

Build MAUI control for AutoNavi map - iOS Native Binding

The last article introduced some of the main knowledge of iOS / Android MAUI controls for AutoNavi maps. Today, I will focus on the knowledge of iOS native library binding, and tell you some skills in the process of binding native libraries, hoping to give some inspiration to my friends.

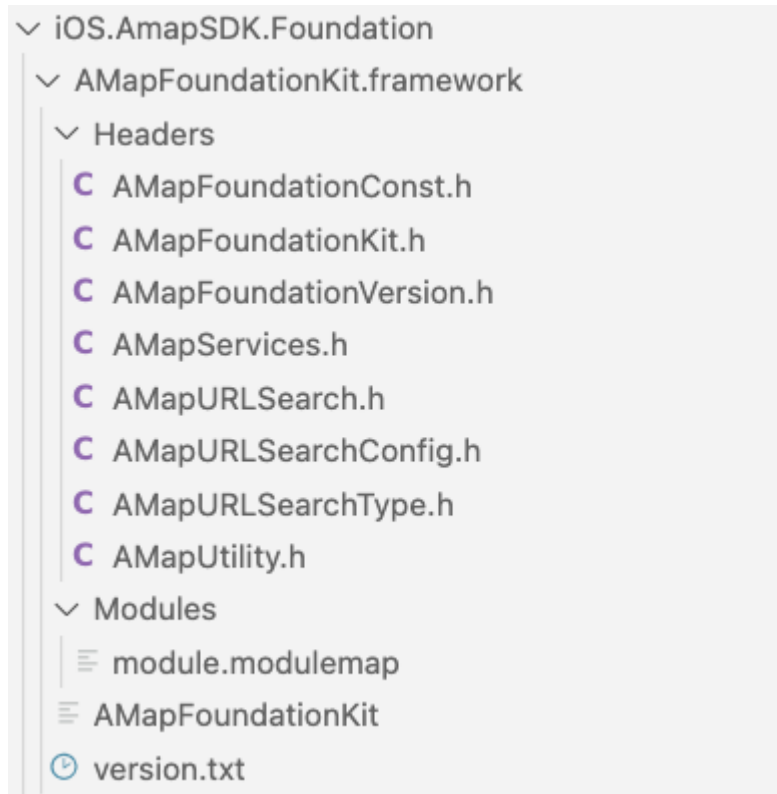
Know dynamic library and static library in iOS

Before binding, we need to learn about iOS dynamic library and static library. The easiest way to understand is that in iOS static libraries end with a .a suffix, and dynamic libraries end with a .dylib suffix. Both static libraries and dynamic libraries can be packaged into Framework .

Difference between static library and dynamic library

1. The characteristic of the static library is that the library file will be copied directly to the target application during compilation, and this copy resides in the target application, so after the compilation is completed, the files of the static library are useless. But there is a disadvantage that because of the need to copy, the capacity of the generated application will be larger
2. Dynamic libraries and static libraries are just the opposite. They will not be copied to the target application when compiling, so the size of the generated application is small, and a dynamic library can be shared with multiple applications. However, the generated application depends on the dynamic library, which often leads to the situation that the dynamic library cannot be found.

Let's disassemble the SDK of AMap Foundation - AMapFoundationKit.framework



This includes the corresponding header file information, module information, and static library. You can clearly see the implementation of the Gaode map packaged into Framework. This is also our understanding of the concept of libraries, compiled binary code, and exposing header files to third-party developers.

Generate an interface for C# calls with Sharpie tool

Shapie is a very useful conversion tool that supports the conversion of Objective-C libraries under macOS. Through Sharpie, the header file given by the library file can be converted to complete the C# binding. The Shapie tool already existed in MAUI's predecessor, and I often use this tool to do conversions.

Because I use 3D for the function of AutoNavi map this time, I will bind and convert the two Frameworks of AutoNavi's AMapFoundationKit.Framework and MAMapKit.framework.

Convert AMapFoundationKit.Framework

```
sharpie bind -framework AMapFoundationKit.framework -sdk iphoneos15.5
```

Convert MAMapKit.framework

```
sharpie bind -framework MAMapKit.framework -sdk iphoneos15.5
```

MAMapKit.framework depends on AMapFoundationKit.framework, so they should be in the same directory.

It should be noted here that you need to install Xcode. It is recommended to install it to the latest and correspond to the latest iOS SDK. Of course, you can also bind different versions of iOS SDK as needed. You can view the environment through one command

```
sharpie xcode -sdks
```

The two files generated by command line binding are StructsAndEnums.cs and ApiDefinitions.cs. StructsAndEnums.cs corresponds to some constants and enumeration types, and ApiDefinitions.cs corresponds to some interfaces and methods.

