



.NET MAUI Android SDK
For ASR-P252B

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 | 2024/09/17 |

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 「AsReaderP252BBluetoothDiscoveryEventListener」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 「AsReaderP252BEventListener」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 | 13 |
| 2.4 AsReaderManager オブジェクトを初期化 | 13 |
| 2.5 OnReceivedDevice を実現 | 13 |
| 2.6 OnFoundDeviceFinished を実現 | 14 |
| 2.7 OnStateChanged を実現 | 14 |
| 2.8 OnReadBarcodeData を実現 | 14 |
| 2.9 OnReadTag を実現 | 14 |
| 2.10 接続 | 14 |
| 2.11 Bluetooth 検索 | 14 |
| 2.12 Bluetooth での接続 | 15 |
| 2.13 Barcode スキャン開始 | 15 |
| 2.14 Barcode スキャン停止 | 15 |
| 2.15 RFID スキャン開始 | 15 |
| 2.16 RFID スキャン停止 | 15 |
| 2.17 切断 | 15 |
| 3 ビルドする .NET バージョンの変更 | 16 |
| 4 API の使用 | 19 |

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 17.6.7 (build 417)
- ・ Android Studio :

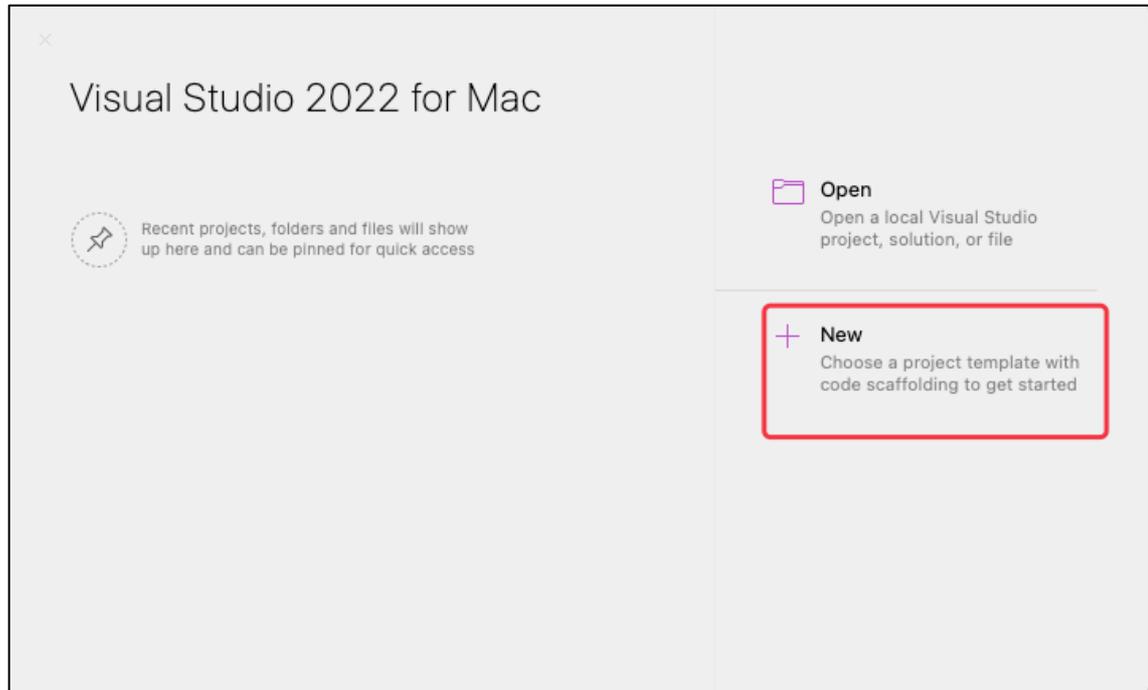
Android Studio Giraffe | 2022.3.1 Patch 4

