requirements.

PART - 5

.NET MAUI LIST VIEW

HOW CAN YOU DEVELOP AN APP USING THE .NET MAUI LIST VIEW

Some report claims that .NET MAUI has taken over Xamarin in terms of various resources, especially in terms of photos. You don’t have to maintain a set of photos for the device-specific or platform needs. A single photo is enough to meet all the device and platform requirements.

STEP 1: DEVELOP THE LIST VIEW CONTROL

In the first stage, you must create a new .NET MAUI app in the visual studio. .NET MAUI components (syncfusion) are available on the NuGet gallery. To attach the SfListView to your project, you need to open the NuGet manager in visual studio and then choose Syncfusion.MAUI.ListView and then download it. In the future, you can import the control namespace in your C# and XAML code.

STEP 2: REGISTER THE HANDLER.

If you want to use the .NET MAUI List view control in your app, you need to put the SfListView handler in download.

STEP 3: BIND DATA

The .NET MAUI List view is nothing but just a data-bound control. Users need to create the entire data model to bind it.

A FEW NEW FEATURES THAT WILL BLOW UP EVERYONE'S MIND

We all know that .NET MAUI is a framework for .NET Core and ASP.NET Core MVC mainly used in cross-platform development for complex web apps.

Microsoft mainly created .NET MAUI to take care of all the things developers had to do themselves, such as setting up routing logic, handling errors, managing HTTP requests, etc. So that they can put them all focus on developing an app that offers something useful to their clients.

PART - 6

MICROSOFT NEW VERSION

MICROSOFT HAS CHANGED A LOT OF CONTROLS IN THE NEW VERSION

BOX VIEW

It is nothing but just a basic rectangle with a given height, color, and width. You can use it for basic graphics, decorating, and touching interaction.

IMAGE BUTTON

This tool has a special ability to combine the image view and button to create a button with a photo as its content. However, to put the command program to work, you need to do a certain activity, and the user needs to push the image button using their mouse.

SHADOW, BORDERS, AND CORNERS

Users can use Microsoft. MAUI. Graphics library provides a +uniform UI drawing API focused on the graphics engine, allowing users to quickly add corner rendering, borders, and great shadows.

THE TAKEAWAY

So today, in this article, we will discuss the MAUI and how to develop a sample app. You can browse the internet if you want to gather knowledge about Xamarin. Forms. Therefore, we have added all the important elements in this article, and you can go through it to gather relevant information. You can also check the latest version of the .NET MAUI. Thus, this UI framework is basically for developers who are interested in:

- Work on the cross-platform app in C# and XAML.
- Share the entire UI design and layout across the platforms.
- Share tests, code, and business logic across the platform.

Contact Us

USA/CA - TollFree

+1 (888) 721-3517

Email us

hello@qservicesit.com

Website

www.qservicesit.com