



MAUI For AsReaderDockSDK4

SDK マニュアル V1.0

変更履歴

No.	バージョン	変更内容	日付
1	1.0	新規作成	2024/01/18

目次

1 SDK の追加	4
1.1 SDK をインポートする.....	4
1.2 プロトコルの追加.....	12
1.3 権限の追加.....	12
1.3.1 Bluetooth の使用権限を追加する.....	12
2 SDK の使用	13
2.1 SDK のインポート.....	13
2.2 「AsReaderDevice」デリゲートクラスを作成して、デリゲートメソッドを実現する.....	14
2.3 AsReaderBarcodeDevice オブジェクトを初期化する.....	15
2.4 Plugged を実現する.....	15
2.5 ReaderConnected を実現する.....	15
2.6 ReceivedScanData を実現する.....	16
2.7 電源 On.....	16
2.8 スキャン開始.....	16
2.9 スキャン停止.....	17
2.10 電源 Off.....	17
3 AsReader Dock SDK4 MAUI .Net の変更	18
4 Api の使用	21



アプリケーションを作成する開発環境は、以下のものを推奨します。

開発環境：

.Net バージョン：.Net 7.0

Visual Studio: Visual Studio Community 2022 for Mac 17.6.7 (build 417)

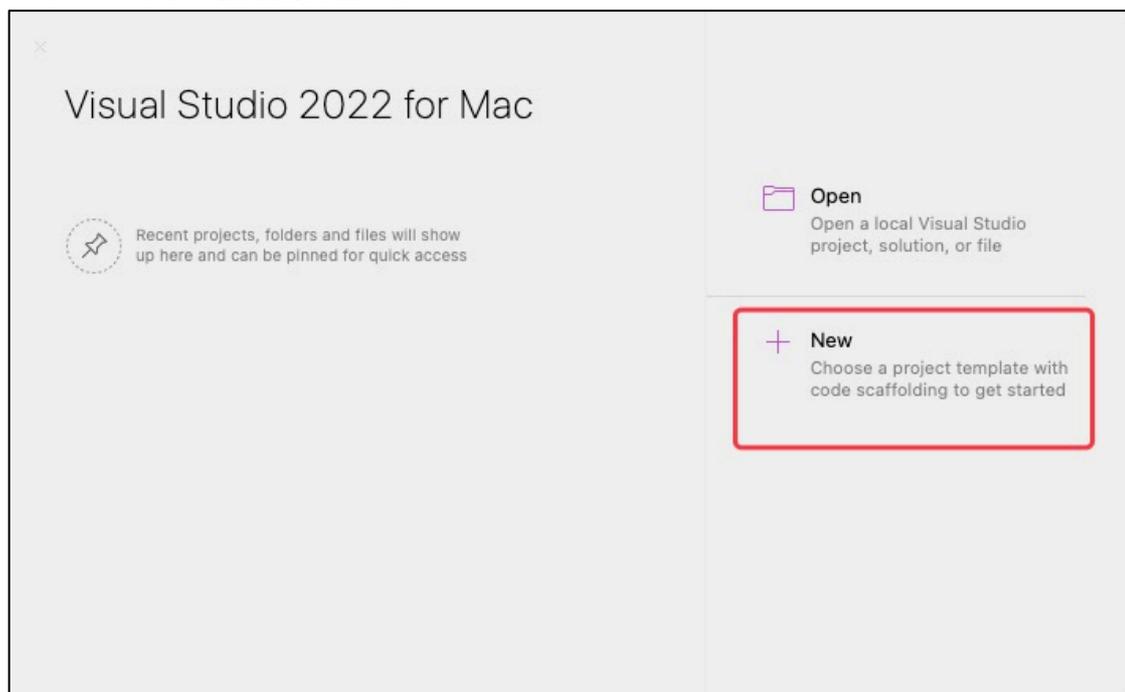
Xcode：15.0.1

この文書の説明には、上記開発環境を使用しています。

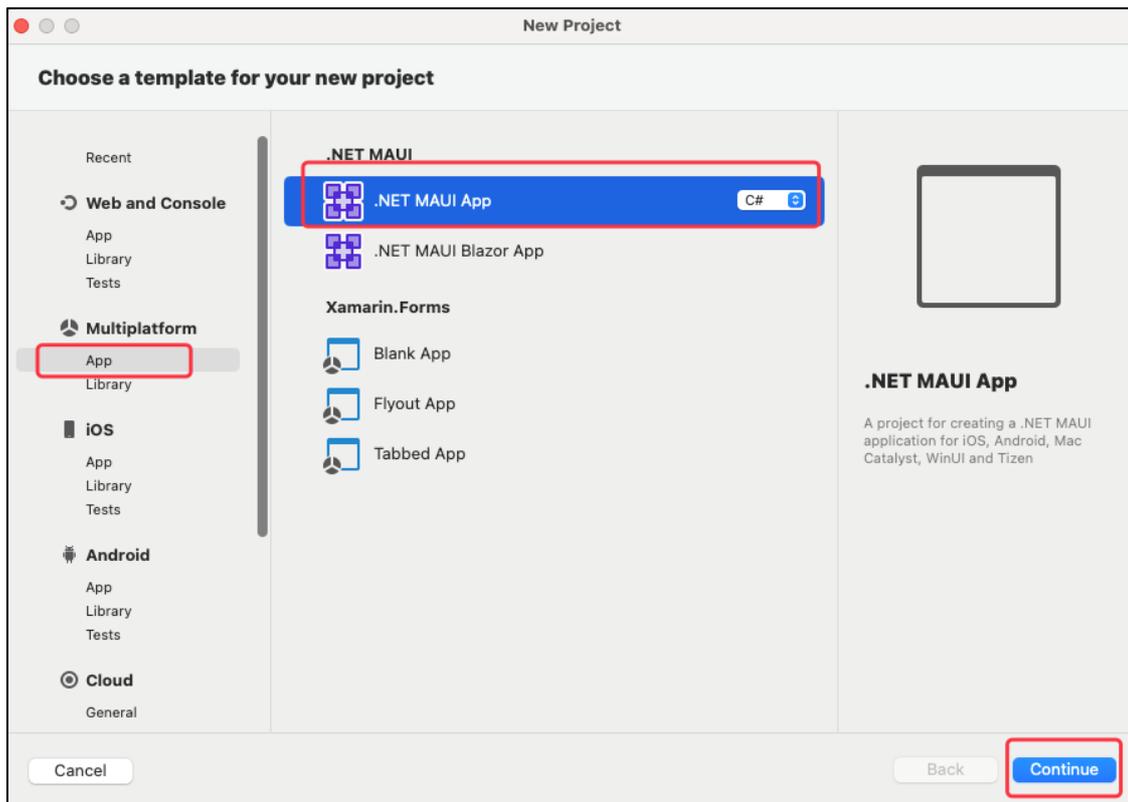
1 SDK の追加

1.1 SDK をインポートする

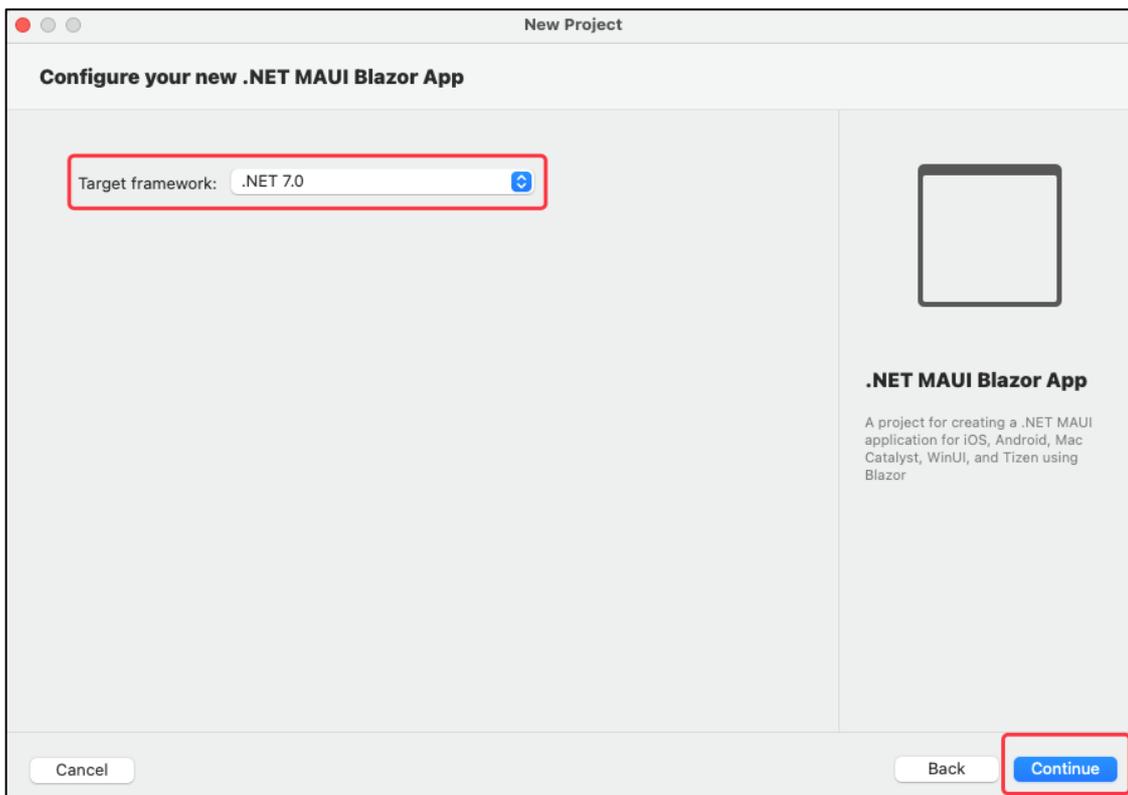
1. AsReaderBLESDK と AsReaderSDK4 をローカルにダウンロードします。
2. プロジェクトを新規作成します。



