

November 9-11, 2021
www.dotnetconf.net

.NET Conf

Discover the world of .NET

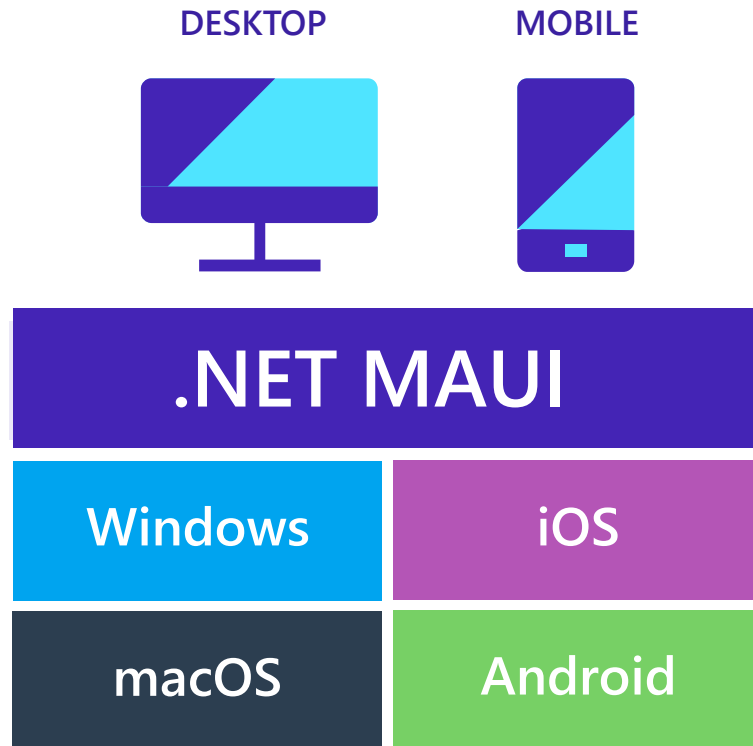


Drawn controls in .NET MAUI

Javier Suárez Ruiz
@jsuarezruiz



.NET MAUI Summary



Compatibility

- **Minimal** API Changes – All add developer value (perf, dev loop, etc.)
- **Migration support** for Xamarin & Xamarin.Forms
- Existing Xamarin.Forms app support for **one-year** window after release of .NET MAUI

SDK-style projects

Single-headed project

CLI support

.NET 6 BCL

Multi-paradigm

- XAML markup & C# UI supported

File | New

- Multi-platform App UI (.NET)

CLI

- `dotnet install maui`
- `dotnet new maui`

Namespaces

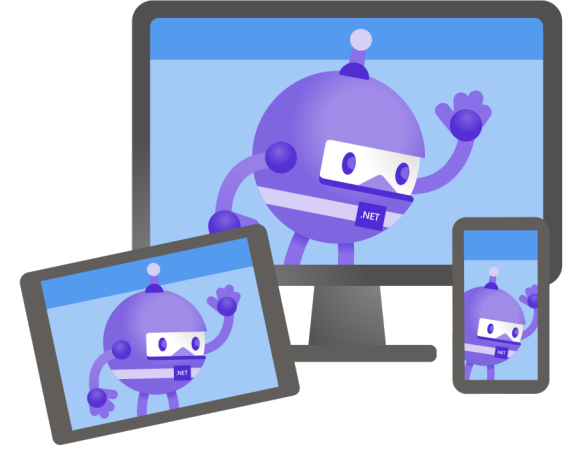
- `Microsoft.Maui` (previously `Xamarin.Forms`)
- `Microsoft.Device` (previously `Xamarin.Essentials`)