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

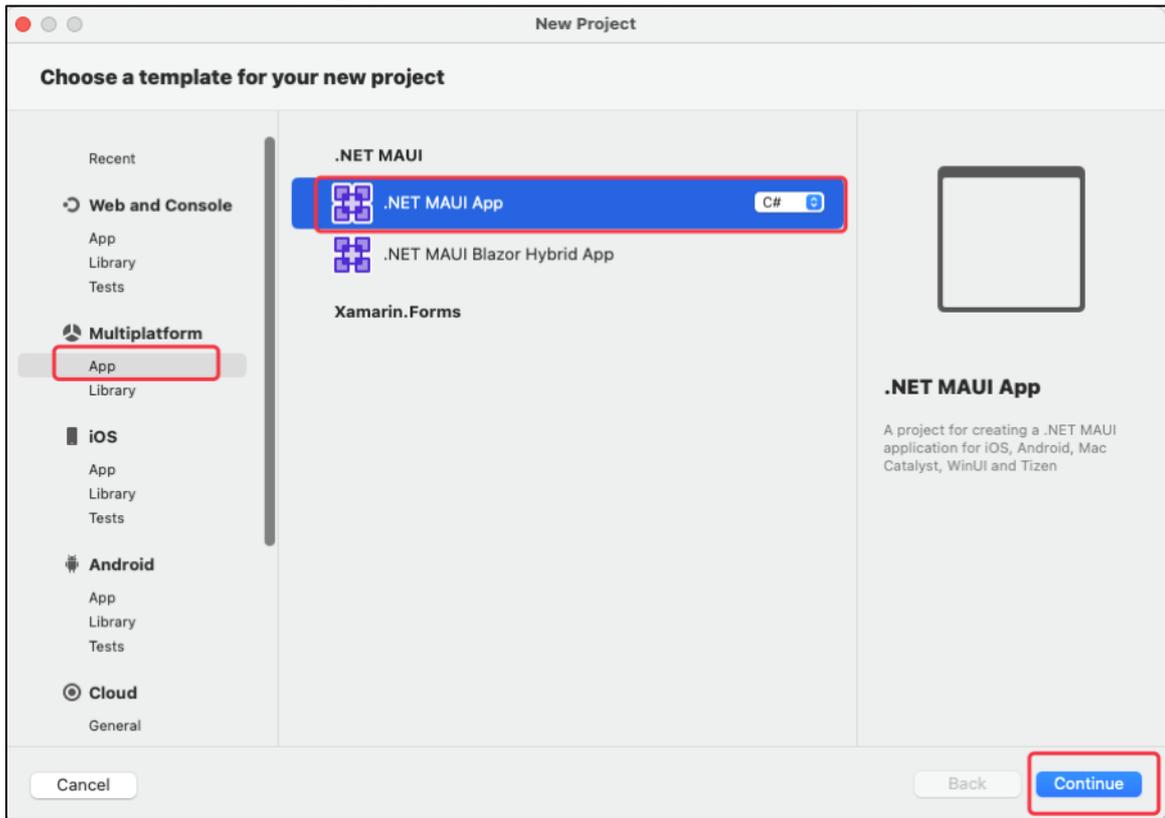
1 SDK の追加

1.1 SDK をインポートする

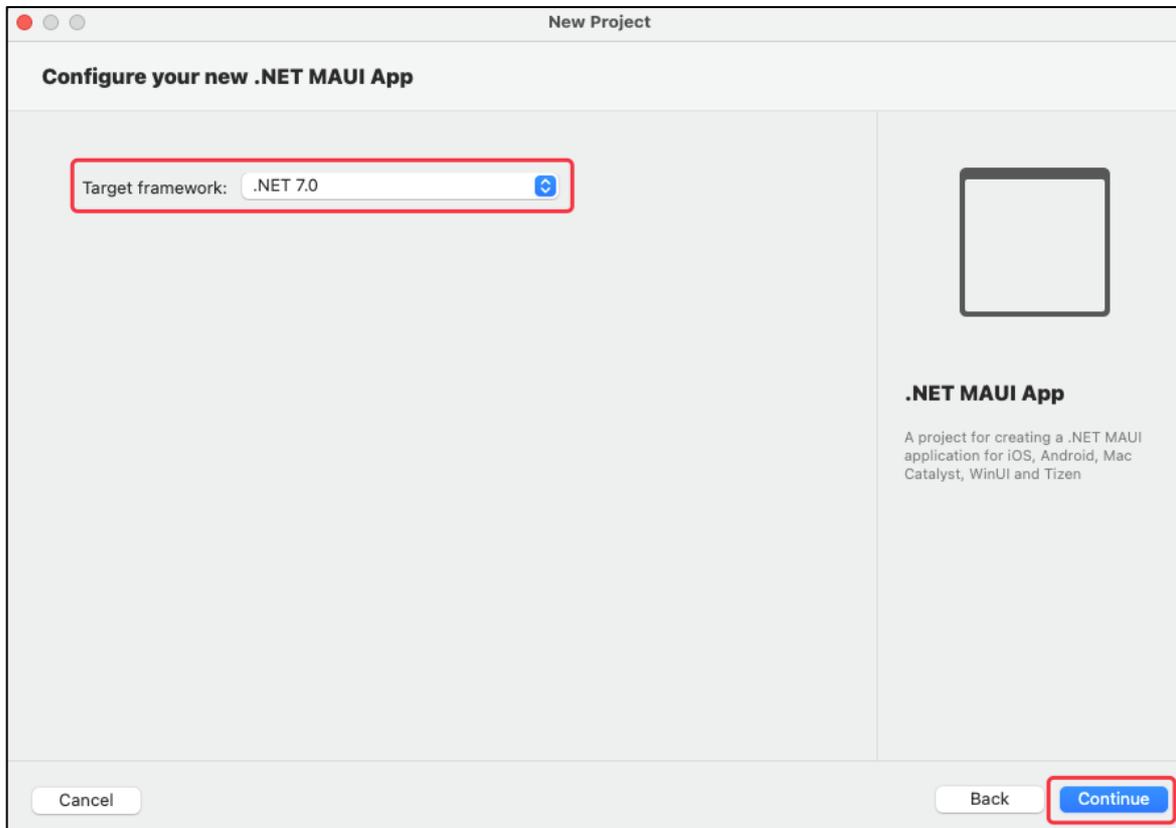
1. AsReaderP252BSDK をローカルにダウンロードします。
2. プロジェクトを新規作成します。