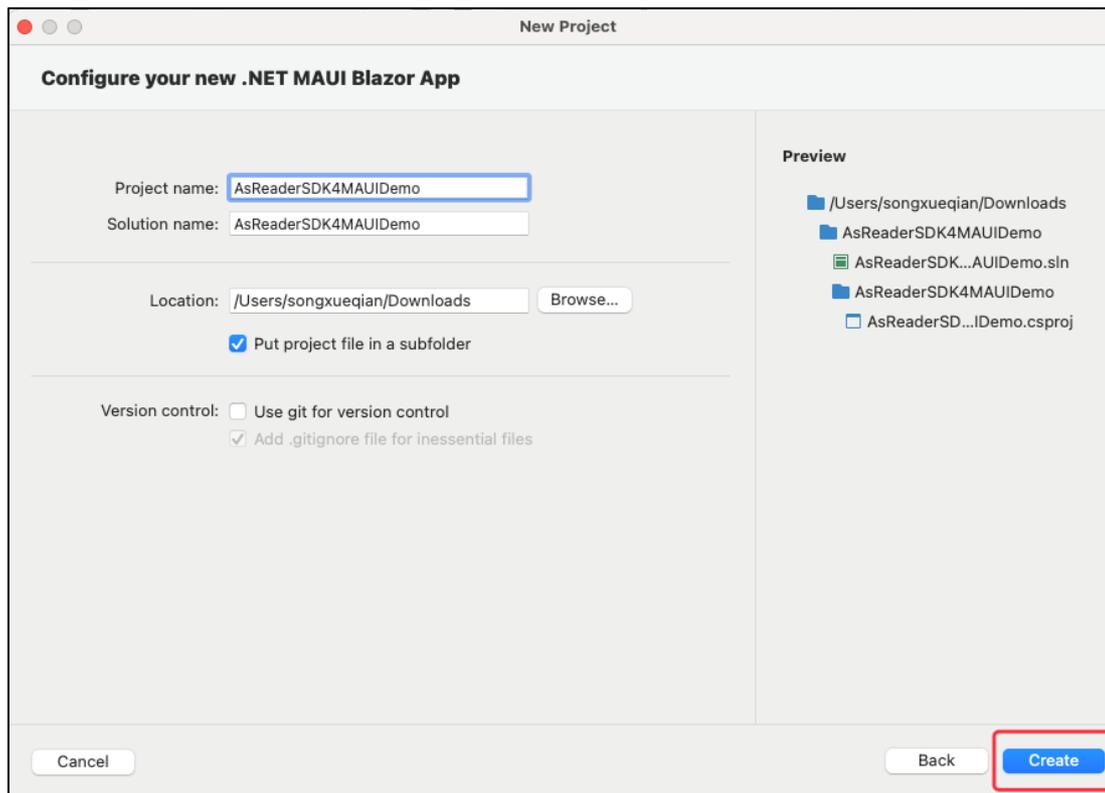
3. 「.NET MAUI APP」を選択します。



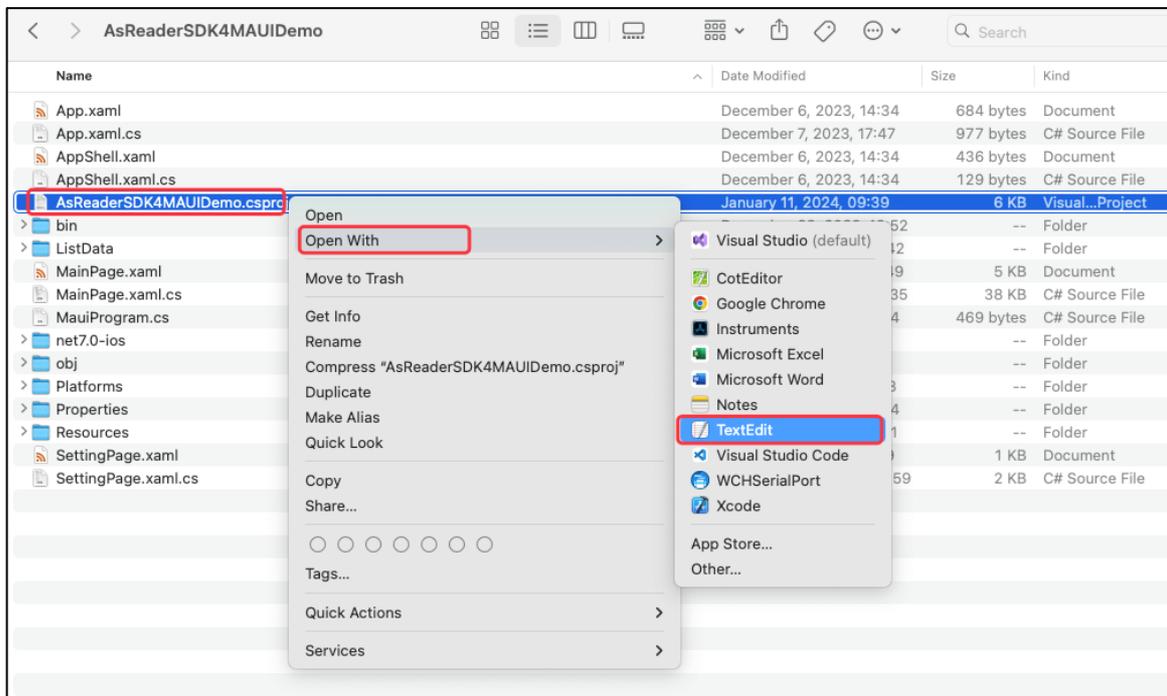
4. 「.NET 7.0」を選択します。



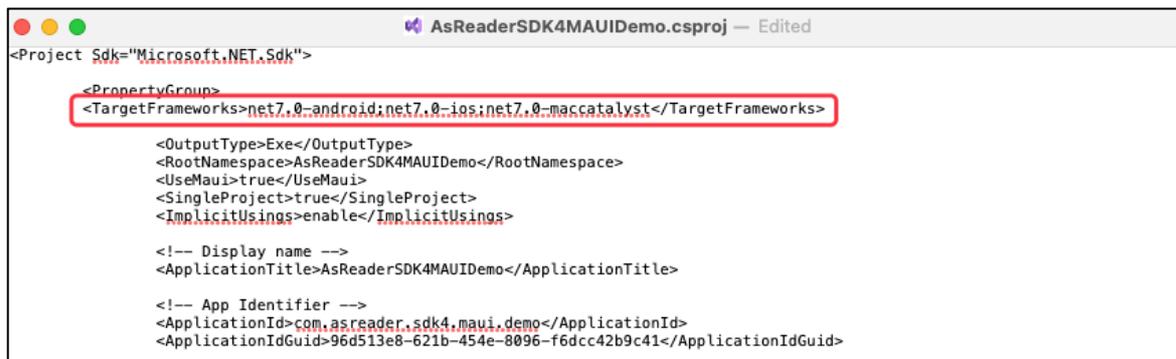
5. アプリ名とロケーションを作成します。



6. 「AsReaderSDK4MAUIDemo.csproj」を右クリックして、TextEdit で開きます。



7. 下図赤枠内 (net7.0-ios) で、必要としない全てのフレームを削除します。(iOS プラットフォームのみ使用した場合の操作です。マルチプラットフォームの場合、この操作は不要です。)

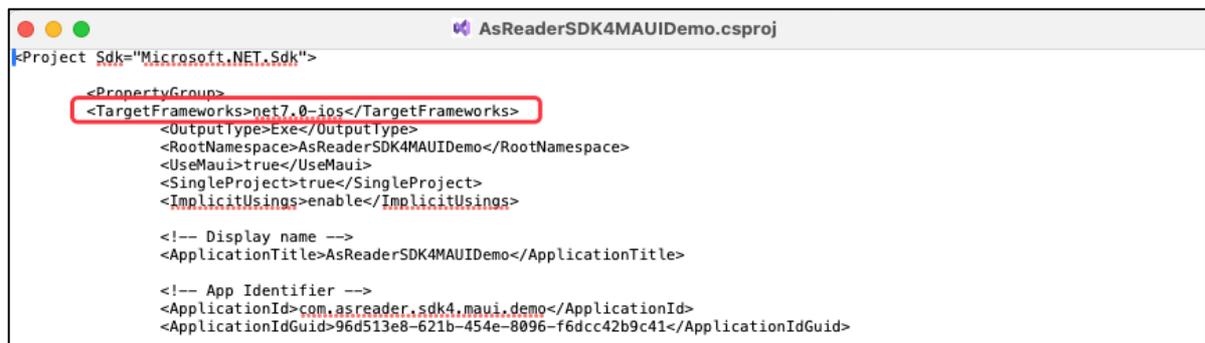


```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net7.0-android;net7.0-ios;net7.0-maccatalyst</TargetFrameworks>
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderSDK4MAUIDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderSDK4MAUIDemo</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.asreader.sdk4.maui.demo</ApplicationId>
    <ApplicationIdGuid>96d513e8-621b-454e-8096-f6dcc42b9c41</ApplicationIdGuid>
  </PropertyGroup>
</Project>
```