Create an iOS binding project for MAUI

It should be noted here that the templates for Visual Studio 2022 are not yet completed. Now you can use the command line to create it. Because we have two projects, the projects that need to create two Bindings are:

AMapFoundationKit.Framework

```
dotnet new iosbinding -o iOS.AMap.Foundation
```

MAMapKit.framework

```
dotnet new iosbinding -o iOS.AMap.3D
```

After generation, you need to put AMapFoundationKit.framework in the iOS.AMap.Foundation directory, and MAMapKit.framework in the iOS.AMap.3D directory. And put the generated StructsAndEnums.cs and ApiDefinitions.cs in the corresponding directory.

```

  v iOS.AmapSDK.3D
    > bin
    > MAMapKit.framework
    > obj
    C# ApiDefinition.cs
    iOS.AmapSDK.3D.csproj
    C# StructsAndEnums.cs
  v iOS.AmapSDK.Foundation
    > AMapFoundationKit.framework
    > bin
    > obj
    C# ApiDefinition.cs
    iOS.AmapSDK.Foundation.csproj
    C# StructsAndEnums.cs

```

Project Build

1. StructsAndEnum.cs in the directory generated by Sharpie, and ApiDefinition.cs in the built Binding directory, replace it. So make changes to the .csproj project

```

<ItemGroup>
  <ObjcBindingApiDefinition Include="ApiDefinitions.cs" />
  <ObjcBindingCoreSource Include="StructsAndEnums.cs" />
</ItemGroup>

```

2. Compile iOS.AMap.Foundation

Add a reference to Framework in AMapFoundationKit.framework.csproj

```

<ItemGroup>

```

```

    <NativeReference Include="AMapFoundationKit.framework">
      <Kind>Framework</Kind>
      <ForceLoad>True</ForceLoad>
      <SmartLink>False</SmartLink>
    </NativeReference>
  </ItemGroup>

```

Kind : The native binding type can be Framework or StaticLibrary

ForceLoad : strong loading, select True

SmartLink: Smart Link

The finished project .csproj is set to

```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net6.0-ios</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>true</ImplicitUsings>
    <IsBindingProject>true</IsBindingProject>
    <AllowUnsafeBlocks>true</AllowUnsafeBlocks>
    <NoBindingEmbedding>>false</NoBindingEmbedding>
  </PropertyGroup>

  <ItemGroup>
    <ObjcBindingApiDefinition Include="ApiDefinitions.cs" />
    <ObjcBindingCoreSource Include="StructsAndEnums.cs" />
  </ItemGroup>

  <ItemGroup>
    <NativeReference Include="AMapFoundationKit.framework">
      <Kind>Framework</Kind>
      <ForceLoad>True</ForceLoad>
      <SmartLink>False</SmartLink>
    </NativeReference>
  </ItemGroup>

</Project>

```

Compiling iOS.AMap.Foundation , you'll get a crash because of a lot of error messages. This is because when Shapie does the conversion, some conversions are not done well. At this time, you need to adjust one by one.

```

piDefinitions.cs(156,2): error CS0246: The type or namespace name 'VerifyAttribute' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(156,2): error CS0246: The type or namespace name 'Verify' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(8,10): error CS0103: The name 'ConstantsInterfaceAssociation' does not exist in the current context [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(156,10): error CS0103: The name 'ConstantsInterfaceAssociation' does not exist in the current context [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(155,2): error CS0579: Duplicate 'Static' attribute [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(31,3): error CS0246: The type or namespace name 'VerifyAttribute' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(31,3): error CS0246: The type or namespace name 'Verify' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
piDefinitions.cs(31,11): error CS0103: The name 'MethodToProperty' does not exist in the current context [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(68,3): error CS0246: The type or namespace name 'VerifyAttribute' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(68,3): error CS0246: The type or namespace name 'Verify' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(68,11): error CS0103: The name 'PlatformInvoke' does not exist in the current context [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(73,3): error CS0246: The type or namespace name 'VerifyAttribute' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(73,3): error CS0246: The type or namespace name 'Verify' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(73,11): error CS0103: The name 'PlatformInvoke' does not exist in the current context [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(78,3): error CS0246: The type or namespace name 'VerifyAttribute' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(78,3): error CS0246: The type or namespace name 'Verify' could not be found (are you missing a using directive or an assembly reference?) [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]
structsAndEnums.cs(78,11): error CS0103: The name 'PlatformInvoke' does not exist in the current context [/Users/lokinfey/Desktop/AMap.iOS.Demo/iOS.AMap.Foundation/iOS.AMap.Foundation.csproj]