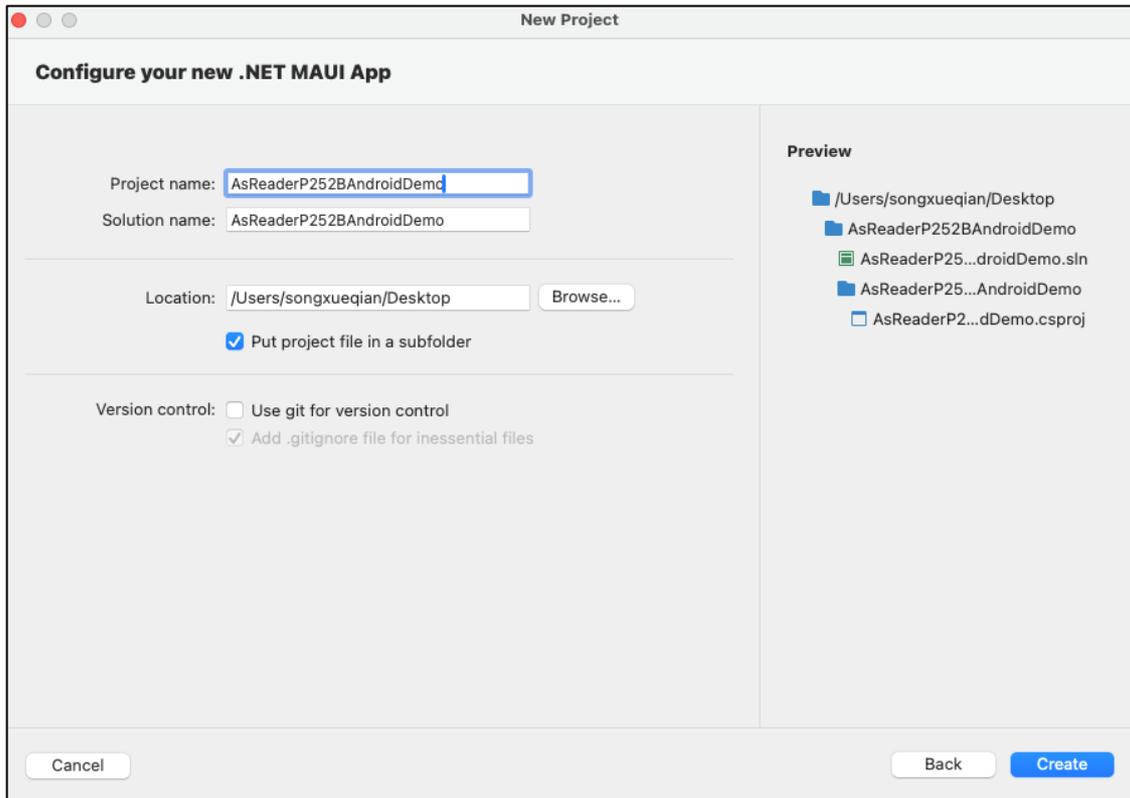
3. 「.NET MAUI App」を選択します。



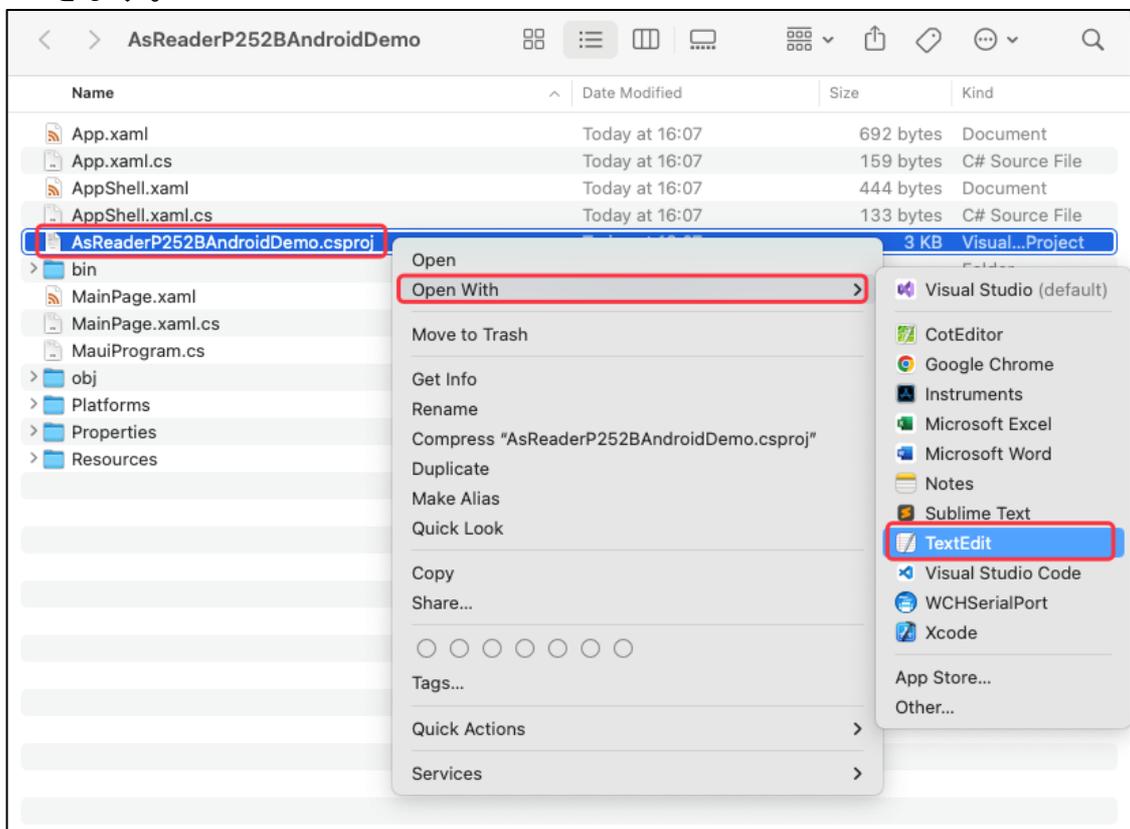
4. 「.NET 7.0」を選択します。



5. アプリ名とロケーションを作成します。



6. 「AsReaderP252BAndroidDemo.csproj」を右クリックして、「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ございません。)

```

<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>AsReaderP252BAndroidDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderP252BAndroidDemo</ApplicationTitle>
  
```

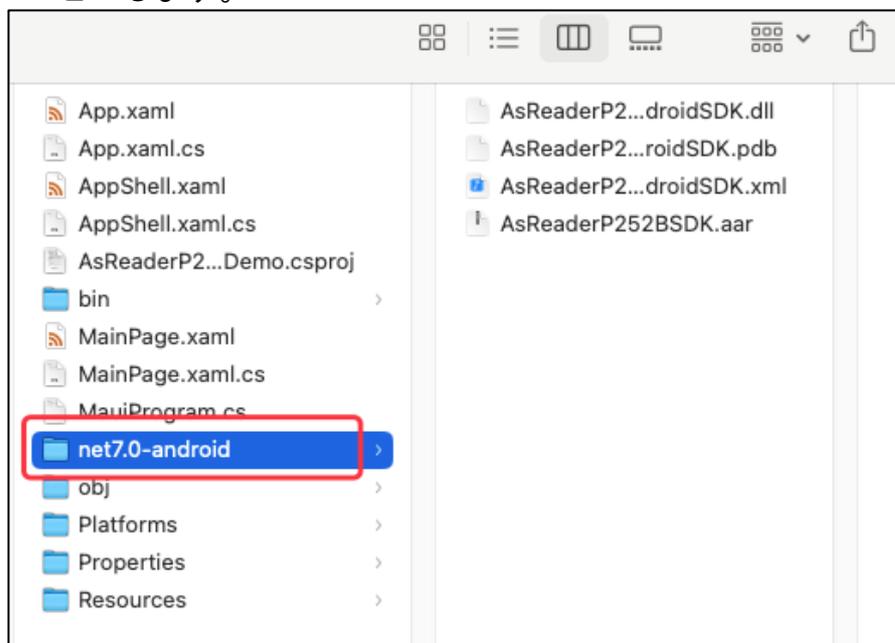
8. 削除すると、下図の赤枠内ようになります。

```

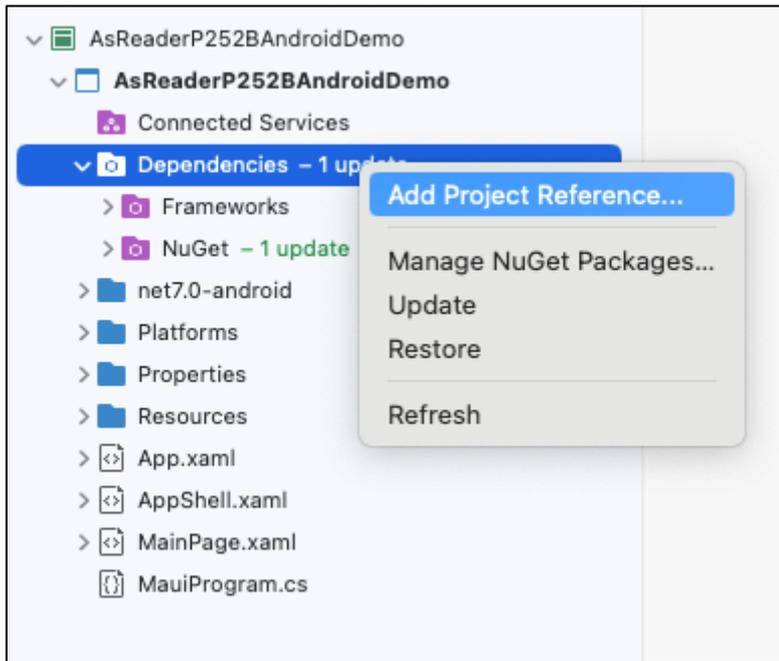
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net7.0-android</TargetFrameworks>
    <TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows'))">$(TargetFrameworks);net7.0-
windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/
Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderP252BAndroidDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderP252BAndroidDemo</ApplicationTitle>
  