8. 削除完了後は下図の通りです。

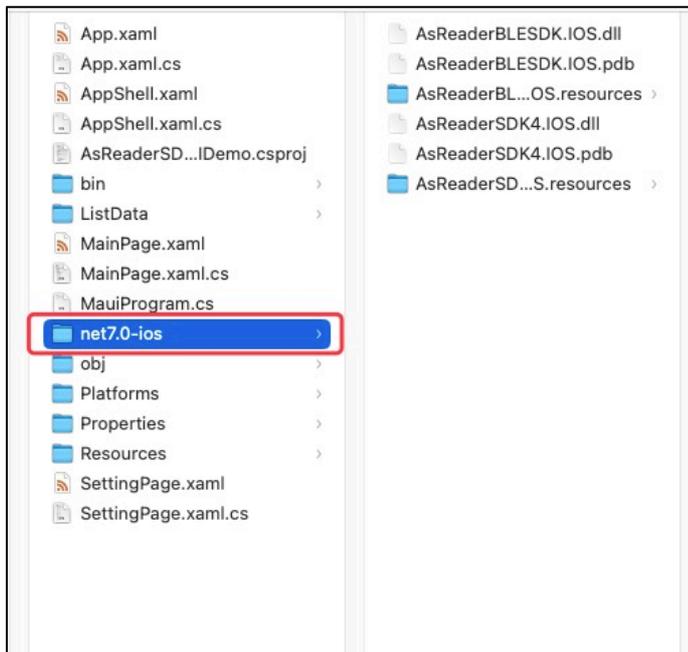


```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net7.0-ios</TargetFrameworks>
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderSDK4MAUIDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

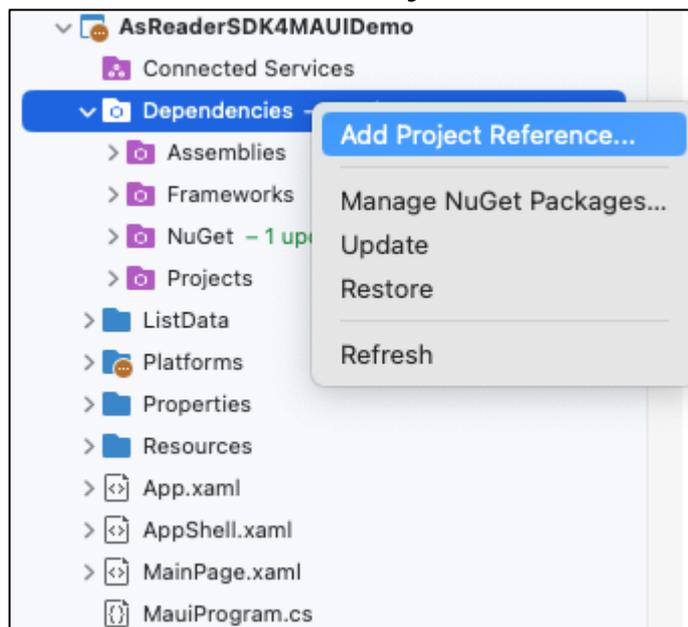
    <!-- Display name -->
    <ApplicationTitle>AsReaderSDK4MAUIDemo</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.asreader.sdk4.maui.demo</ApplicationId>
    <ApplicationIdGuid>96d513e8-621b-454e-8096-f6dcc42b9c41</ApplicationIdGuid>
  </PropertyGroup>
</Project>
```

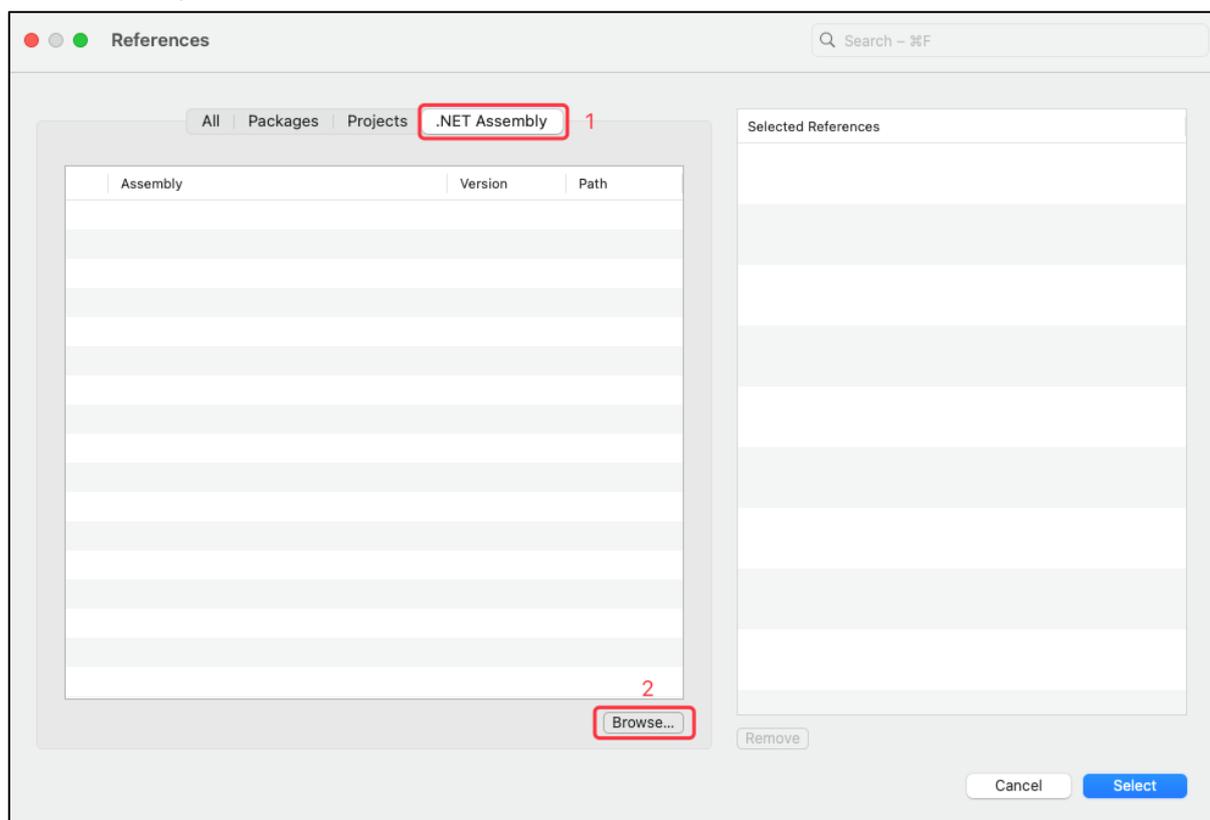
9. ローカルにダウンロードした AsReaderBLESDK と AsReaderSDK4 をプロジェクトフォルダーにコピーします。



