



.NET MAUI Android SDK For ASR-P3xU

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	10
2 SDK の使用	11
2.1 SDK のインポート	11
2.2 「MyAsReaderDeviceListener」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	11
2.3 AsReaderP3xUMAUIManager オブジェクトを初期化	11
2.4 OnStateChanged を実現	12
2.5 OnReadTag を実現	12
2.6 OnKeyEvent を実現	12
2.7 OnActionChanged を実現	12
2.8 OnAccessResult を実現	12
2.9 USB 接続	12
2.10 RFID スキャン開始	12
2.11 RFID スキャン停止	13
2.12 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 17.6.7 (build 417)
- ・ Android Studio :

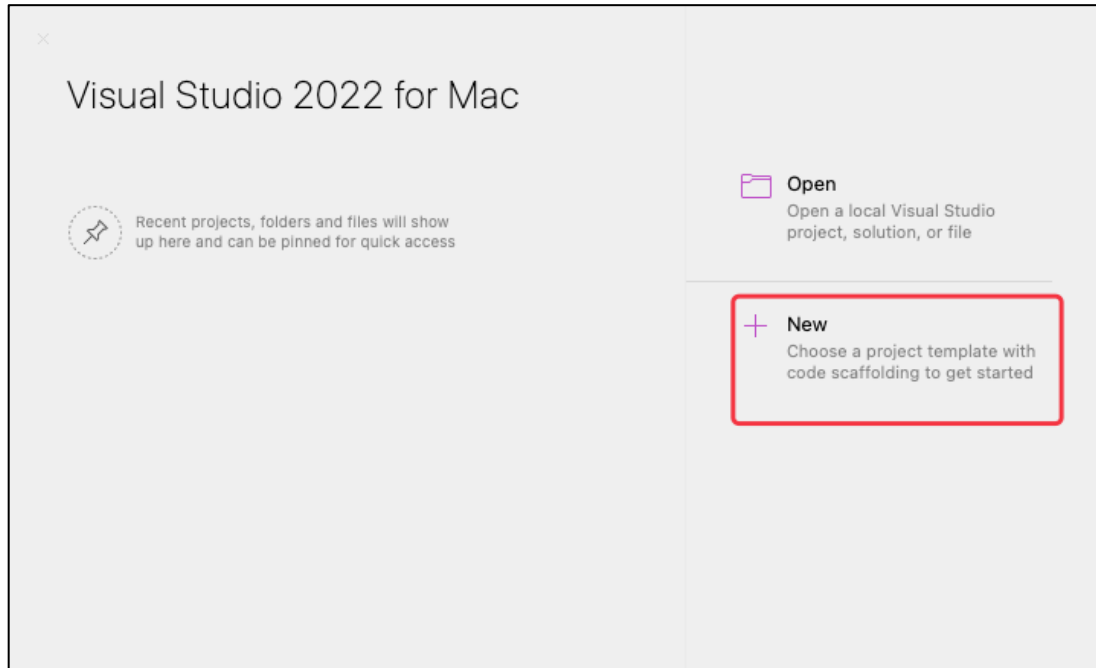
Android Studio Giraffe | 2022.3.1 Patch 4