```

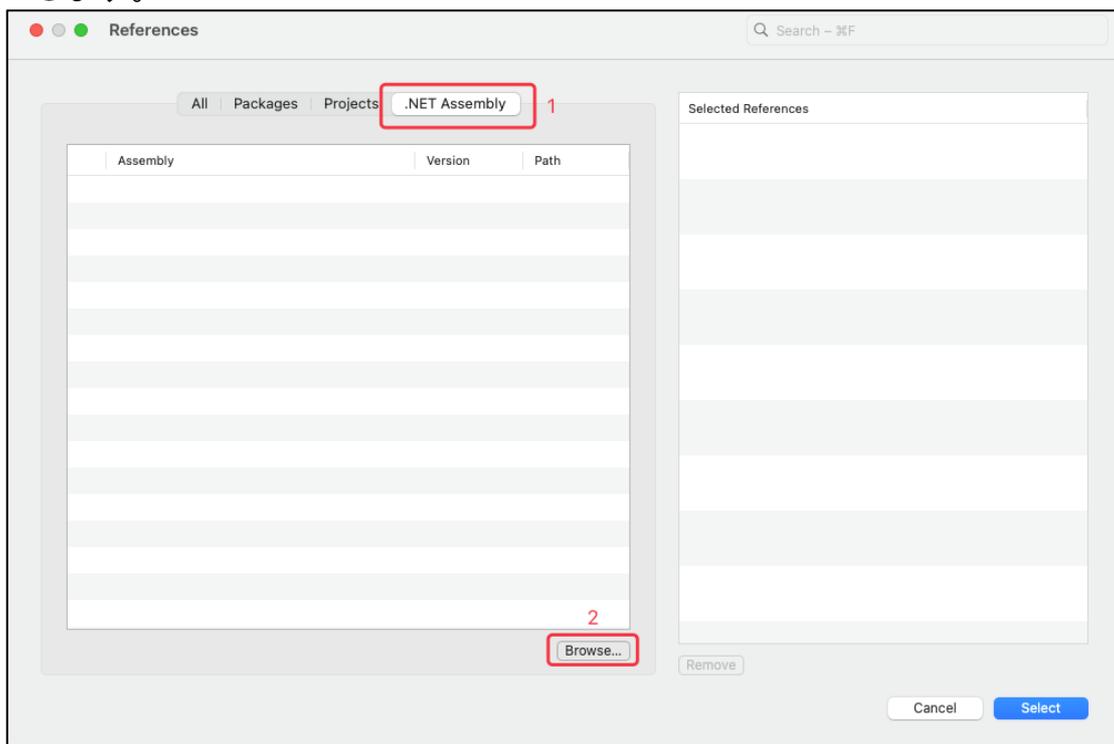
9. ローカルにダウンロードした AsReaderP252BAndroidSDK をプロジェクトフォルダーにコピーします。