10. Dependencies -> 右クリック -> Add Project Reference...

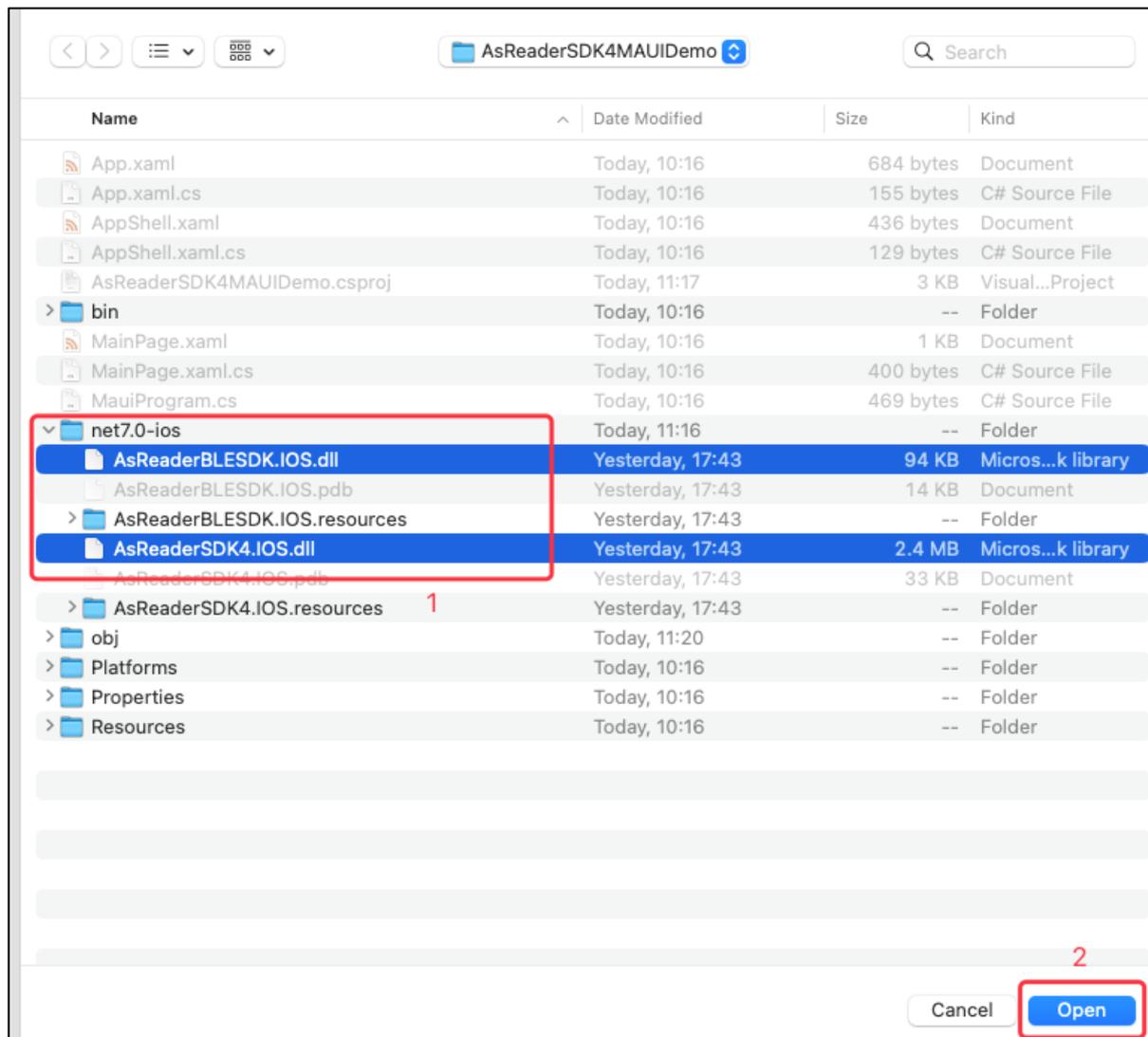


11. ポップアップしたウィンドウに「.Net Assembly」（下図 1）をクリックして、「Browse…」（下図 2）を選択します。

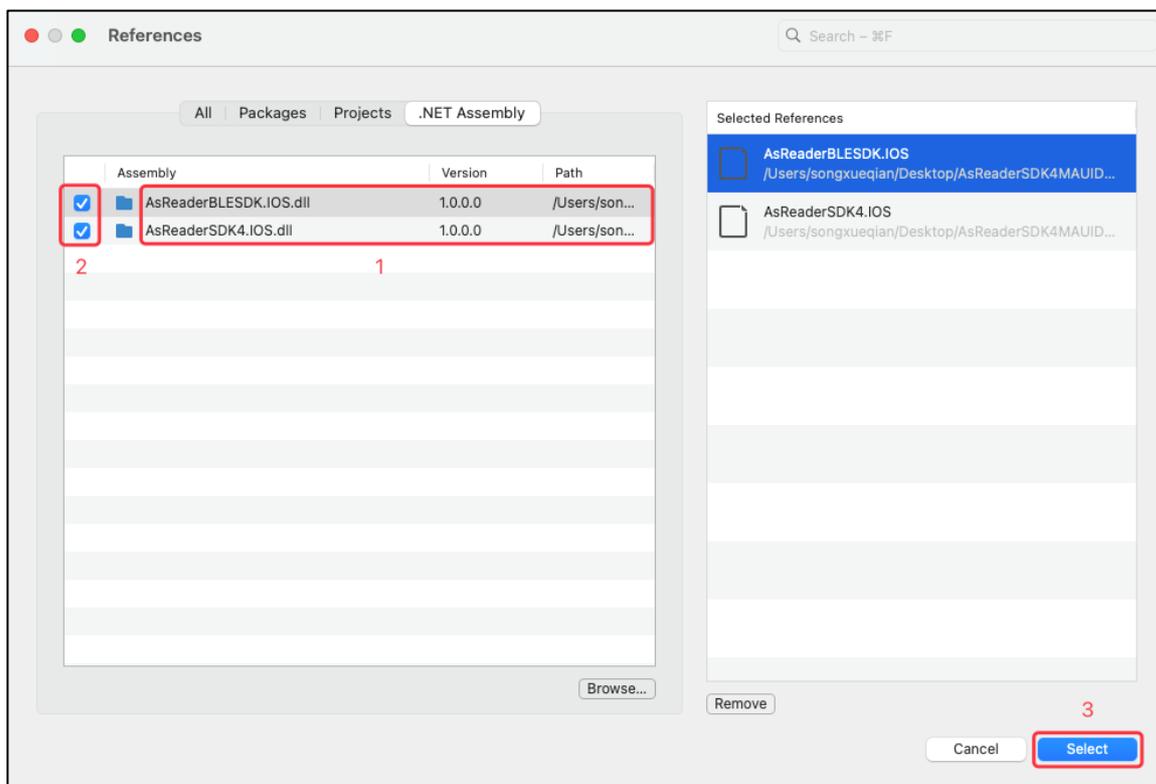


12. ポップアップしたウィンドウに「AsReaderBLESDK.IOS.dll」と

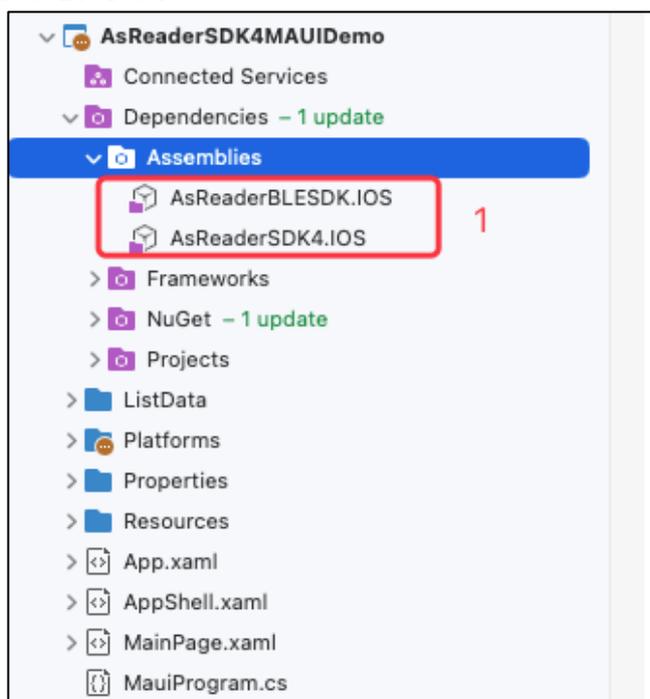
「AsReaderSDK4.IOS.dll」（下図 1）を選択して、「Open」ボタン（下図 2）をクリックします。



13. 「References」ウィンドウのリストにある「AsReaderBLESDK.IOS.dll」と「AsReaderSDK4.IOS.dll」左側のチェックボックスを選択して、「Select」ボタンをクリックします。



14. 追加後、「Assemblies」に「AsReaderBLESDK.IOS」と「AsReaderSDK4.IOS」(下図 1) が表示されます。



15. マルチプラットフォームを使用する場合、以下の操作を行ってください。

```
<ItemGroup Condition="'$(TargetFramework)' == 'net7.0-ios' ">
  <Reference Include="AsReaderBLESDK.IOS">
    <HintPath>net7.0-ios\AsReaderBLESDK.IOS.dll</HintPath>
  </Reference>
  <Reference Include="AsReaderSDK4.IOS">
