



.NET MAUI Android SDK For AsReaderSLIM

SDK マニュアル 第 1 版

変更履歴

No.	版数	変更内容	日付
1	第 1 版	新規作成	2024/10/30

目 次

1 SDK の追加	4
1.1 SDK をインポートする	4
1.2 権限の追加	11
2 SDK の使用	12
2.1 SDK のインポート	12
2.2 「MyAsReaderDeviceDelegate」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	12
2.3 AsReaderManager オブジェクトを初期化	12
2.4 OnStateChanged を実現	13
2.5 OnReadBarcodeData を実現.....	13
2.6 USB 接続	13
2.7 Barcode スキャン開始	13
2.8 Barcode スキャン停止	13
2.9 切断.....	13
3 ビルドする.NET バージョンの変更	14
4 API の使用	17

この文書の説明には、下記開発環境を使用しています。

開発環境：

.NET バージョン：.NET 7.0

Visual Studio: Visual Studio Community 2022 for Mac 17.6.12 (build 410)

Android Studio：

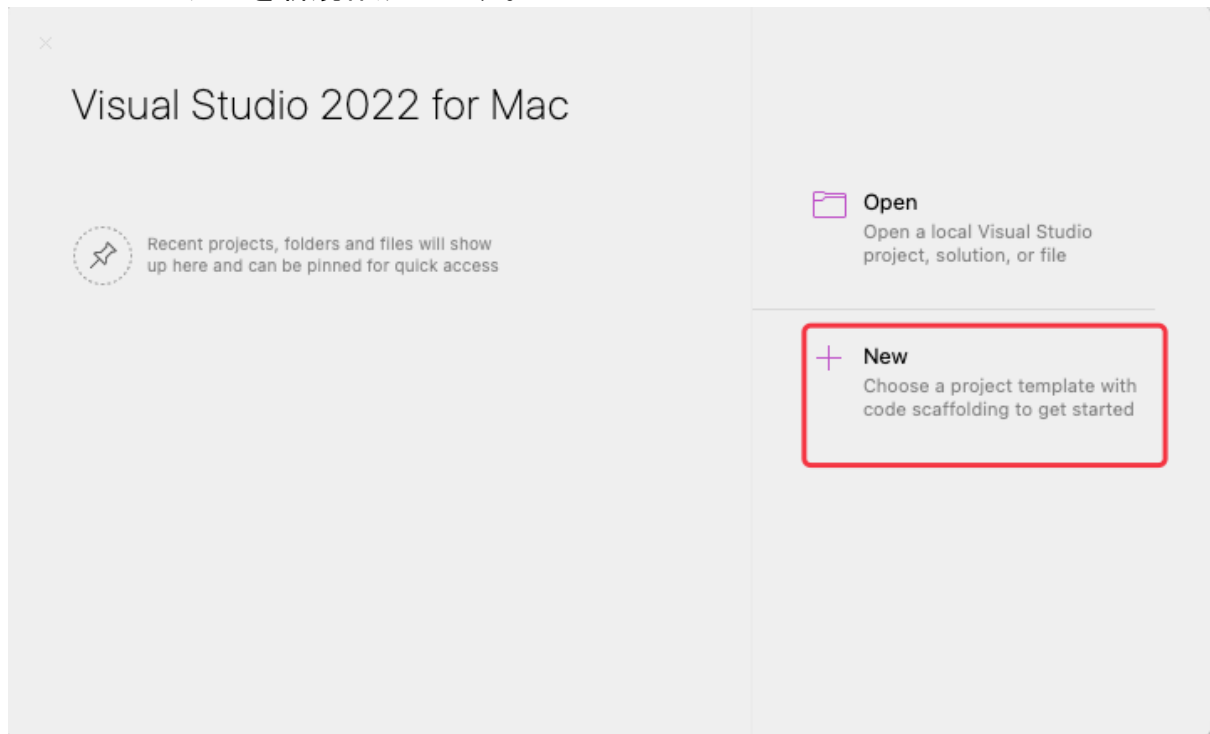
Android Studio Giraffe | 2022.3.1 Patch 4