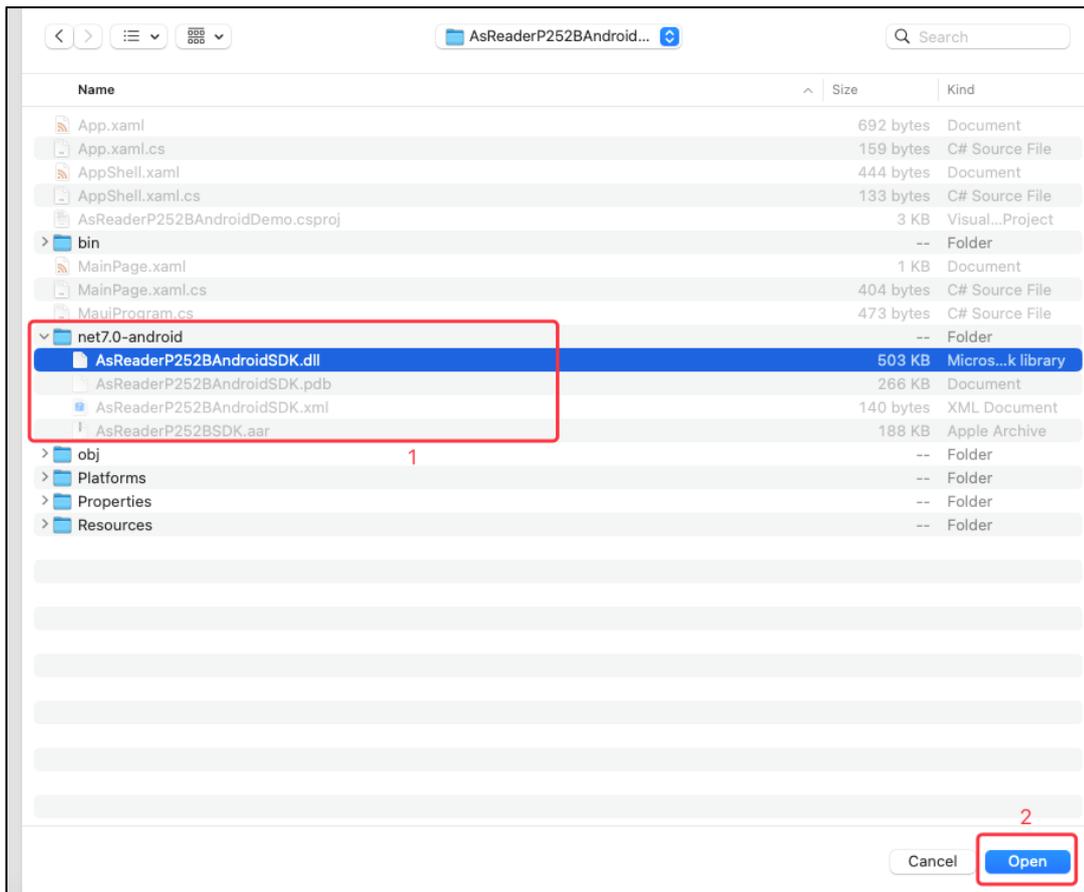
10. Dependencies -> 右クリック -> Add Project Reference...を選択します。



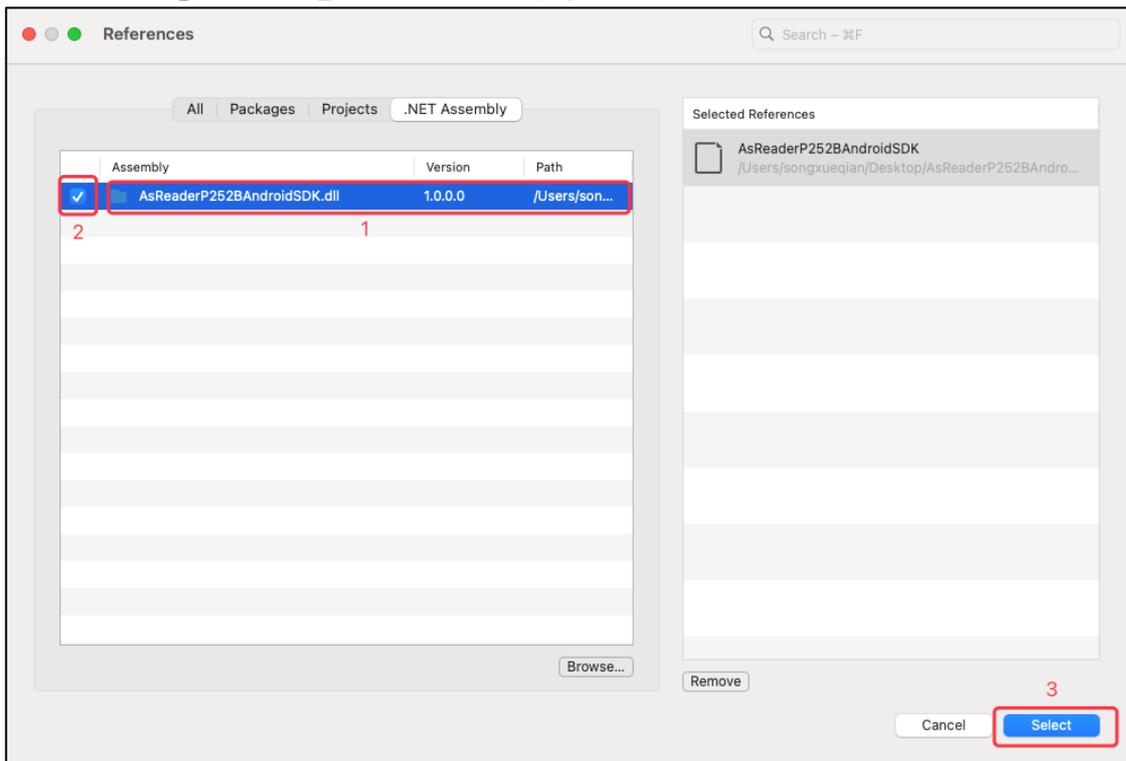
11. その後「.NET Assembly」（下図1）を選び、「Browse...」（下図2）を選択します。



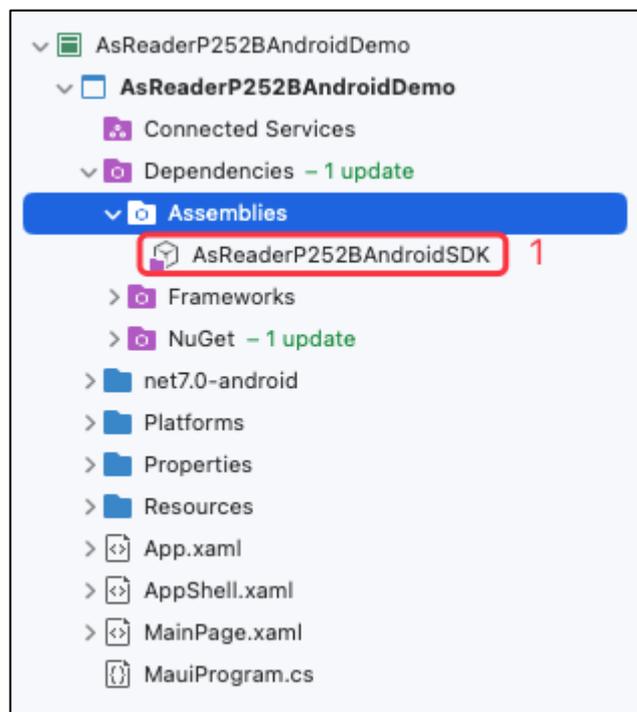
12. 「AsReaderP252BAndroidSDK.dll」 (下図 1) を選択して、「Open」ボタン (下図 2) をクリックします。