    <HintPath>net7.0-ios\AsReaderSDK4.IOS.dll</HintPath>
  </Reference>
</ItemGroup>
```

```
</Project>
```

1.2 プロトコルの追加

Info.plist にサポートしているプロトコルを追加します。

▼ Supported external accessory protocols	⇅	Array	(8 items)
Item 0	⊕ ⊖	String	kr.co.sps.sreader.universal.sr120
Item 1		String	jp.co.asx.asreader.nfc
Item 2		String	jp.co.asx.asreader.6dongle.rfid
Item 3		String	jp.co.asx.asreader.6dongle.barcode
Item 4		String	jp.co.asx.asreader.rfid
Item 5		String	jp.co.asx.asreader.barcode
Item 6		String	jp.co.asx.asreader.0240D
Item 7		String	jp.co.asx.asreader.0230D

1.3 権限の追加

1.3.1 Bluetooth の使用権限を追加する

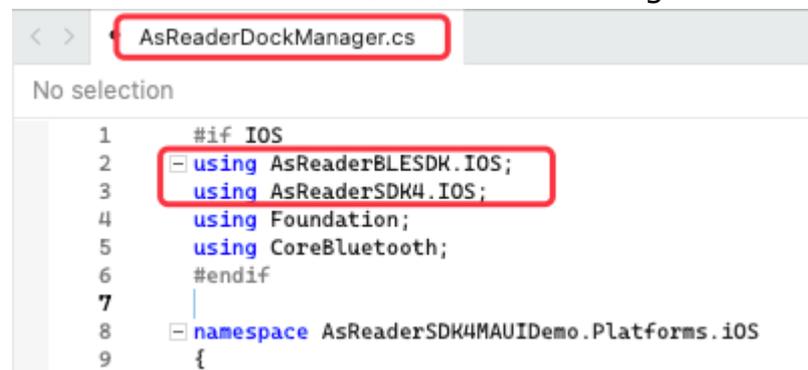
Info.plist に Bluetooth の使用権限を追加します。

NSBluetoothPeripheralUsageDescription	⇅	String	Bluetooth description
NSBluetoothAlwaysUsageDescription	⇅	String	Bluetooth description