Build #AI-223.8836.35.2231.11090377, built on November 14, 2023

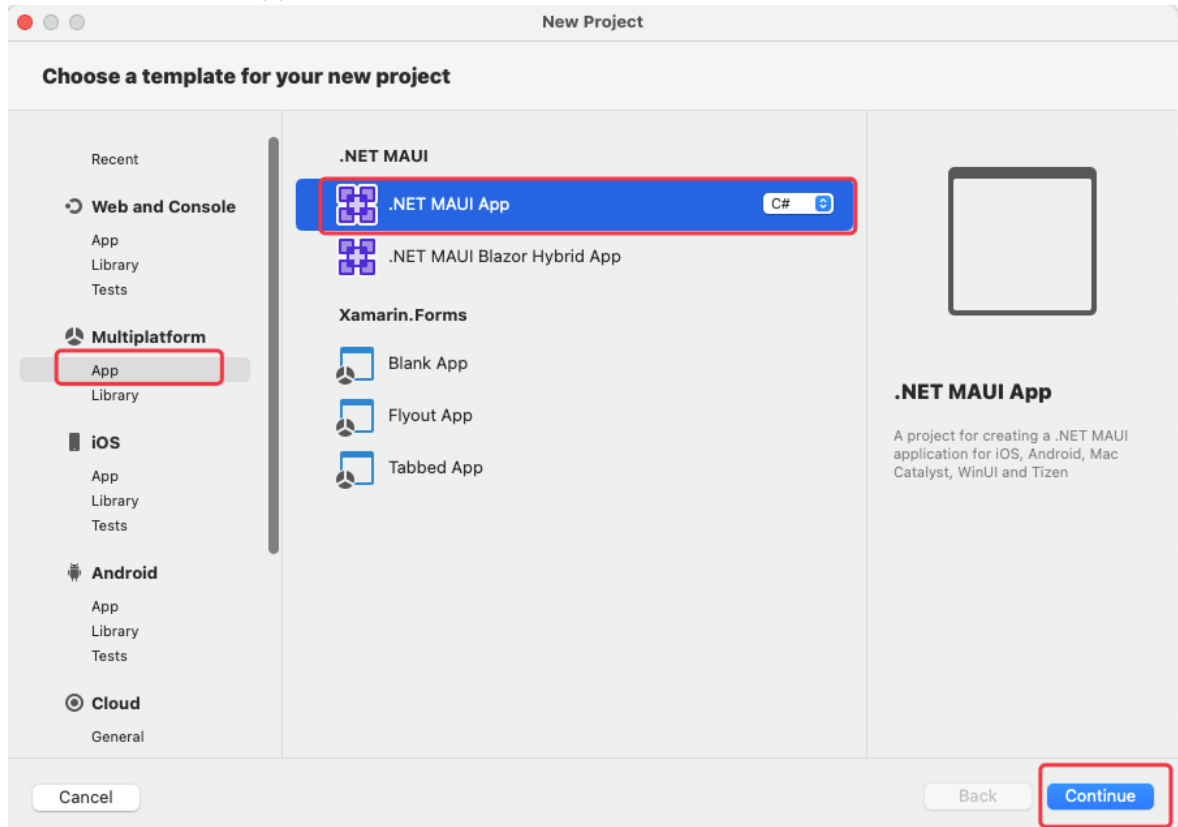
1 SDK の追加

1.1 SDK をインポートする

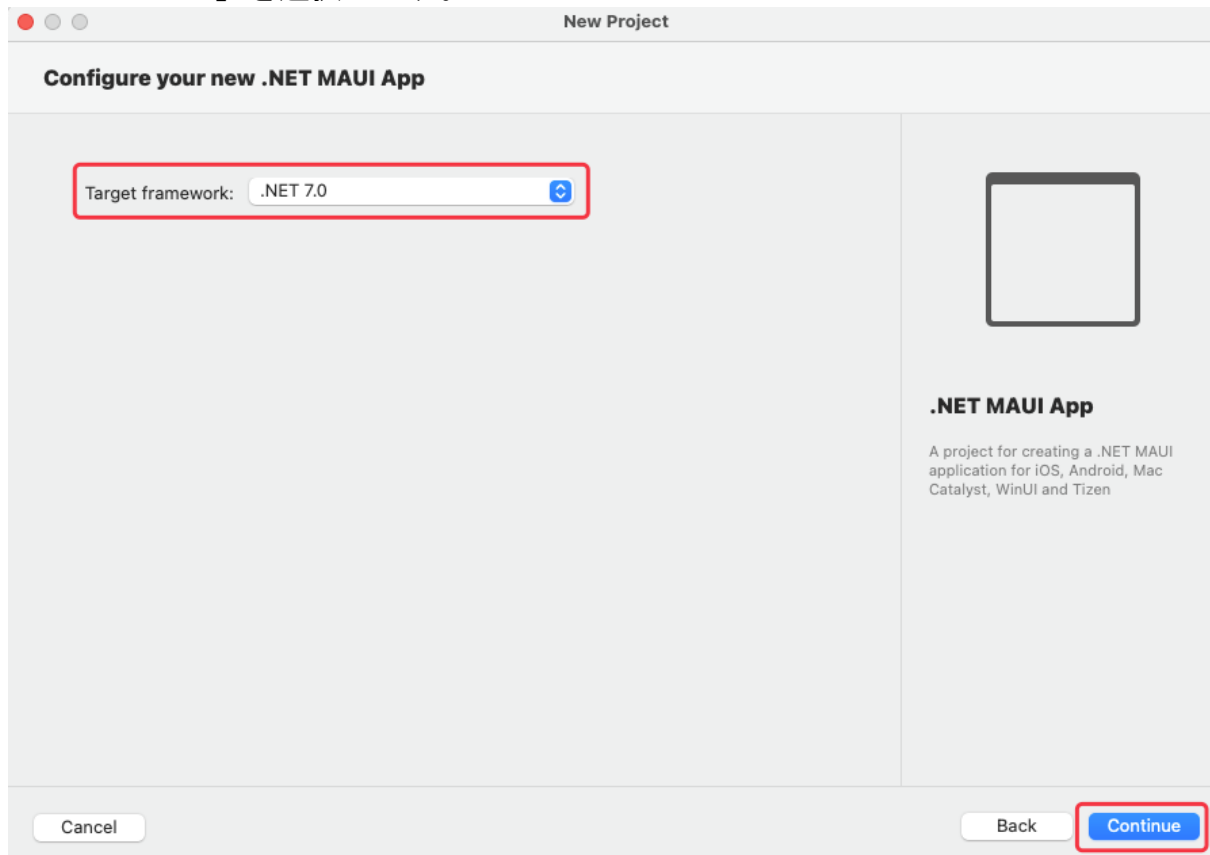
1. AsReaderSLIMAndroidSDK をローカルにダウンロードします。
2. プロジェクトを新規作成します。