13. 「References」 ウィンドウ内のリストにある「AsReaderP252BAndroidSDK.dll」左側のチェックボックスを選び、「Select」ボタンをクリックします。



14. 「Assemblies」に「AsReaderP252BAndroidSDK」（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。

```

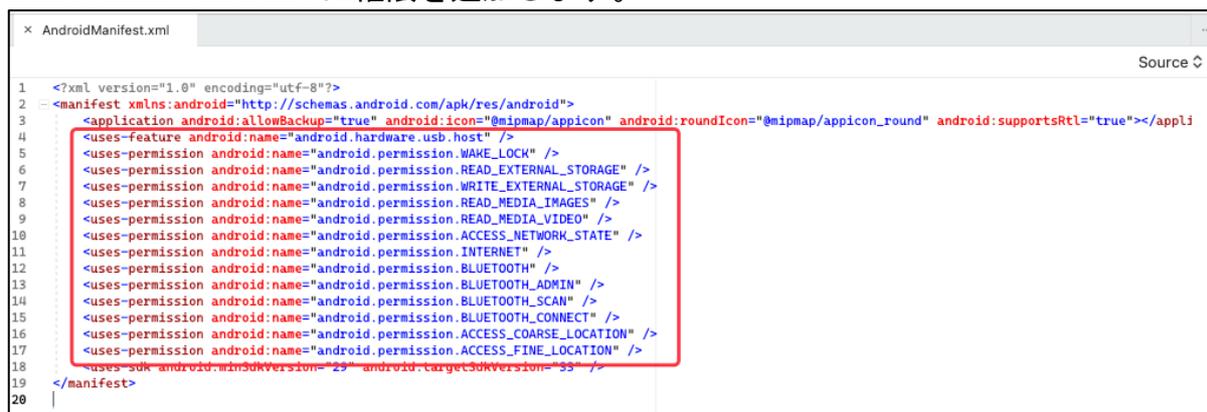
<ItemGroup Condition="'$(TargetFramework)' == 'net7.0-android'">
  <Reference Include="AsReaderP252BAndroidSDK">
    <HintPath>net7.0-android\AsReaderP252BAndroidSDK.dll</HintPath>
  </Reference>
</ItemGroup>

</Project>

```

1.2 権限の追加

AndroidManifest.xml に権限を追加します。



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
3   <application android:allowBackup="true" android:icon="@mipmap/appicon" android:roundIcon="@mipmap/appicon_round" android:supportsRtl="true"></appli
4   <uses-feature android:name="android.hardware.usb.host" />
5   <uses-permission android:name="android.permission.WAKE_LOCK" />
6   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
7   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
8   <uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
9   <uses-permission android:name="android.permission.READ_MEDIA_VIDEO" />
10  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
11  <uses-permission android:name="android.permission.INTERNET" />
12  <uses-permission android:name="android.permission.BLUETOOTH" />
13  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
14  <uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
15  <uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
16  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
17  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
18  <uses-sdk android:minSdkVersion="25" android:targetSdkVersion="33" />
19 </manifest>
20

```

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 Com.Asreader.P252b;
using Com.Asreader.P252b.Rfid.Event;
using Com.Asreader.P252b.Rfid.Type;
using Com.Asreader.P252b.Device.Type;
using Com.Asreader.P252b.Barcode;
using Com.Asreader.P252b.Device;
using Com.Asreader.P252b.Rfid.Params;
using Com.Asreader.P252b.Device.BluetoothEvent;
using Android.Bluetooth;
#endif
```

2.2 「AsReaderP252BBluetoothDiscoveryEventListener」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 MyAsReaderBLEDeviceDelegate : Java.Lang.Object, IAsReaderP252BBluetoothDiscoveryEventListener
{
    public void OnFoundDeviceFinished()
    {
    }

    public void OnReceivedDevice(BluetoothDevice p0)
    {
    }
}
```

2.3 「AsReaderP252BEventListener」 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, IAsReaderP252BEventListener
{
    public void OnStateChanged(AsReaderP252B p0, AsReaderP252BConnectionState p1)
    {
    }

    public void OnReadTag(AsReaderP252B p0, AsReaderP252BActionState p1, string p2, float p3, float p4, float p5)
    {
    }

    public void OnReadBarcodeData(AsReaderP252B p0, AsReaderP252BBarcodeType p1, byte[] p2)
    {
    }

    public void OnAccessResult(AsReaderP252B p0, AsReaderP252BResultCode p1, AsReaderP252BActionState p2, string p3, string p4, float p5, float p6, float p7)
    {
    }

    public void OnActionChanged(AsReaderP252B p0, AsReaderP252BActionState p1)
    {
    }

    public bool OnKeyEvent(AsReaderP252B p0, AsReaderP252BKeyState p1)
    {
        return false;
    }

    public bool OnModeKeyEvent(AsReaderP252B p0, AsReaderP252BKeyState p1)
    {
        return false;
    }

    public void OnReadBarcode(AsReaderP252B p0, AsReaderP252BBarcodeType p1, string p2)
    {
    }

    public void OnReadHumidityTag(AsReaderP252B p0, AsReaderP252BActionState p1, string p2, float p3, float p4, float p5, float p6)
    {
    }

    public void OnReadTemperatureTag(AsReaderP252B p0, AsReaderP252BActionState p1, string p2, float p3, float p4, float p5, float p6)
    {
    }

    public void OnReceivedData(byte[] p0)
    {
    }
}
```

2.4 AsReaderManager オブジェクトを初期化

```
AsReaderP252B mReader = AsReaderP252BManager.Instance;
MyAsReaderDeviceDelegate myAsReaderDeviceDelegate = new MyAsReaderDeviceDelegate();
mReader.SetEventListener(myAsReaderDeviceDelegate);
```

```
AsReaderP252BDevice usbCdc = new AsReaderP252BDeviceUsbCdc(MainActivity.Instance);
AsReaderP252BDeviceBluetoothCdc deviceBluetoothCdc = new AsReaderP252BDeviceBluetoothCdc(MainActivity.Instance);
MyAsReaderBLEDeviceDelegate myAsReaderBLEDeviceDelegate = new MyAsReaderBLEDeviceDelegate();
deviceBluetoothCdc.SetBluetoothDiscoveryEventListener(myAsReaderBLEDeviceDelegate);
```