2 SDK の使用

2.1 SDK のインポート

アプリケーションの SDK を使用するクラスに using 文で SDK をインポートします。



```
< > AsReaderDockManager.cs
No selection
1  #if IOS
2  using AsReaderBLESDK.IOS;
3  using AsReaderSDK4.IOS;
4  using Foundation;
5  using CoreBluetooth;
6  #endif
7
8  namespace AsReaderSDK4MAUIDemo.Platforms.iOS
9  {
```

2.2 「AsReaderDevice」デリゲートクラスを作成して、デリゲートメソッドを実現する

※Platforms の iOS フォルダに作成することをお勧めします。

```
public class AsReaderDockManagerDeviceDelegate : AsReaderDeviceDelegate
{
    public override void Plugged(bool plug)
    {
        if (plug)
        {
            //Connection successful
        }
        else
        {
            //Disconnect
        }
    }
    public override void ReaderConnected(int status)
    {
        if (status == 0xff)
        {
            //Power on
        }
        else
        {
            //Power off
        }
    }
    public override void ReceivedScanData(NSData readData)
    {
        if (AsReaderInfo.SharedInstance().CurrentReaderMode == ReaderMode.Barcode)
        {
            NSString str = new NSString();
            try
            {
                str = new NSString(readData, NSStringEncoding.ShiftJIS);
            }
            catch
            {
                try
                {
                    str = new NSString(readData, NSStringEncoding.UTF8);
                }
                catch
                {
                    try
                    {
                        str = new NSString(readData, NSStringEncoding.ASCIIStringEncoding);
                    }
                    catch
                    {
                        str = new NSString("Encoding Error");
                    }
                }
            }
        }
    }
}
```

2.3 AsReaderBarcodeDevice オブジェクトを初期化する

```
AsReaderBarcodeDevice asReaderBarcodeDevice = AsReaderBarcodeDevice.SharedInstance();  
asReaderBarcodeDevice.DelegateDevice = new AsReaderDockManagerDeviceDelegate();
```

2.4 Plugged を実現する

```
public override void Plugged(bool plug)  
{  
    if (plug)  
    {  
        //Connection successful  
    }  
    else  
    {  
        //Disconnect  
    }  
}
```

2.5 ReaderConnected を実現する

```
public override void ReaderConnected(int status)  
{  
    if (status == 0xff)  
    {  
        //Power on  
    }  
    else  
    {  
        //Power off  
    }  
}
```

2.6 ReceivedScanData を実現する

```
public override void ReceivedScanData(NSData readData)
{
    if (AsReaderInfo.SharedInstance().CurrentReaderMode == ReaderMode.Barcode)
    {
        NSString str = new NSString();
        try
        {
            str = new NSString(readData, NSStringEncoding.ShiftJIS);
        }
        catch
        {
            try
            {
                str = new NSString(readData, NSStringEncoding.UTF8);
            }
            catch
            {
                try
                {
                    str = new NSString(readData, NSStringEncoding.ASCIIStringEncoding);
                }
                catch
                {
                    str = new NSString("Encoding Error");
                }
            }
        }
    }
}
```

2.7 電源 On

デリゲート Plugged からコールバック成功の結果を受け取り、電源 On のメソッドを呼び出します。

```
AsReaderDevice device = AsReaderBarcodeDevice.SharedInstance();
device.SetReaderPower(true, true, true, true, true, true, 0);
```

2.8 スキャン開始

```
_ = AsReaderBarcodeDevice.SharedInstance().StartScan;
```

スキャン結果はデリゲート ReceivedScanData にコールバックされます。

2.9 スキャン停止

```
_ = AsReaderBarcodeDevice.SharedInstance().StopScan;
```

2.10 電源 Off

```
AsReaderDevice device = AsReaderBarcodeDevice.SharedInstance();  
device.SetReaderPower(false, true, true, true, true, true, 0);
```