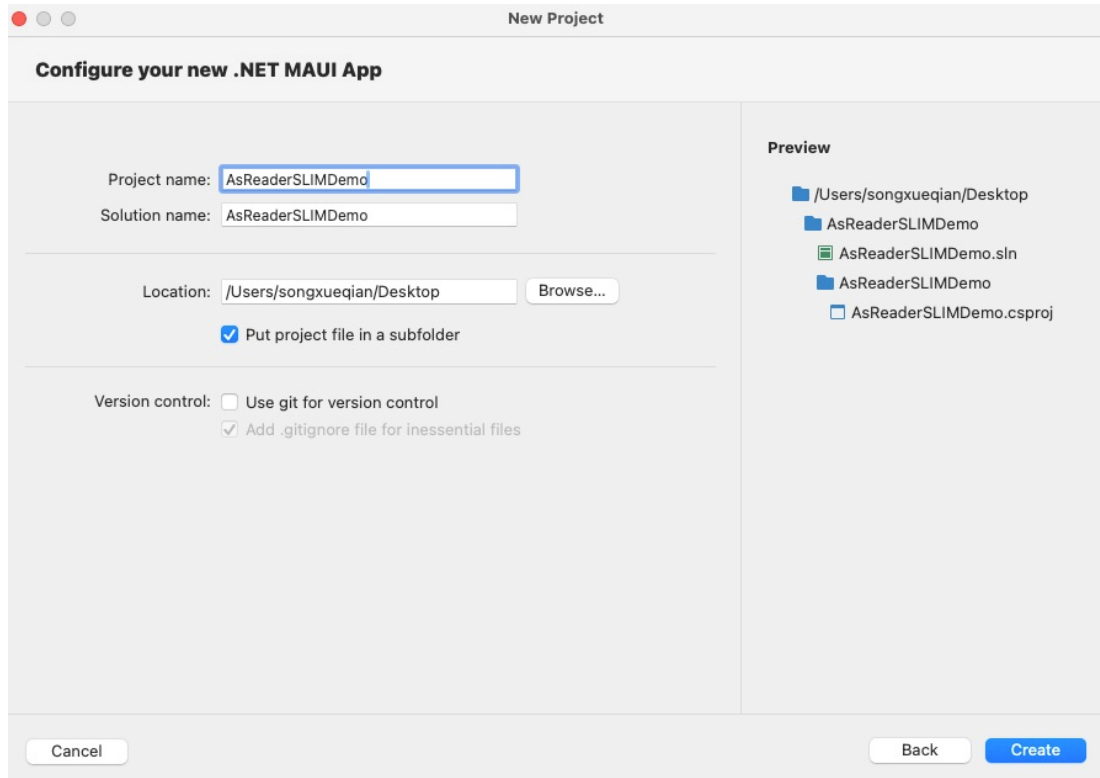
3. 「.NET MAUI App」を選択します。



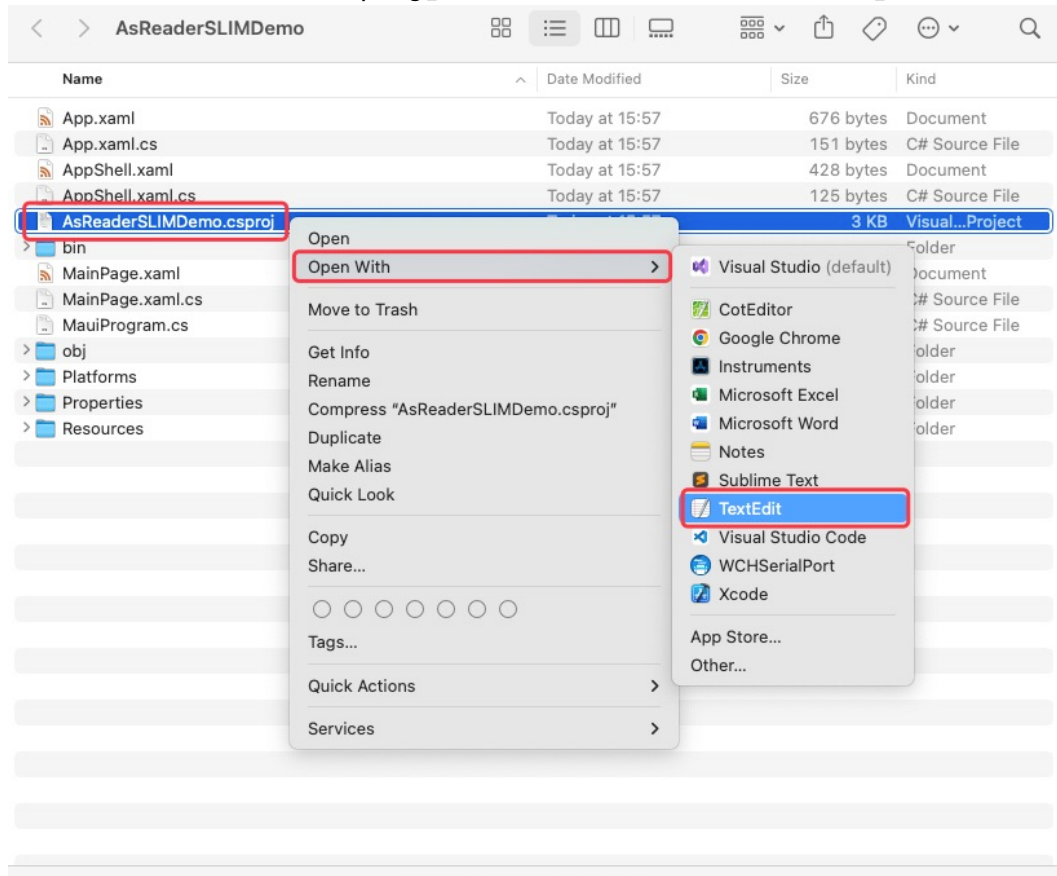
4. 「.NET 7.0」を選択します。



5. アプリ名とロケーションを作成します。

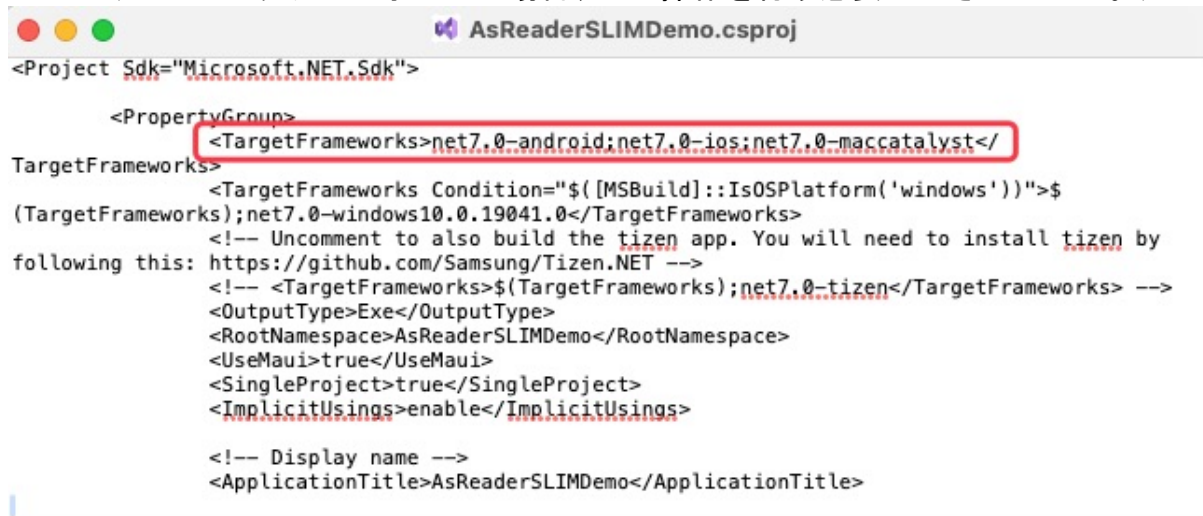


6. AsReaderSLIMiOSDemo.csproj」を右クリックし、「TextEdit」を開きます。



7. 下図赤枠内の「<Target Frameworks>」で、使用しないフレームを削除します。

(マルチプラットフォームの場合、7の操作を行う必要はありません。)