2.5 OnReceivedDevice を実現

```
public void OnReceivedDevice(BluetoothDevice p0)
{
}
}
```

2.6 OnFoundDeviceFinished を実現

```
public void OnFoundDeviceFinished()
{
}
}
```

2.7 OnStateChanged を実現

```
public void OnStateChanged(AsReaderP252B p0, AsReaderP252BConnectionState p1)
{
    if (p1 == AsReaderP252BConnectionState.Connected) {
        //Connection successful
    } else if (p1 == AsReaderP252BConnectionState.Disconnected)
    {
        //Disconnect
    }
}
```

2.8 OnReadBarcodeData を実現

```
public void OnReadBarcodeData(AsReaderP252B p0, AsReaderP252BBarcodeType p1, byte[] p2)
{
}
}
```

2.9 OnReadTag を実現

```
public void OnReadTag(AsReaderP252B p0, AsReaderP252BActionState p1, string p2, float p3, float p4, float p5)
{
}
}
```

2.10 接続

```
mDevice = AsReaderP252BManager.Instance.CurrentDevice;
if (mDevice == null)
{
    mDevice = usbCdc;
}
mReader.ConnectDevice(usbCdc);
```

mDevice : AsReaderP252BDevice クラスのインスタンスオブジェクト、

mReader : AsReaderP252B クラスのインスタンスオブジェクト

usbCdc : AsReaderP252BDevice クラスのインスタンスオブジェクト

2.11 Bluetooth 検索

```
deviceBluetoothCdc.StartDiscovery();
```

deviceBluetoothCdc : AsReaderP252BDevice クラスのインスタンスオブジェクト

2.12 Bluetooth での接続

```
if (deviceBluetoothCdc != null)
{
    mDevice = deviceBluetoothCdc;
    mDevice.Address = macAddress;
    mReader.ConnectDevice(mDevice);
}
```