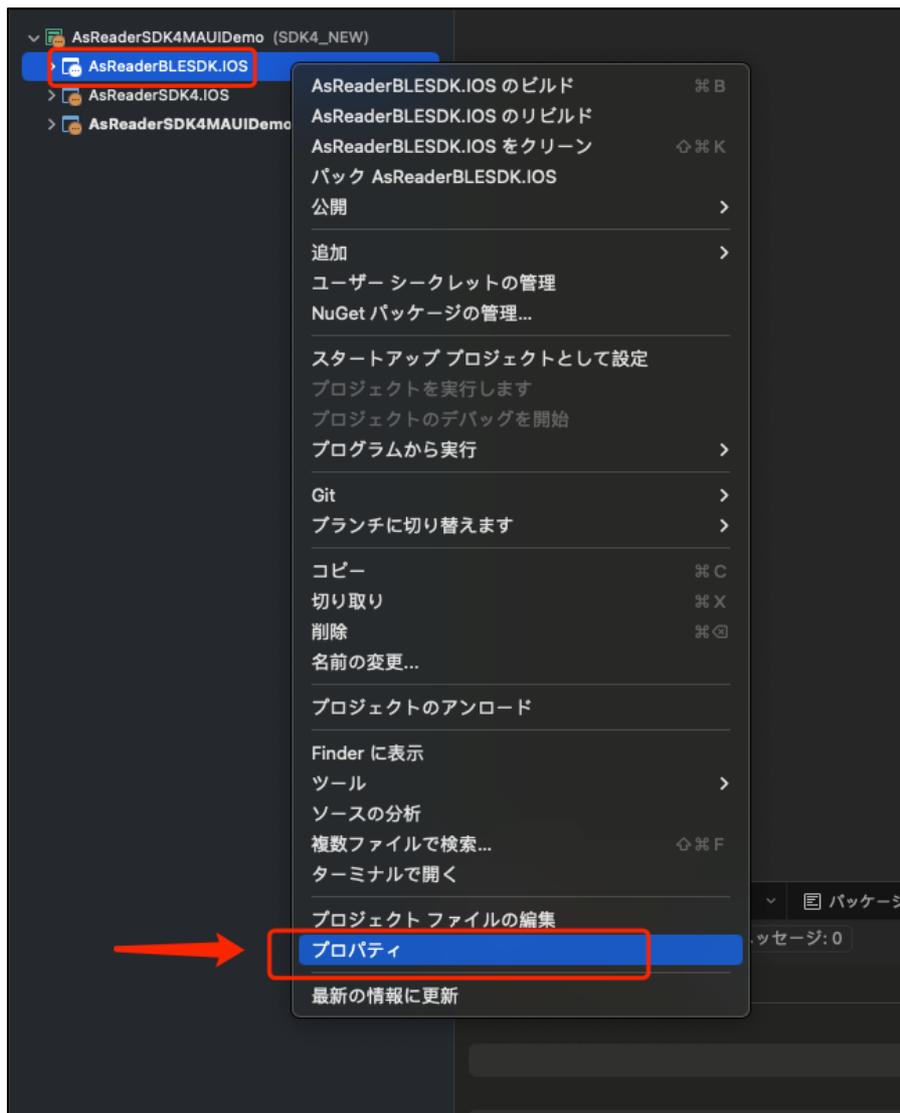
3 AsReader Dock SDK4 MAUI .Net の変更

デモアプリはビルドした時に使用したバージョンのツールでしか使えません。

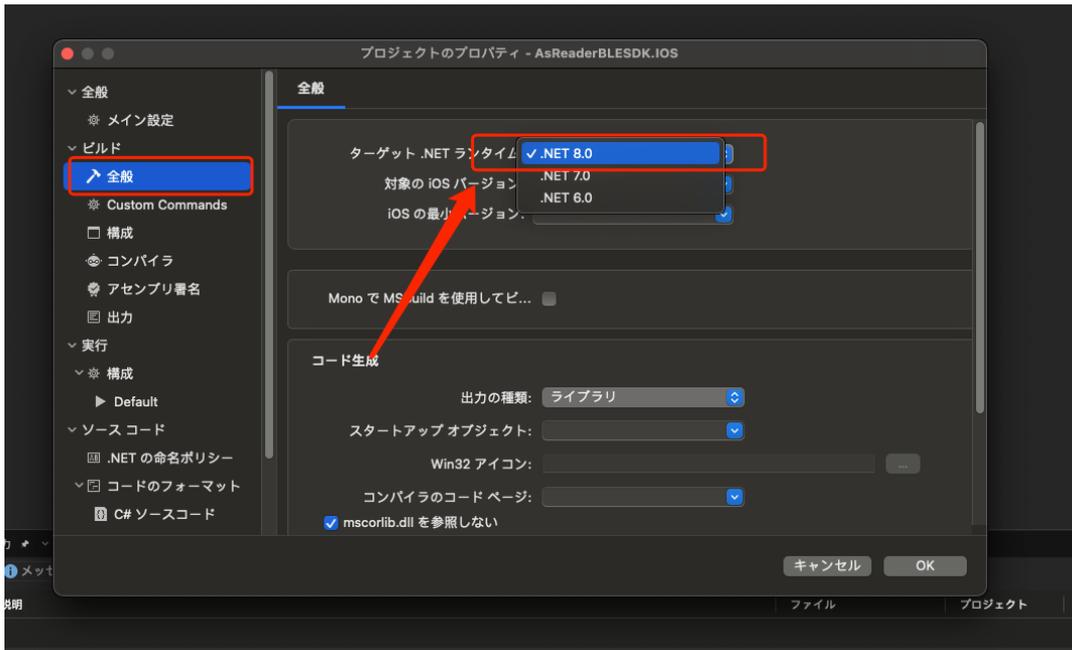
例：「.net MAUI 8」でビルドした場合、「.net MAUI 8」だけで使えます。「.net MAUI 7」や「.net MAUI 9」で使えません。

以下は例として「.net MAUI 7」を「.net MAUI 8」に変更する方法です。

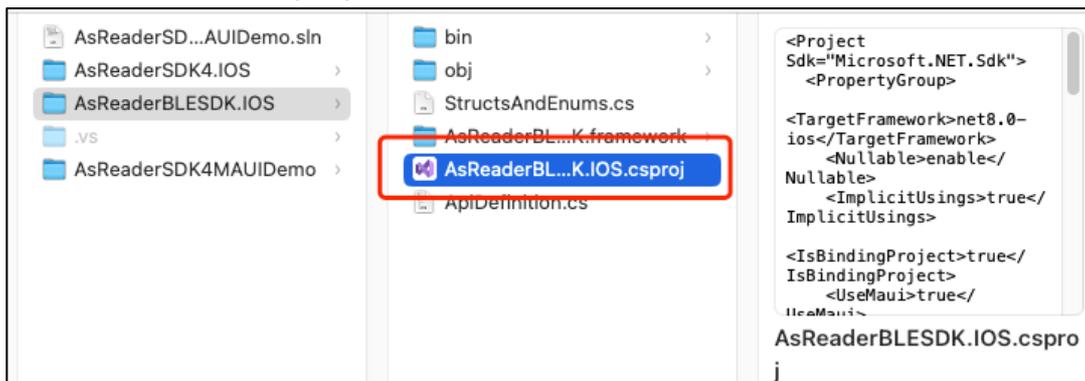
1. プロジェクトを右クリック->プロパティ



2. 全般->ターゲット.NET ランタイムに「.NET 8.0」を選択します。



3. プロジェクトフォルダに.csproj を選択して、editor で開きます。



4. 以下の修正を行います。



<TargetFramework>net7.0-ios</TargetFramework>

↓

<TargetFramework>net8.0-ios</TargetFramework>



以下の配置を追加します。

```
<UseMaui>true</UseMaui>
```

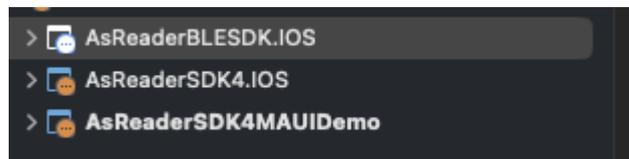
```
<MauiVersion>8.0.3</MauiVersion>
```

```
<SkipValidateMauiImplicitPackageReferences>true</SkipValidateMauiImplicitPackageReferences>
```

修正後にファイルを保存します。

4.保存後に改めてプロジェクトをローディングします。

5.以下のプロジェクトは、全て上記 1 ～ 4 の通りに修正する必要があります。





4 Api の使用

Api の使用について、Objective-C の SDK マニュアルを参照してください。

「AsReaderDockSDK4_SDK_Reference_Guide_ObjectiveC_JP_1_13」