Experimental drawn controls



Feedback received

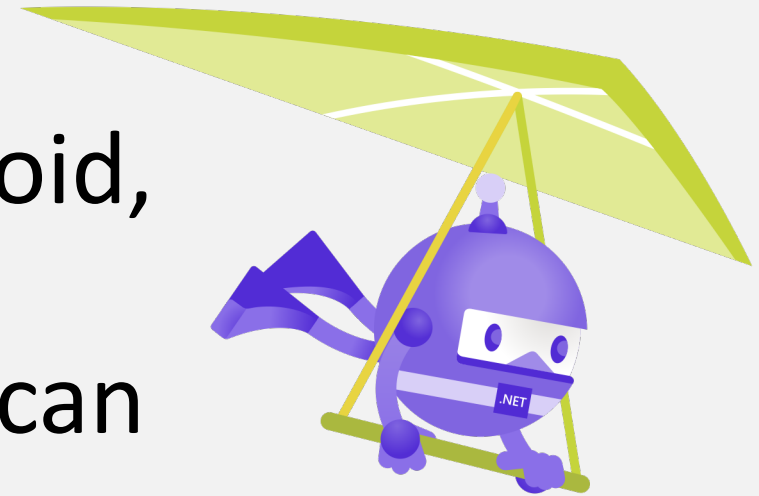
- Consistent look and behavior.
- Easy way to extend and customize.
- The Visual Material implementation is consistent, but rigid in its prescriptive design which doesn't match some customer designs.



<pre><Button Text="Button" HorizontalOptions="Center" VerticalOptions="CenterAndExpand" /></pre>	
Android	iOS
	

Microsoft.Maui.Graphics

Microsoft.Maui.Graphics is a **cross-platform graphics library** for iOS, Android, Windows, macOS, Tizen and Linux completely in C#. With this library you can use a common API to target multiple abstractions allowing you to share your drawing code between platforms, or mix and match graphics implementations within a singular application.



Microsoft.Maui.Graphics: Supported platforms

Platform	Supported Abstractions
Xamarin.iOS	CoreGraphics & SkiaSharp
Xamarin.Android	Android.Graphics & SkiaSharp
Xamarin.Mac	CoreGraphics & SkiaSharp
WPF	SharpDX, SkiaSharp, Xaml & GDI
UWP	SharpDX, Win2D, Xaml, SkiaSharp
WinForms	SharpDX, SkiaSharp & GDI
Tizen	SkiaSharp
Linux	SkiaSharp
Xamarin.Forms	Dependent on native platform support (noted above)