```

Error messages

**The type or namespace name 'VerifyAttribute' could not be found*

For this type of information, because the attributes are not confirmed during the conversion, the VerifyAttribute field will be added. In general, this field can be commented out, such as

```

static class CFunctions
{
    // NSString * AMapEmptyStringIfNil (NSString *s);
    [DllImport ("__Internal")]
    // [Verify (PlatformInvoke)]
    static extern NSString AMapEmptyStringIfNil (NSString s);

    // extern CLLocationCoordinate2D AMapCoordinateConvert
    (CLLocationCoordinate2D coordinate, AMapCoordinateType type);
    [DllImport ("__Internal")]
    // [Verify (PlatformInvoke)]
    static extern CLLocationCoordinate2D AMapCoordinateConvert
    (CLLocationCoordinate2D coordinate, AMapCoordinateType type);

    // extern BOOL AMapDataAvailableForCoordinate (CLLocationCoordinate2D
    coordinate);
    [DllImport ("__Internal")]
    // [Verify (PlatformInvoke)]
    static extern bool AMapDataAvailableForCoordinate
    (CLLocationCoordinate2D coordinate);
}

```

***The type or namespace name 'AMapFoundationKit'**

Namespace problem, you need to add named controls to StructsAndEnums.cs and ApiDefinitions.cs. You can use AMapFoundationKit directly, or you can modify the name you like. I use iOS.AMap.Foundation here to correspond to the name of the project

***Duplicate 'Static' attribute**

This is because the Constants of ApiDefinitions.cs are repeatedly defined, and this needs to be rearranged and merged into one.

***Unsupported type for Fields: bool for 'iOS.AMap.Foundation.Constants_omapLocationOverseas'.e**

The type does not correspond to cause the compilation to fail. At this time, I modified it to

```
[Field ("_omapLocationOverseas", "__Internal")]
IntPtr _omapLocationOverseas { get; }
```

这样你就可以编译通过 iOS.AMap.Foundation

This way you can compile via iOS.AMap.Foundation

Add reference to iOS.AMap.Foundation

Because MAMapKit.framework depends on AMapFoundationKit.framework , so iOS.AMap.3D depends on iOS.AMap.Foundation

```
<ItemGroup>
  <ProjectReference
    Include="..\iOS.Amap.Foundation\iOS.Amap.Foundation.csproj" />
</ItemGroup>
```

import MAMapKit.framework

```
<ItemGroup>
  <NativeReference Include="MAMapKit.framework">
    <Kind>Framework</Kind>
    <ForceLoad>True</ForceLoad>
    <SmartLink>True</SmartLink>
```

```

    <Frameworks>GLKit OpenGLES UIKit Foundation CoreGraphics QuartzCore
CoreLocation CoreTelephony SystemConfiguration Security AdSupport
JavaScriptCore</Frameworks>
    <LinkerFlags>-lz -lstdc++ -lc++</LinkerFlags>
  </NativeReference>
</ItemGroup>

```

This is different from AMapFoundationKit.framework. You need to add the items that the Framework needs to depend on when compiling, and the compilation method used. This is related to the framework you bind. I choose Gaode Map here, so I did it according to their documentation requirements. related settings.

The finished project .csproj is set to

```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net6.0-ios</TargetFramework>
    <RootNamespace>iOS.Amap._3D</RootNamespace>
    <Nullable>enable</Nullable>
    <ImplicitUsings>true</ImplicitUsings>
    <IsBindingProject>true</IsBindingProject>
    <AllowUnsafeBlocks>true</AllowUnsafeBlocks>
    <NoBindingEmbedding>>false</NoBindingEmbedding>
  </PropertyGroup>
  <ItemGroup>
    <ObjcBindingApiDefinition Include="ApiDefinitions.cs" />
    <ObjcBindingCoreSource Include="StructsAndEnums.cs" />
  </ItemGroup>
  <ItemGroup>
    <NativeReference Include="MAMapKit.framework">
      <Kind>Framework</Kind>
      <ForceLoad>True</ForceLoad>
      <SmartLink>True</SmartLink>
      <Frameworks>GLKit OpenGLES UIKit Foundation CoreGraphics QuartzCore
CoreLocation CoreTelephony SystemConfiguration Security AdSupport
JavaScriptCore</Frameworks>
      <LinkerFlags>-lz -lstdc++ -lc++</LinkerFlags>
    </NativeReference>
  </ItemGroup>

  <ItemGroup>
    <ProjectReference
Include="..\iOS.Amap.Foundation\iOS.Amap.Foundation.csproj" />
  </ItemGroup>
</Project>

```


Compile iOS.AMap.3D, you will be more broken than before. At this time, you need to have enough patience. In addition to the same error message as before, there are some new situations. I will list them here.