macAddress (string 型) : AsReader の Mac アドレス

2.13 Barcode スキャン開始

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

2.14 Barcode スキャン停止

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

2.15 RFID スキャン開始

```
mReader.Inventory();
```

2.16 RFID スキャン停止

```
mReader.Stop();
```

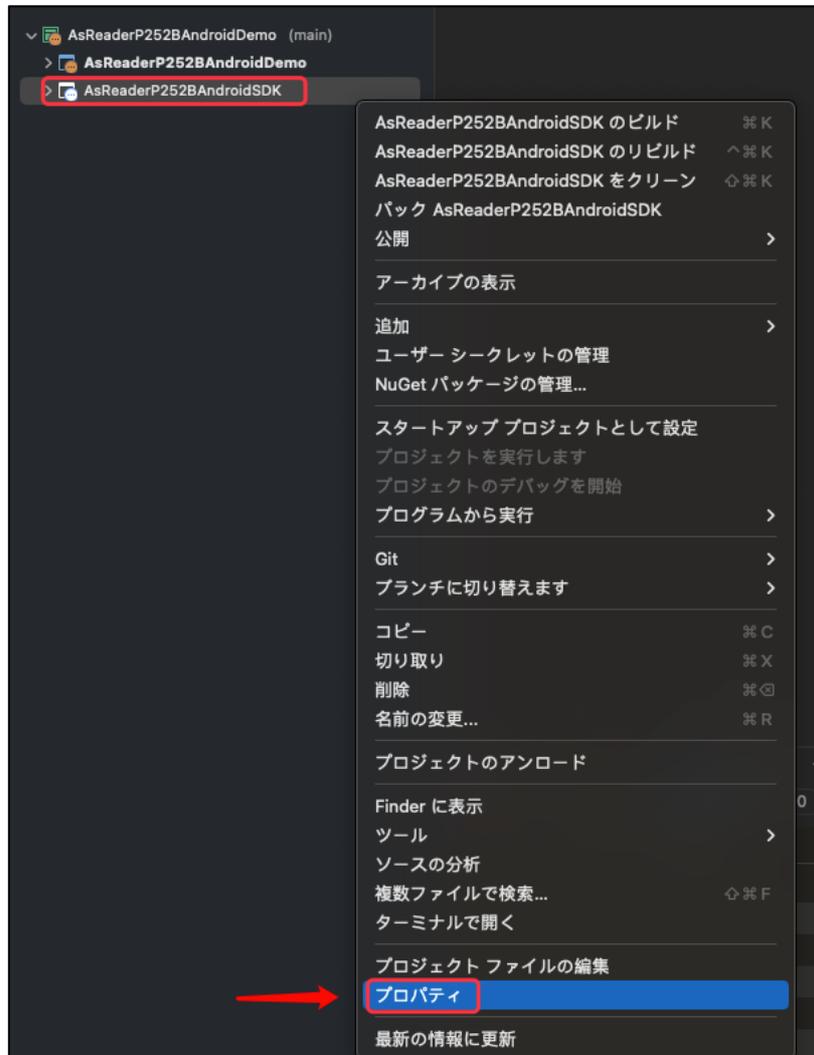
2.17 切断

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

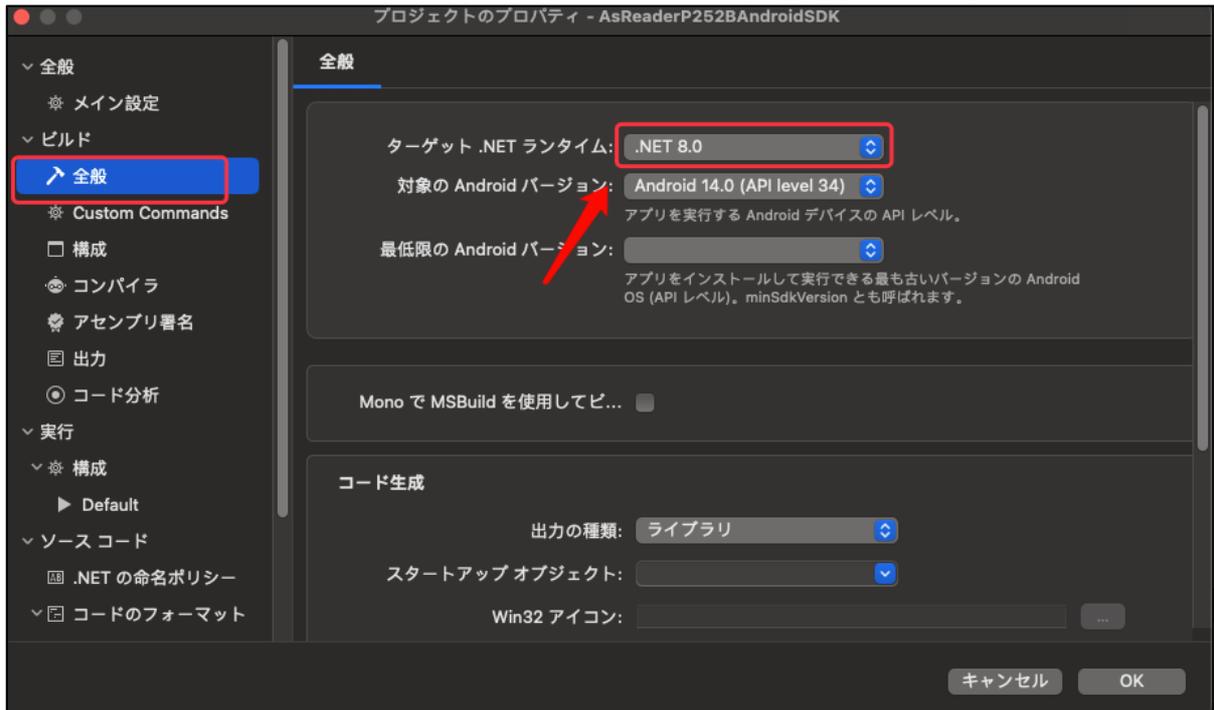
3 ビルドする.NET バージョンの変更

「.NET 7.0」を「.NET 8.0」に変更する方法です。

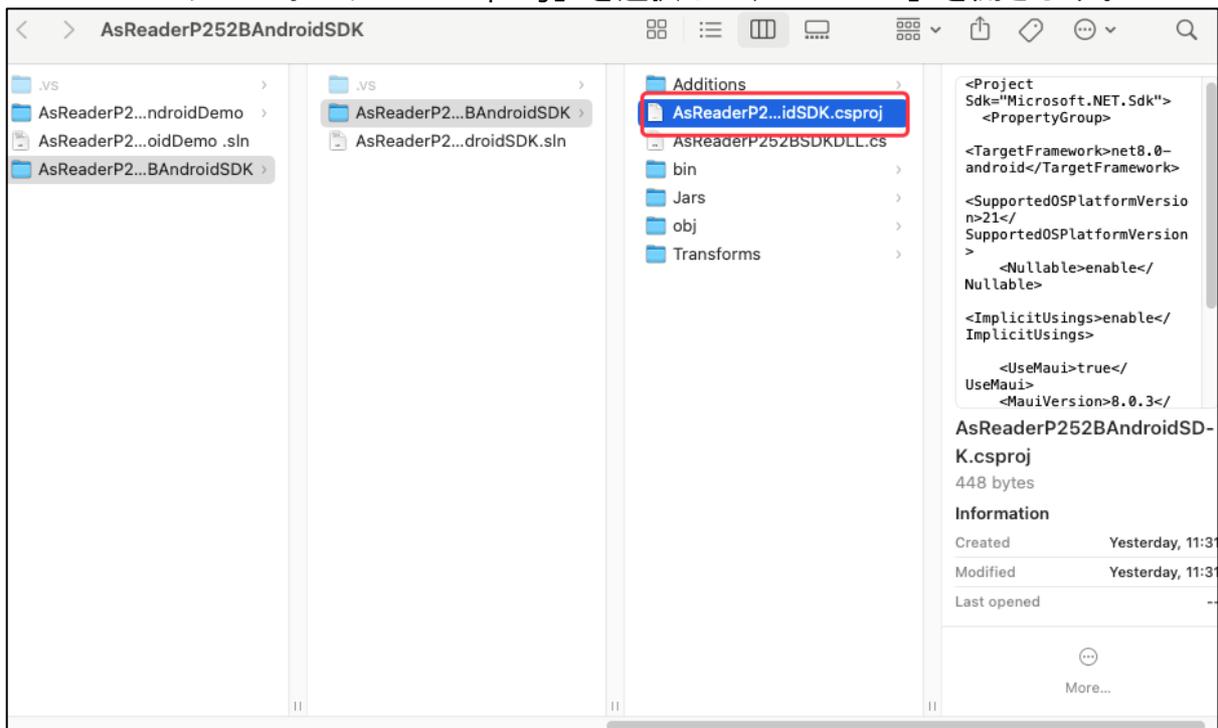
1. プロジェクトを右クリック→プロパティ



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



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



4. 以下の修正を行い、ファイルを保存。

```

1 <Project Sdk="Microsoft.NET.Sdk">
2   <PropertyGroup>
3     <TargetFramework>net7.0-android</TargetFramework>
4     <SupportedOSPlatformVersion>21</SupportedOSPlatformVersion>
5     <Nullable>enable</Nullable>
6     <ImplicitUsings>enable</ImplicitUsings>
7   </PropertyGroup>
8 </Project>

```

```

1 <Project Sdk="Microsoft.NET.Sdk">
2   <PropertyGroup>
3     <TargetFramework>net8.0-android</TargetFramework>
4     <SupportedOSPlatformVersion>21</SupportedOSPlatformVersion>
5     <Nullable>enable</Nullable>
6     <ImplicitUsings>enable</ImplicitUsings>
7   </PropertyGroup>
8   <UseMaui>true</UseMaui>
9   <MauiVersion>8.0.3</MauiVersion>
10  <SkipValidateMauiImplicitPackageReferences>true</SkipValidateMauiImplicitPackageReferences>
11 </PropertyGroup>
12 </Project>

```

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

↓

<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>

4. 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、Objective-C の SDK マニュアルを参照してください。
「ASR-P252B_Android_SDK_Reference_Guide_JP_1_」