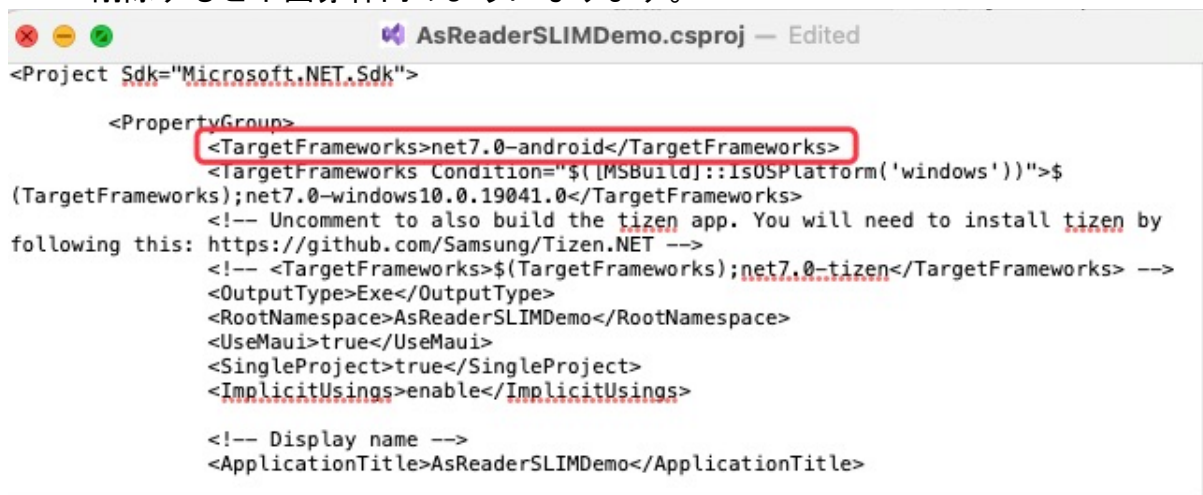


```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <TargetFrameworks>net7.0-android;net7.0-ios;net7.0-maccatalyst</
TargetFrameworks>
    <TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows'))">$
(TargetFrameworks);net7.0-windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by
following this: https://github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderSLIMDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderSLIMDemo</ApplicationTitle>
```

8. 削除すると下図赤枠内ようになります。

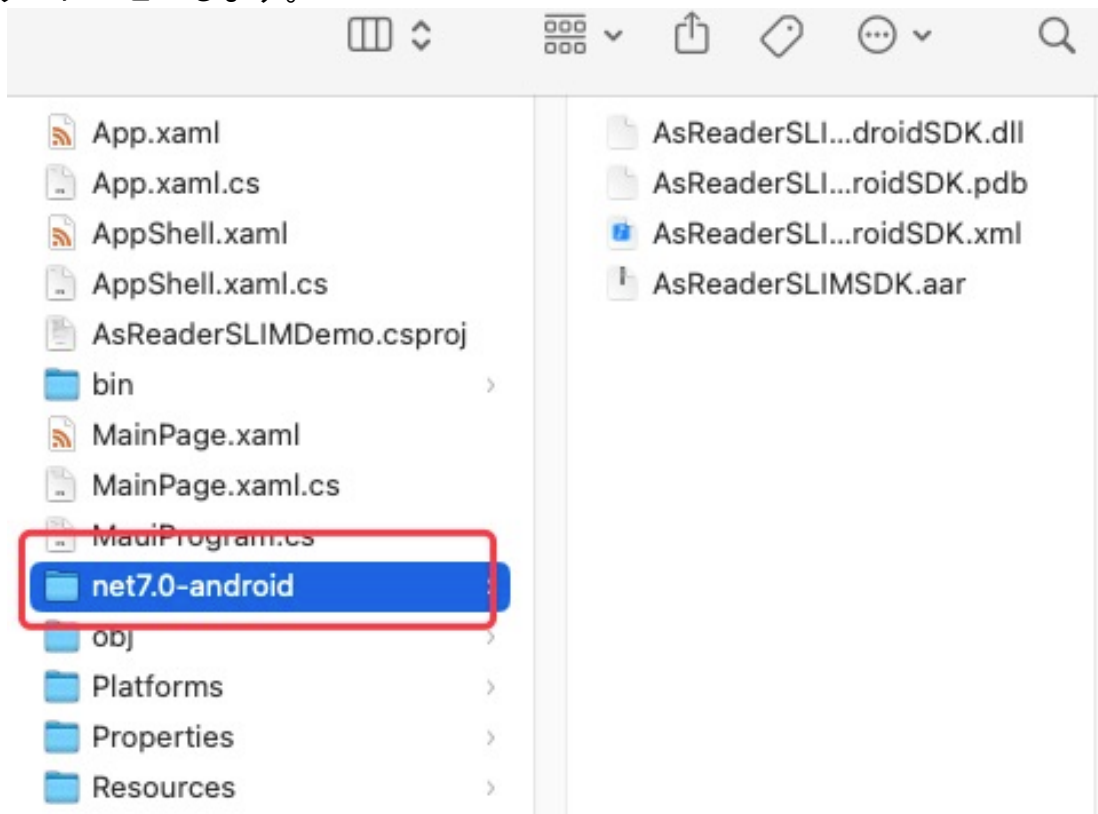


```
<Project Sdk="Microsoft.NET.Sdk">

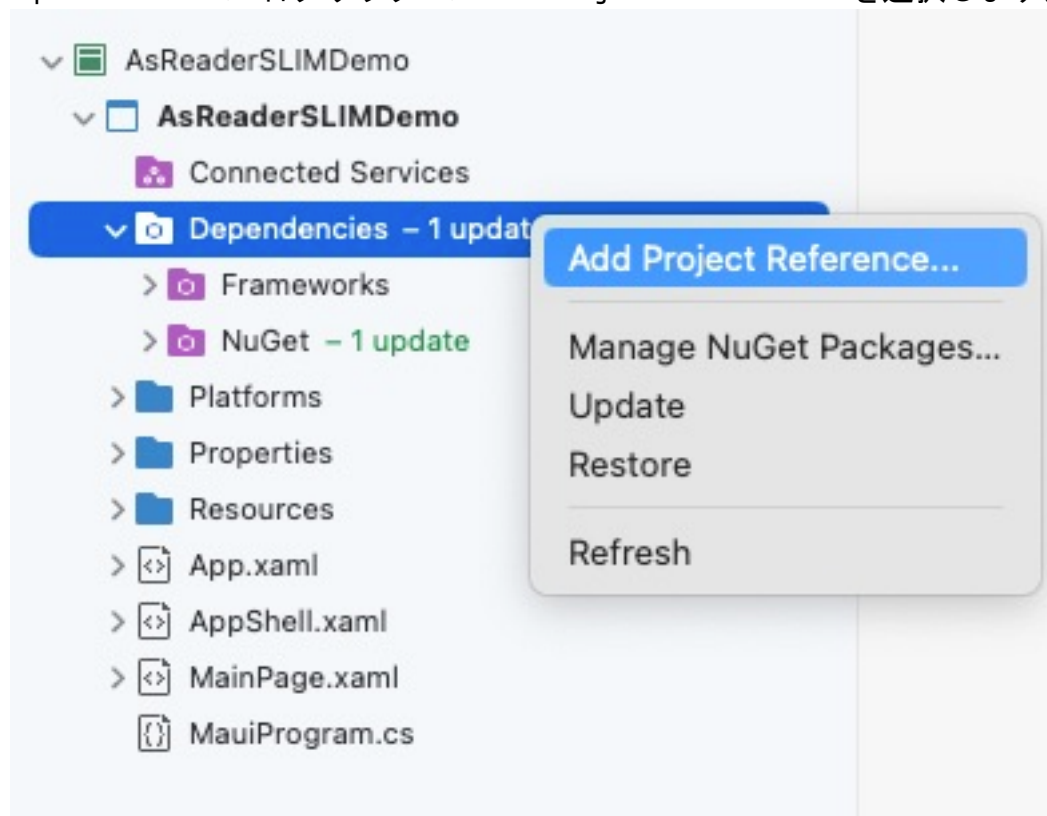
  <PropertyGroup>
    <TargetFrameworks>net7.0-android</TargetFrameworks>
    <TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows'))">$
(TargetFrameworks);net7.0-windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by
following this: https://github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderSLIMDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderSLIMDemo</ApplicationTitle>
```

