# .NET MAUI Project, Item Templates and Code Snippets

This VS extension is loaded with Project, Item Templates and Code Snippets for working with .NET MAUI in Visual Studio 2022.

This has Project Template for the following:

- .NET MAUI App – All All-in-One App Project Template. For more details check out this blog post
- .NET MAUI App (C#)
- .NET MAUI Class Library
- Shared Class Library (Xamarin.Forms and .NET MAUI)

And reg. Item templates (find them in the section titled **MAUI**):

**ContentPage**, in both XAML and C#, and has been named as:

- Content Page (.NET MAUI)
- Content Page (C#) (.NET MAUI)

**ContentView**, in both XAML and C#, and has been named as:

- Content View (.NET MAUI)
- Content View (C#) (.NET MAUI)

**Shell**, a page for defining app visual hierarchy along with navigation.

**ResourceDictionary**, a page for managing resources, in both XAML and XAML only (as its the C# code-behind is used rarely).

**.NET MAUI Custom View and Handler**:

- An item template for creating a Custom View and its associated Handler definitions
- Available in both Regular definition with Handler source files in corresponding Platforms folder or Defined with Conditional Definition in a single folder
  - *For conditional compilation, ensure it is enabled for this particular item template to work properly. An additional option provided during project creation*

## Has XAML Code Snippets for:

In the XAML page, type the short name and hit the Tab key twice to insert the snippet.

Snippets mentioned in bold-face also works as a **SurroundWith** snippet too (from **Xaml** section).

In the Output Format column, text highlighted in different colors infer the following:

- Yellow color are placeholders where user can modify the values
- Green color are derived values, can't be modified. For example, containing class name
- Turquoise color are reflected values, where the placeholder value is filled-in

| Snippet | Short Name | Output Format |
|---|---|---|
| **Grid** | **grid** | <Grid ColumnDefinitions="" RowDefinitions=""><br><br></Grid> |
| **Flex Layout** | **flex** | <FlexLayout><br><br></FlexLayout> |
| **Stack Layout** | **stack** | <StackLayout><br><br></StackLayout> |
| **Horizontal Stack Layout** | **hstack** | <HorizontalStackLayout><br><br></HorizontalStackLayout> |
| **Vertical Stack Layout** | **vstack** | <VerticalStackLayout><br><br></VerticalStackLayout> |
| **Style** | **style** | <Style TargetType="Page"><br><br><Style> |
| Color | color | <Color x:Key="Success">Green</Color> |
| Resources | res | <ContentPage.Resources><br><br><ResourceDictionary><br><br></ResourceDictionary><br><br></ContentPage.Resources> |
| Gestures | gesture | <Label.GestureRecognizers><br><br></Label.GestureRecognizers> |
| Tap Gesture Recognizer | tap | <TapGestureRecognizer /> |
| Drag Gesture | drag | <DragGestureRecognizer /> |

| Snippet | Short Name | Output Format |
|---|---|---|
| Recognizer | | |
| Drop Gesture Recognizer | drop | <DropGestureRecognizer /> |
| Pan Gesture Recognizer | pan | <PanGestureRecognizer /> |
| Pinch Gesture Recognizer | pinch | <PinchGestureRecognizer /> |
| Swipe Gesture Recognizer | swipe | <SwipeGestureRecognizer /> |
| Blazor Web View | bwv | <b:BlazorWebView HostPage="wwwroot/index.html"><br><br>    <b:BlazorWebView.RootComponents><br><br>        <b:RootComponent ComponentType="{x:Type }" Selector="#app" /><br><br>    </b:BlazorWebView.RootComponents><br><br></b:BlazorWebView> |
| .NET MAUI Blazor Namespace | mb | xmlns:b="clr-namespace:Microsoft.AspNetCore.Components.WebView.Maui<br><br>;assembly=Microsoft.AspNetCore.Components.WebView.Maui" |
| WPF Blazor Namespace | wb | xmlns:b="clr-namespace:Microsoft.AspNetCore.Components.WebView.Wpf<br><br>;assembly=Microsoft.AspNetCore.Components.WebView.Wpf" |

Has C# Code Snippets for:

In the C# code file, type the short name and hit the Tab key twice to insert the snippet.

Snippets mentioned in bold-face also works as a **SurroundWith** snippet too (from **CSharp** section).

In the Output Format column, text highlighted in different colors infer the following:

- Yellow color are placeholders where user can modify the values

- Green color are derived values, can't be modified. For example, containing class name
- Turquoise color are reflected values, where the placeholder value is filled-in

| Snippet | Short Name | Output Format |
|---|---|---|
| **Async Event Handler** | **aeh** | private async void MyMethod(object sender, EventArgs e)<br>{<br>} |
| Attached Property | propap | *Here assuming MyClass is the containing type.*<br>public static readonly BindableProperty NameProperty = BindableProperty.CreateAttached(nameof(NameProperty), typeof(string), typeof(MyClass), default(string));<br>public static string GetName(BindableObject bindable) => (string)bindable.GetValue(NameProperty);<br>public static void SetName(BindableObject bindable, string value) => bindable.SetValue(NameProperty, value); |
| Bindable Property | propbp | *Here assuming MyClass is the containing type.*<br>public static readonly BindableProperty NameProperty = BindableProperty.Create(nameof(Name), typeof(string), typeof(MyClass), default(string));<br>public string Name<br>{<br>get => (string)GetValue(NameProperty);<br>set => SetValue(NameProperty, value);<br>} |
| Comet Property (MVU) | propc<br>(This has been shortened to **propc** from **propcomet**) | public string Name<br>{<br>get => GetProperty<string>();<br>set => SetProperty(value);<br>} |
| Cross Platform | cp<br>(This has been | #if ANDROID<br>#elif IOS |

| **Snippet** | **Short Name** | **Output Format** |
|---|---|---|
| | updated to **cp** from **xplat**) | #elif MACCATALYST<br><br>#elif TIZEN<br><br>#elif WINDOWS<br><br>#endif |
| **Event Handler** | **eh** | private void MyMethod(object sender, EventArgs e)<br><br>{<br><br>} |
| **Method** | **method** | private void MyMethod()<br><br>{<br><br>} |
| **Async Method** | **amethod** | private async Task MyMethod()<br><br>{<br><br>} |
| **Record**<br><br>**(C# 9.0 or higher)** | **record** | record MyRecord<br><br>{<br><br>} |
| **Record Struct**<br><br>**(C# 10.0 or higher)** | **rstruct**<br><br>(This has been updated to **rstruct** from **recstruct**) | record struct MyRecStruct<br><br>{<br><br>} |
| ViewModel Property | propvm | private string name;<br><br>public string Name<br><br>{<br><br>get => name;<br><br>set => SetProperty(ref name, value);<br><br>} |

| Snippet | Short Name | Output Format |
|---|---|---|
| C# Markup Extension Method | cmem | public static TBindable MyMethod&lt;TBindable&gt;(this TBindable bindable) where TBindable : BindableObject<br>{<br>    return bindable;<br>} |

Note: Currently, these project templates target .NET MAUI GA (VS2022 17.3 Preview 1.1 or later) and to support further changes in newer .NET MAUI releases, an update to this VS extension will be made available accordingly.