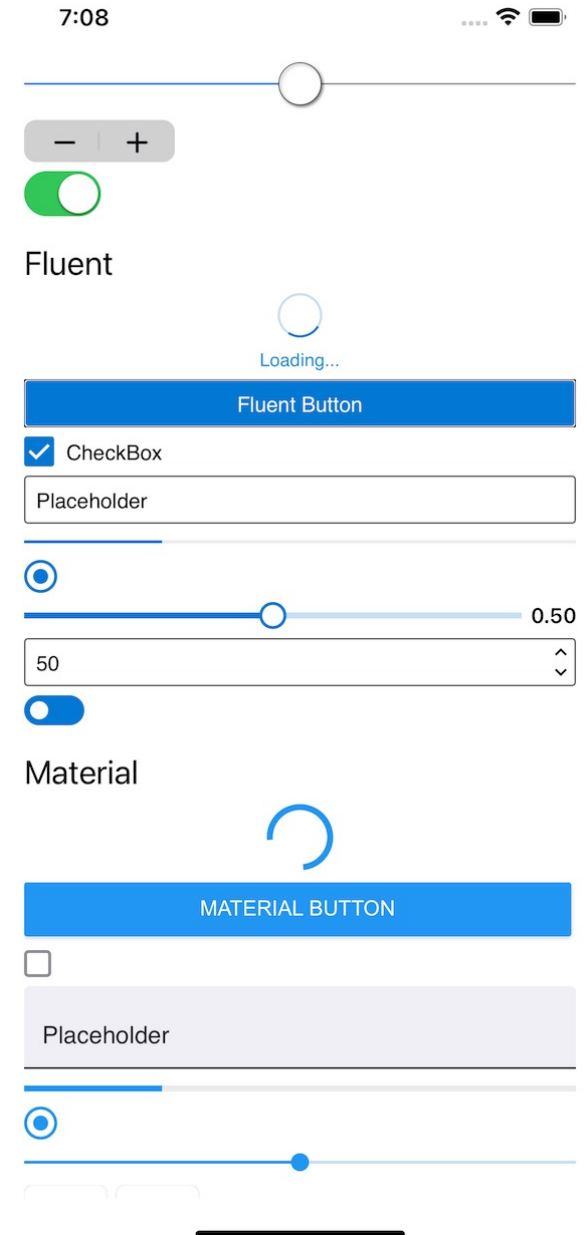
Demo

Microsoft.Maui.Graphics

Microsoft.Maui.Graphics.Controls

Microsoft.Maui.Graphics.Controls (GraphicsControls) is a library that offers the controls available in Xamarin.Forms Visual allowing to choose between **Cupertino**, **Fluent** and **Material** but, instead of creating native controls, are **drawn controls**.

This project is a **.NET experiment** related to .NET MAUI with different tests to validate the interest and feasibility of drawn controls but is not covered by Microsoft support (yet).



Microsoft.Maui.Graphics.Controls

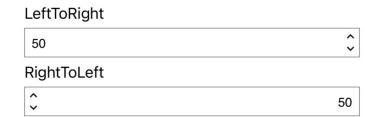
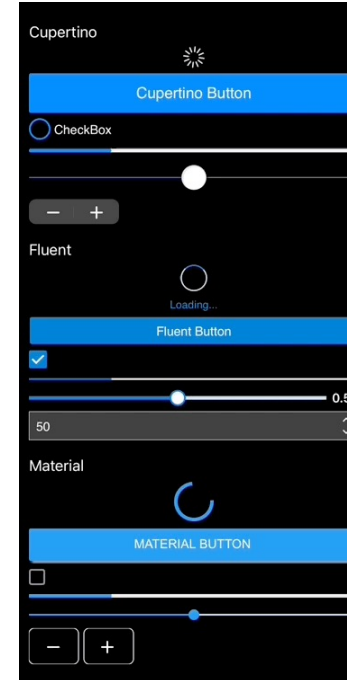
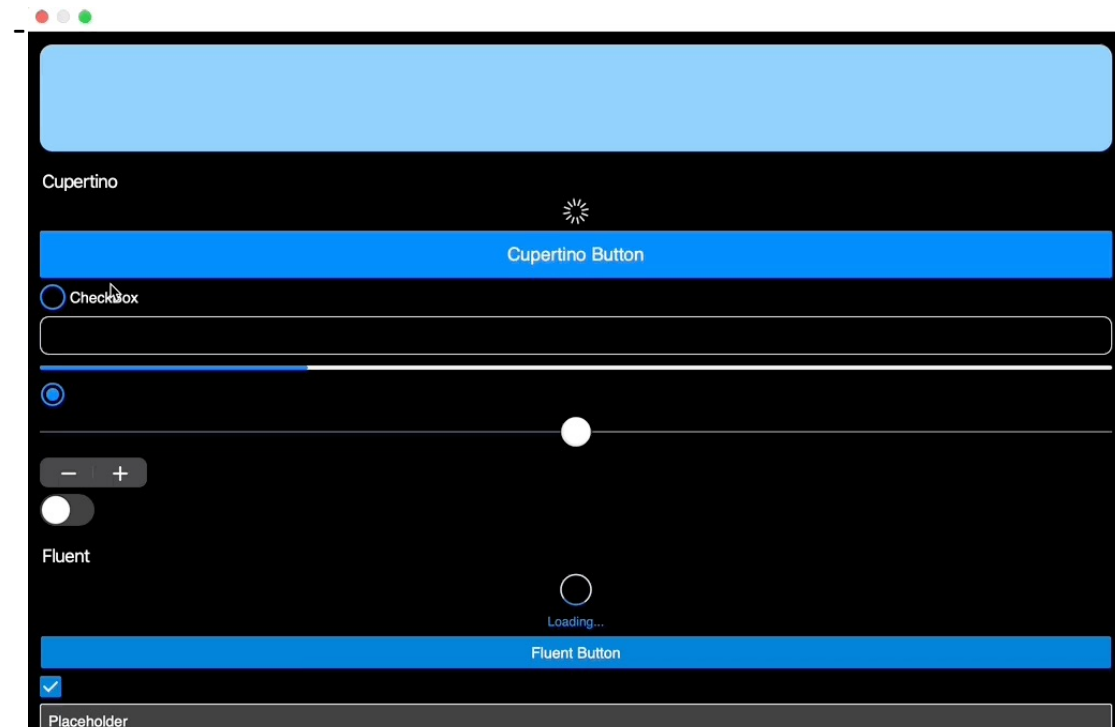
Available controls:

- **ActivityIndicator** (Cupertino, Fluent, Material)
- **Button** (Cupertino, Fluent, Material)
- **CheckBox** (Cupertino, Fluent, Material)
- **DatePicker** (Cupertino, Fluent, Material)
- **Editor** (Cupertino, Fluent, Material)
- **Entry** (Cupertino, Fluent, Material)
- **ProgressBar** (Cupertino, Fluent, Material)
- **RadioButton** (Cupertino, Fluent, Material)
- **Slider** (Cupertino, Fluent, Material)
- **Stepper** (Cupertino, Fluent, Material)
- **Switch** (Cupertino, Fluent, Material)
- **TimePicker** (Cupertino, Fluent, Material)

Microsoft.Maui.Graphics.Controls

Features:

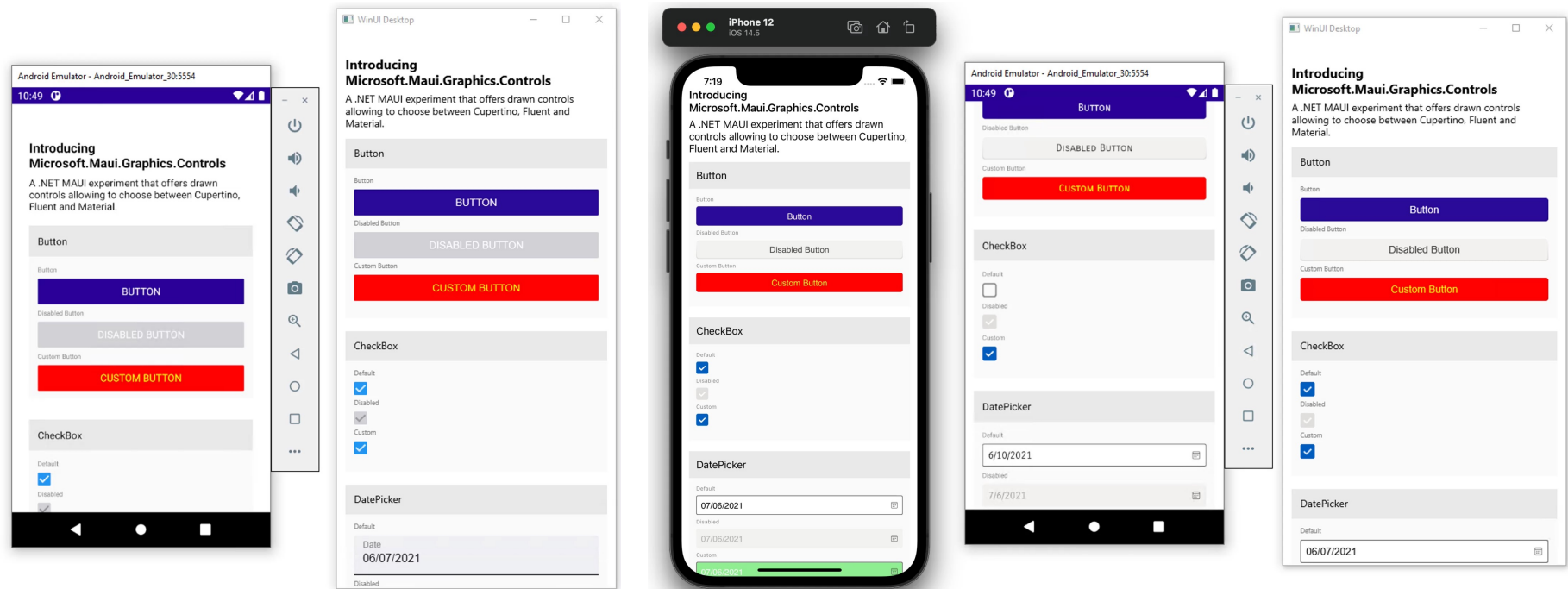
- Pixel perfect drawn and same behavior in all the platforms.
- RTL support.
- Accessibility support.
- Themes support.