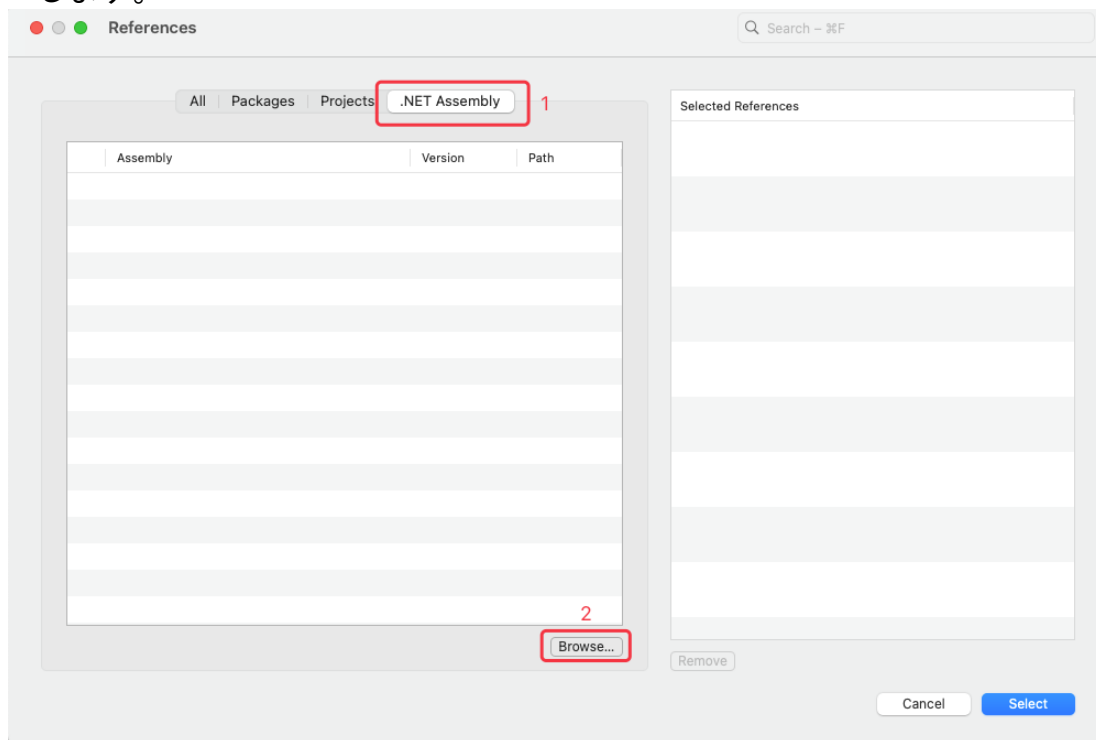
9. ローカルにダウンロードした AsReaderSLIMAndroidSDK をプロジェクトフォルダーにコピーします。