****Type 'MAMapViewDelegate' already defines a member called 'MapView' with the same parameter types***

The reason for this is because the method has the same name, which is also the difference between the Objective-C declarative syntax and the traditional syntax, so you need to rename it for this

such as this

```
// @optional -(void)mapView:(MAMapView *)mapView didAnnotationViewTapped:
(MAAnnotationView *)view;
[Export ("mapView:didAnnotationViewTapped:")]
void MapView (MAMapView mapView, MAAnnotationView view);
```

Changed

```
// @optional -(void)mapView:(MAMapView *)mapView didAnnotationViewTapped:
(MAAnnotationView *)view;
[Export ("mapView:didAnnotationViewTapped:")]
void MapViewDidAnnotationViewTapped (MAMapView mapView, MAAnnotationView
view);
```

****The type or namespace name 'IMAOOverlay' could not be found***

This is a naming error, in the ApiDefinitions.cs file you can find MAOverlay

```
[Protocol]
interface MAOverlay : IMAAnnotation
{
    // @required -(CLLocationCoordinate2D)coordinate;
    [Abstract]
    [Export ("coordinate")]
    // [Verify (MethodToProperty)]
    CLLocationCoordinate2D Coordinate { get; }

    // @required -(MAMapRect)boundingMapRect;
    [Abstract]
    [Export ("boundingMapRect")]
    // [Verify (MethodToProperty)]
    MAMapRect BoundingMapRect { get; }
```

```
}
```

So replace all IMAOverlay with MAOverlay.

****The type or namespace name 'AutoGeneratedName' could not be found***

Cancel AutoGeneratedName

****Constant value '-1' cannot be converted to a 'ulong'***

Specified type error AllCorners = ~0x0 Changed to AllCorners = 0x0

Do not know how to make a signature for CoreLocation.CLLocationCoordinate2D in parameter 'coordinates'

C# doesn't have pointers, and Sharpie casts an error

****'MAMapView_UserLocation.HeadingFilter': cannot declare instance members in a static class***

```
// @property (nonatomic) CLLocationDegrees headingFilter;
[Export ("headingFilter")]
double HeadingFilter( { get; set; })
```

Changed

```
// @property (nonatomic) CLLocationDegrees headingFilter;
[Export ("headingFilter")]
double HeadingFilter();
```

****Cannot convert type 'Foundation.NSObject' to 'nint'***

```
// @property (nonatomic, weak) id<MAOverlayRenderDelegate>
rendererDelegate;
[Nullable, Export ("rendererDelegate", ArgumentSemantic.Weak)]
NSObject WeakRendererDelegate { get; set; }
```

Changed

```
// @property (nonatomic, weak) id<MAOverlayRenderDelegate>
rendererDelegate;
[Nullable, Export ("rendererDelegate", ArgumentSemantic.Weak)]
IntPtr WeakRendererDelegate { get; set; }
```

Or exclusion is a long process, but you will be very excited when the compilation is successful, so we have successfully bound AMapFoundationKit.framework and MAMapKit.framework.

Build .NET for iOS





Please download from my GitHub Repo :

<https://github.com/kinfey/AMapMAUIControls/tree/main/samples/iOS.Bindings/AMap.iOS.Demo>

Summary

Although the native library binding is more complicated, it is actually very healing. When you see the compilation pass, you will understand the happiness. Another point, many people think that cross-platform

mobile development does not require basic knowledge of the platform, but it is still required. Especially in the binding of this kind of native library, you need to know both C# and Objective-C. I hope this example can inspire you. Please look forward to the next one.

More

1. Learn about MAUI through Microsoft Docs <https://aka.ms/Docs.MAUI>
2. Learn MAUI through Microsoft Learn <https://aka.ms/Learn.MAUI>
3. To use Amap SDK for iOS, please visit <https://developer.amap.com/api/ios-sdk/gettingstarted>
4. To understand the content of iOS native library binding, please visit <https://docs.microsoft.com/en-us/xamarin/cross-platform/macios/binding/?context=xamarin%2Fios>