Microsoft.Maui.Graphics.Controls

Features:

- Cupertino, Fluent and Material design.



Material

Fluent

Demo

Microsoft.Maui.Graphics.Controls

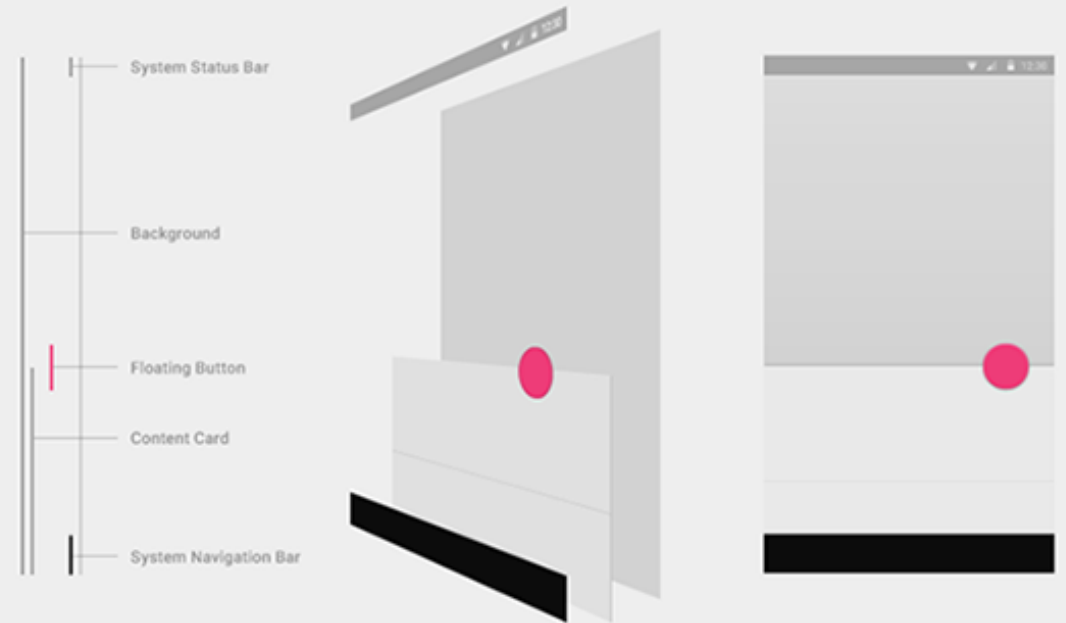
Customize drawn controls



Extend graphics controls

All drawn controls are created by **layers**. It is possible to access each layer, in addition to customizing a specific layer or adding/removing layers.

As in .NET MAUI, drawn controls use Handlers and a **DrawMapper** with information about each layer.



Extend a drawn control

The easiest way to extend an existing drawn control is to simply override its **Draw method** to draw any extra content.

```
public class CustomSliderHandler : SliderHandler
{
    protected override ISliderDrawable CreateDrawable() => new MaterialSliderDrawable();

    public override void Draw(ICanvas canvas, RectangleF dirtyRect)
    {
        base.Draw(canvas, dirtyRect);

        // Your own content
    }
}
```


Extend a drawn control

Customize how to draw a **specific layer**.

```
DrawMapper[nameof(ISliderDrawable.DrawThumb)] = (canvas, rect, drawable,
slider) =>
{

}
}
```

Also can override all the parts from a control.

Demo

Customize drawn controls

Summary

- **Microsoft.Maui.Graphics.Controls** is an **experimental** library that offers the controls available in Xamarin.Forms Visual allowing to choose between **Cupertino, Fluent and Material** but, instead of creating native controls, are **drawn controls**.
- Already implemented Visual controls with theme, RTL and accessibility support.
- Could be part of .NET MAUI in the future, at the moment, we collect feedback.
- Already available NuGet package: <https://aka.ms/maui-graphics-controls-nightly/index.json>

Questions?



Thanks for joining!