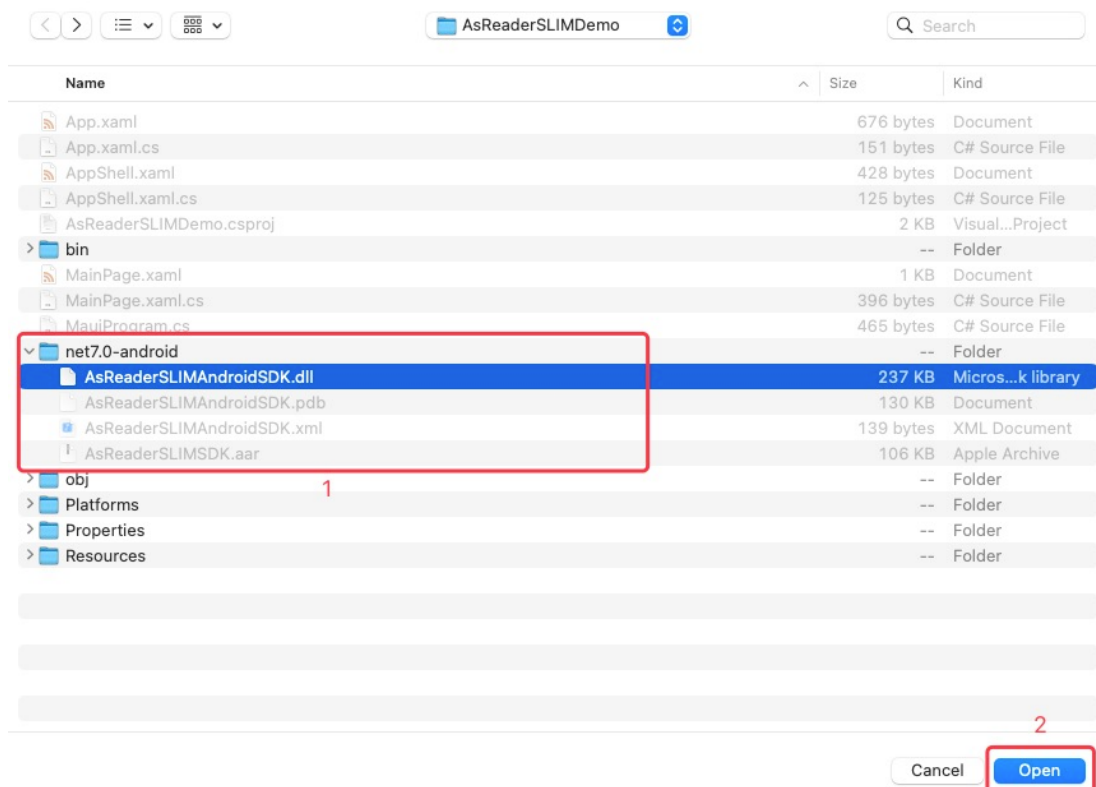
10. Dependencies -> 右クリック -> Add Project Reference...を選択します。



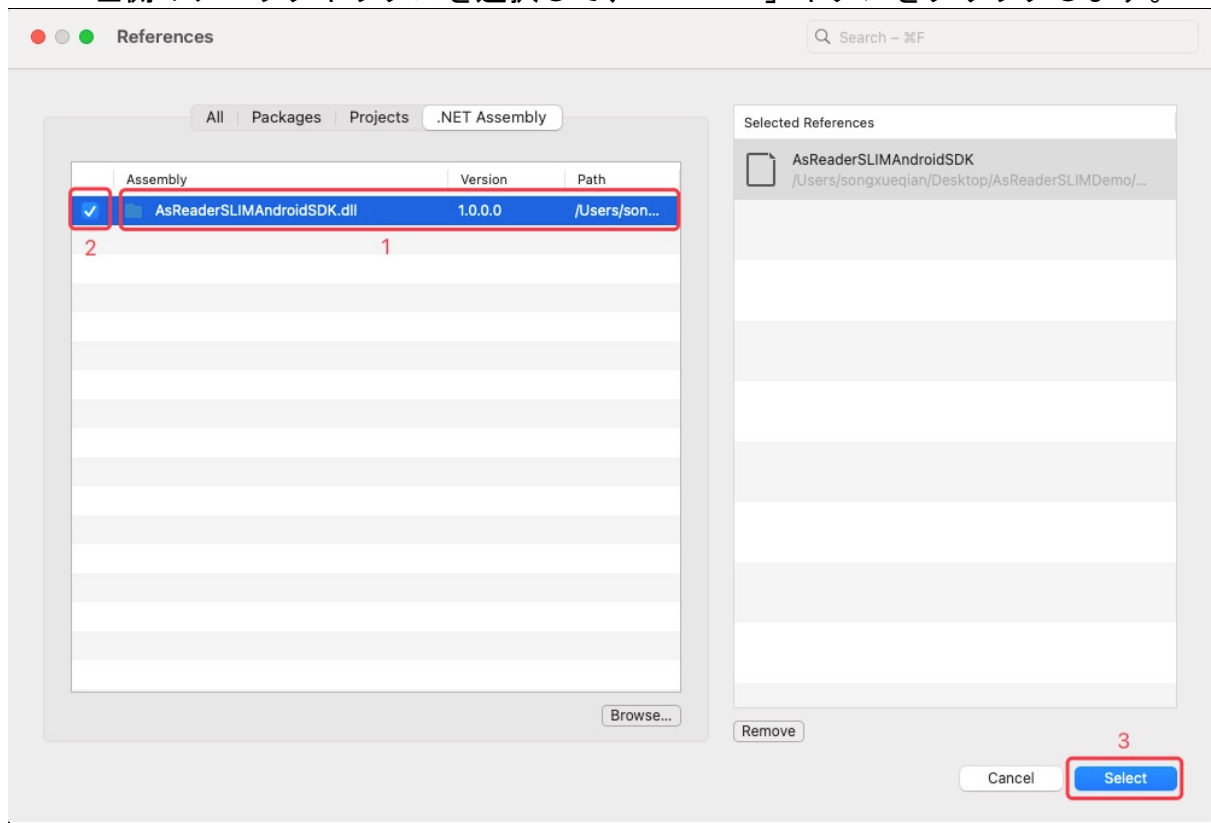
11. その後「.NET Assembly」（下図 1）を選び、「Browse…」（下図 2）を選択します。



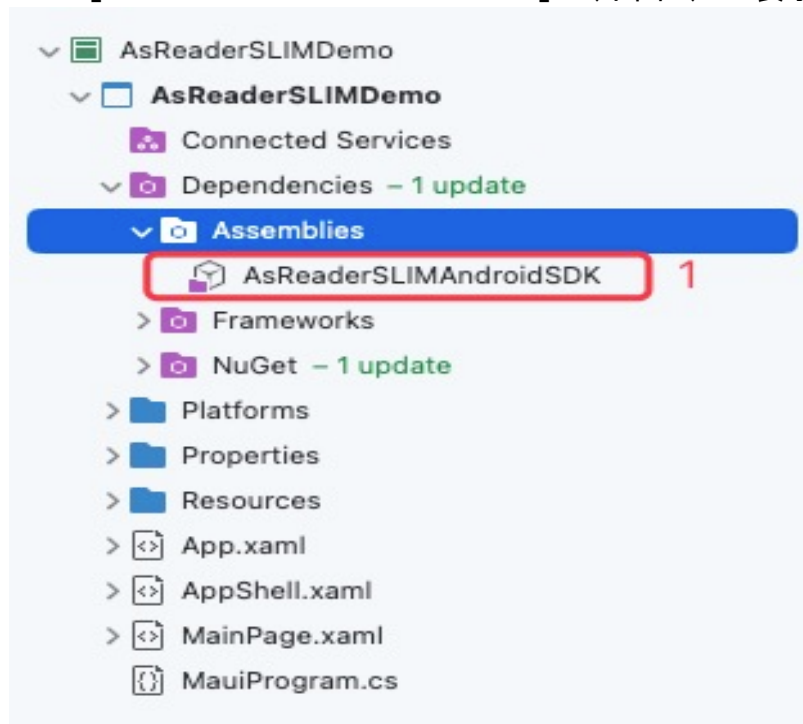
12. 「AsReaderSLIMAndroidSDK.dll」（下図 1）を選び、「Open」ボタン（下図 2）をクリックします。



13. 「References」ウィンドウのリストにある「AsReaderSLIMAndroidSDK.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. 「Assemblies」に「AsReaderSLIMAndroidSDK」（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、下記内容を追記して下さい。

```
<ItemGroup Condition="'$(TargetFramework)' == 'net7.0-android'">
  <Reference Include="AsReaderSLIMAndroidSDK">
    <HintPath>net7.0-android\AsReaderSLIMAndroidSDK.dll</HintPath>
  </Reference>
</ItemGroup>

</Project>
```

1.2 権限の追加

AndroidManifest.xml に権限を追加します。

```
AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.asreader.slim.maui.android.demo" android:versionCode="1" android:versionName="1.0.0">
  <application android:allowBackup="true" android:supportsRtl="true" android:label="AsReaderSLIMDemo"></application>
  <uses-feature android:name="android.hardware.usb.host" />
  <uses-permission android:name="android.permission.WAKE_LOCK" />
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
  <uses-permission android:name="android.permission.READ_MEDIA_VIDEO" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-sdk android:minSdkVersion="29" android:targetSdkVersion="34" />
</manifest>
```

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。

```
#if ANDROID
using AsReaderSLIMAndroidSDK;
using Com.Asreader.Slim;
using Com.Asreader.Slim.Barcode;
using Com.Asreader.Slim.Device;
using Com.Asreader.Slim.Device.Type;
using Com.Asreader.Slim.Barcode.Type;
using Com.Asreader.Slim.Barcode.Event;
#endif
```

2.2 「MyAsReaderDeviceDelegate」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

※Platforms の Android フォルダに作成することをお勧めします。

```
public class MyAsReaderDeviceDelegate : Java.Lang.Object, IAsReaderSLIMEventListener
{
    public void OnActionChanged(AsReaderSLIM p0, AsReaderSLIMActionState p1)
    {
    }

    public bool OnKeyEvent(AsReaderSLIM p0, AsReaderSLIMKeyState p1)
    {
        return false;
    }

    public void OnReadBarcode(AsReaderSLIM p0, AsReaderSLIMBarcodeType p1, string p2)
    {
    }

    public void OnReadBarcodeData(AsReaderSLIM p0, AsReaderSLIMBarcodeType p1, byte[] p2)
    {
    }

    public void OnStateChanged(AsReaderSLIM p0, AsReaderSLIMConnectionState p1)
    {
    }
}
```

2.3 AsReaderManager オブジェクトを初期化

```
AsReaderSLIM mReader = AsReaderSLIMManager.Instance;
MyAsReaderDeviceDelegate myAsReaderDeviceDelegate = new MyAsReaderDeviceDelegate();
mReader.SetEventListener(myAsReaderDeviceDelegate);
AsReaderSLIMDeviceUsbCdc usbCdc = new AsReaderSLIMDeviceUsbCdc(MainActivity.Instance);
```

2.4 OnStateChanged を実現

```
public void OnStateChanged(AsReaderSLIM p0, AsReaderSLIMConnectionState p1)
{
    if (p1 == AsReaderSLIMConnectionState.Connected)
    {
        //Connection successful
    }
    else if (p1 == AsReaderSLIMConnectionState.Disconnected)
    {
        //Disconnect
    }
}
```

2.5 OnReadBarcodeData を実現

```
public void OnReadBarcodeData(AsReaderSLIM p0, AsReaderSLIMBarcodeType p1, byte[] p2)
{
}
```

2.6 USB 接続

```
mDevice = AsReaderSLIMManager.Instance.CurrentDevice;

if (mDevice == null)
{
    mDevice = usbCdc;
}
mReader.ConnectDevice(mDevice);
```

mDevice : AsReaderSLIMDevice クラスのインスタンスオブジェクト

mReader : AsReaderSLIM クラスのインスタンスオブジェクト

usbCdc : AsReaderSLIMDeviceUsbCdc クラスのインスタンスオブジェクト

2.7 Barcode スキャン開始

```
mReader.StartDecode();
```

2.8 Barcode スキャン停止

```
mReader.StopDecode();
```

2.9 切断

```
if (mDevice != null)
{
    mReader.DisconnectDevice();
}
```

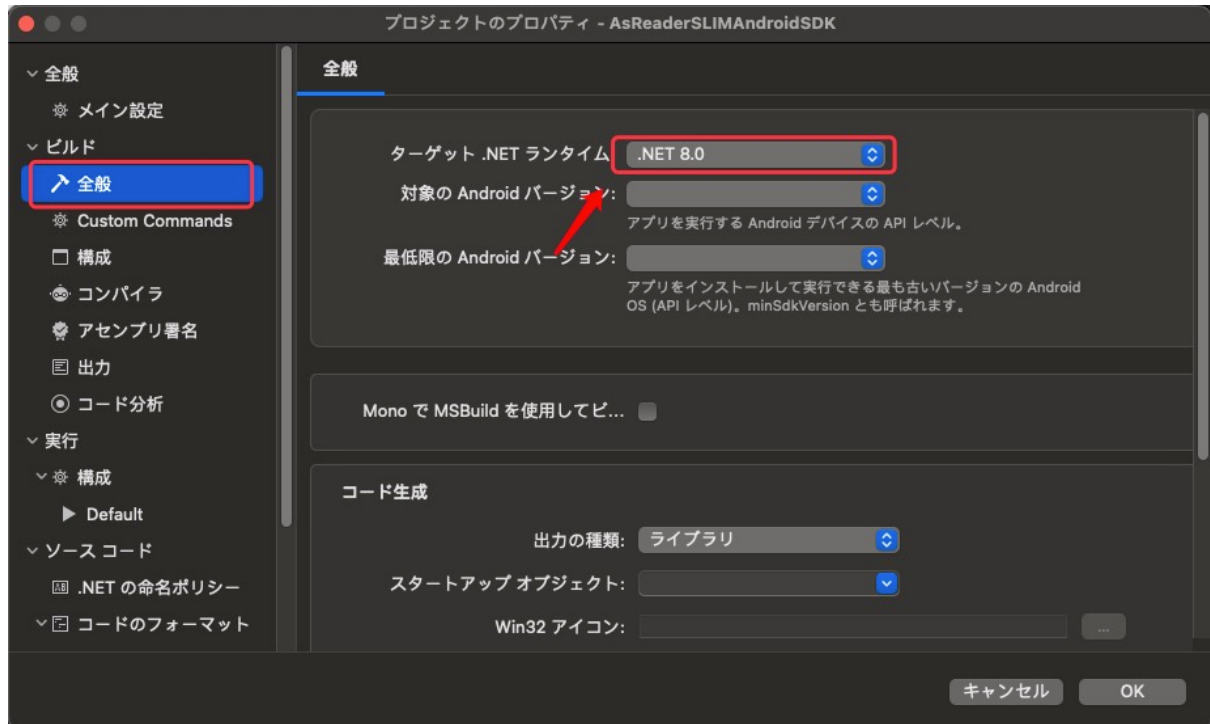
3 ビルドする.NET バージョンの変更

「.NET 7.0」を「.NET 8.0」に変更する方法です。

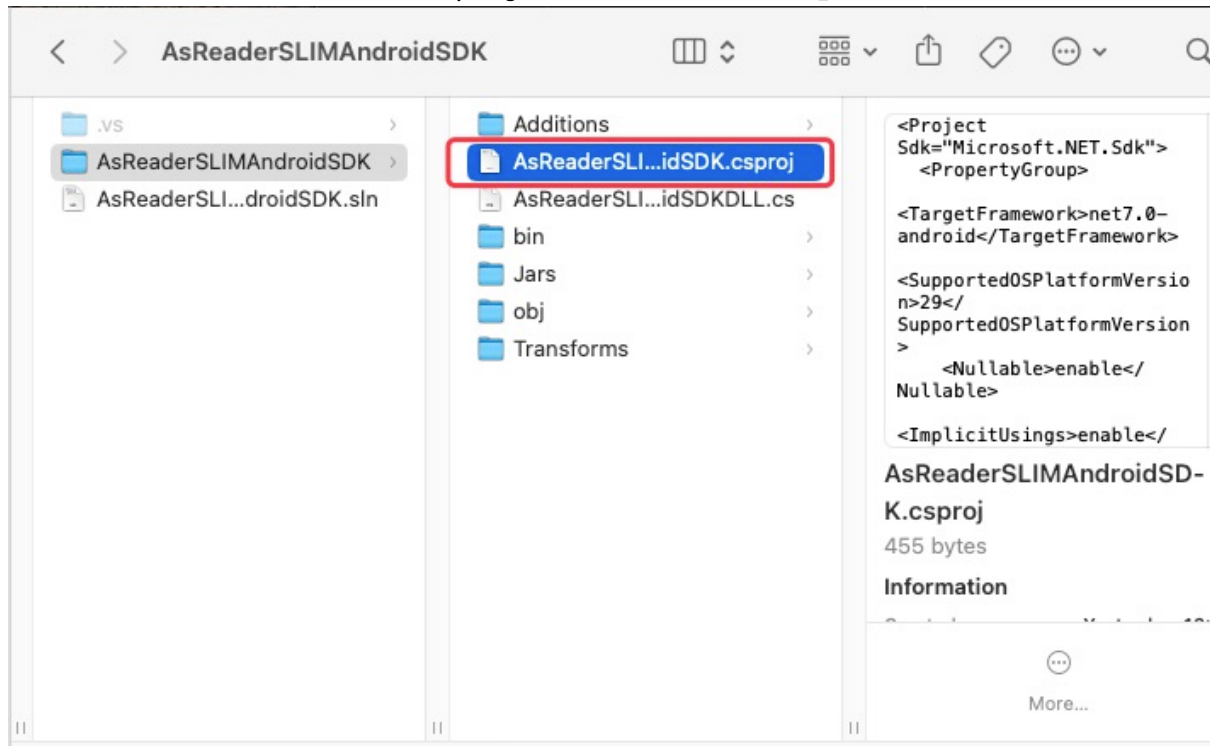
1. プロジェクトを右クリック→プロパティ



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



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



4. 以下の修正を行い、ファイルを保存。



```

<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net7.0-android</TargetFramework>
    <SupportedOSPlatformVersion>21</SupportedOSPlatformVersion>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>
</Project>

```

↓

```

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

```

以下の配置を追加します。

```

<UseMaui>true</UseMaui>
<MauiVersion>8.0.3</MauiVersion>
<SkipValidateMauiImplicitPackageReferences>true</SkipValidateMauiImplicitPackageReferences>

```

5. 改めてプロジェクトをローディングします。

※ビルドした時に作成されるアプリは使用したバージョンツールでのみ使用できます。

例えば、「.NET 8.0」でビルドした場合、そのツールでのみご使用でき、「.NET 7.0」や「.NET 9.0」ではご使用できません。

4 API の使用

API の使用について、Android の SDK リファレンスガイドを参照してください。

https://asreader.jp/download/asr-a25s_slim/sdk_guide/android/index.html