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

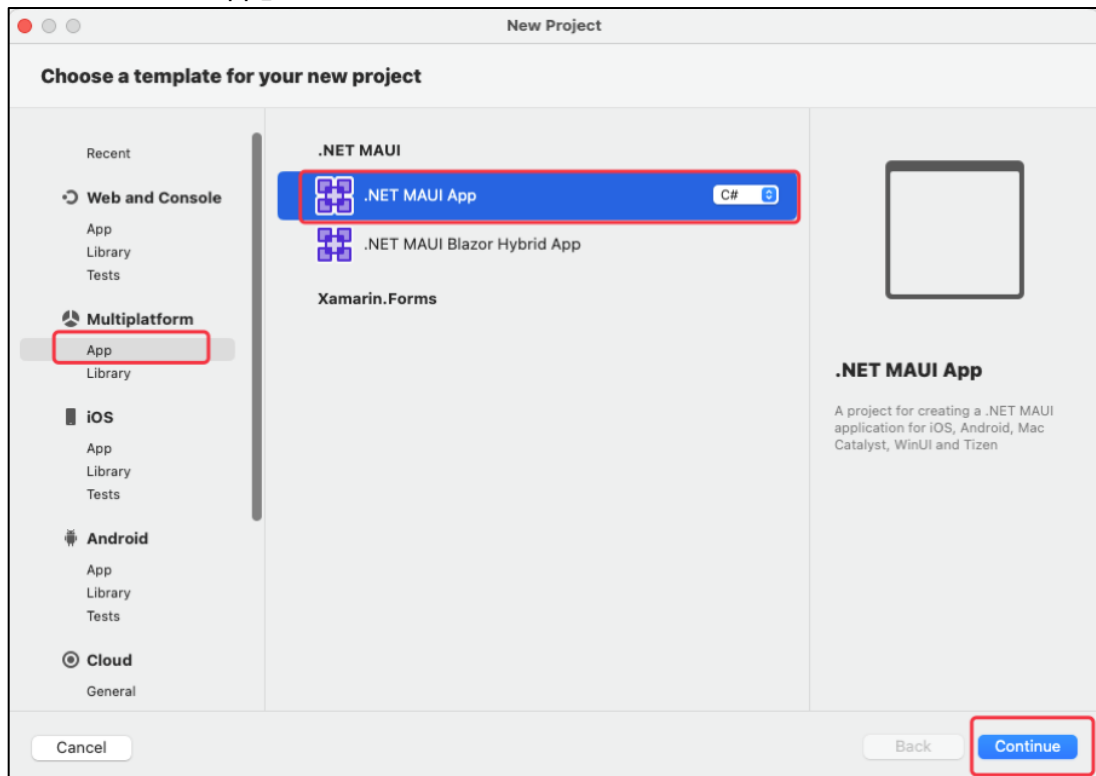
1 SDK の追加

1.1 SDK をインポートする

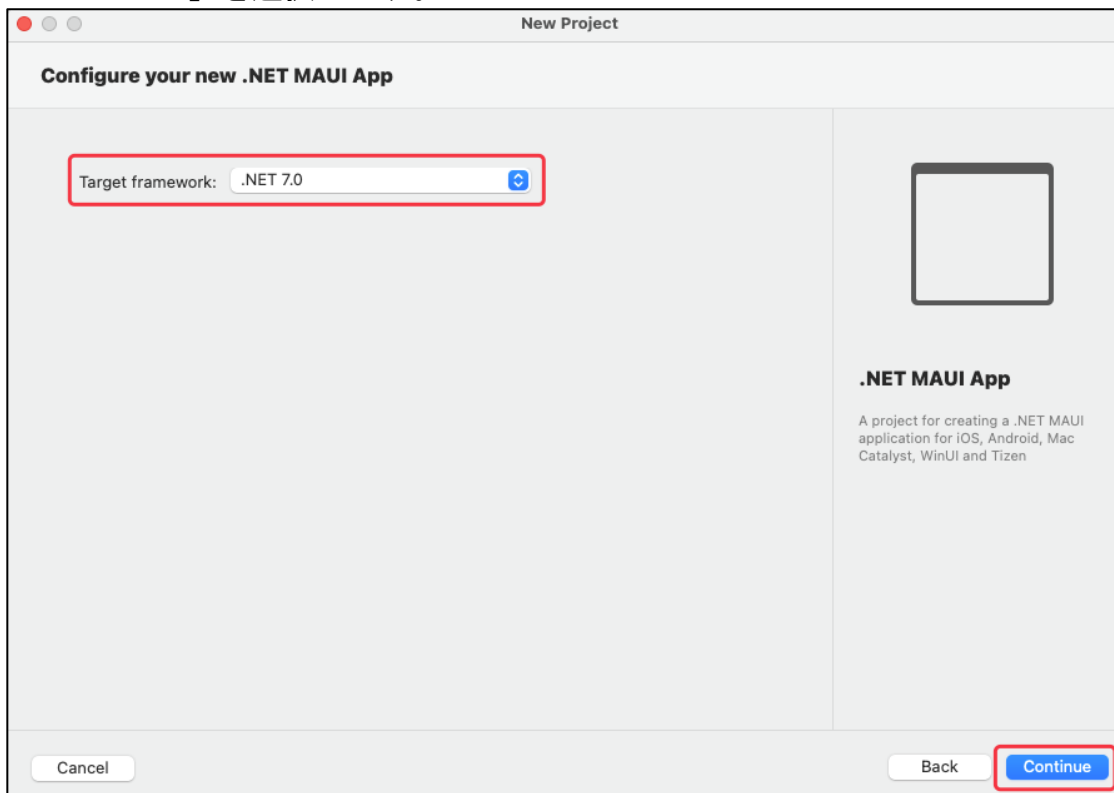
1. AsReaderP3xUSDK をローカルにダウンロードします。
2. プロジェクトを新規作成します。



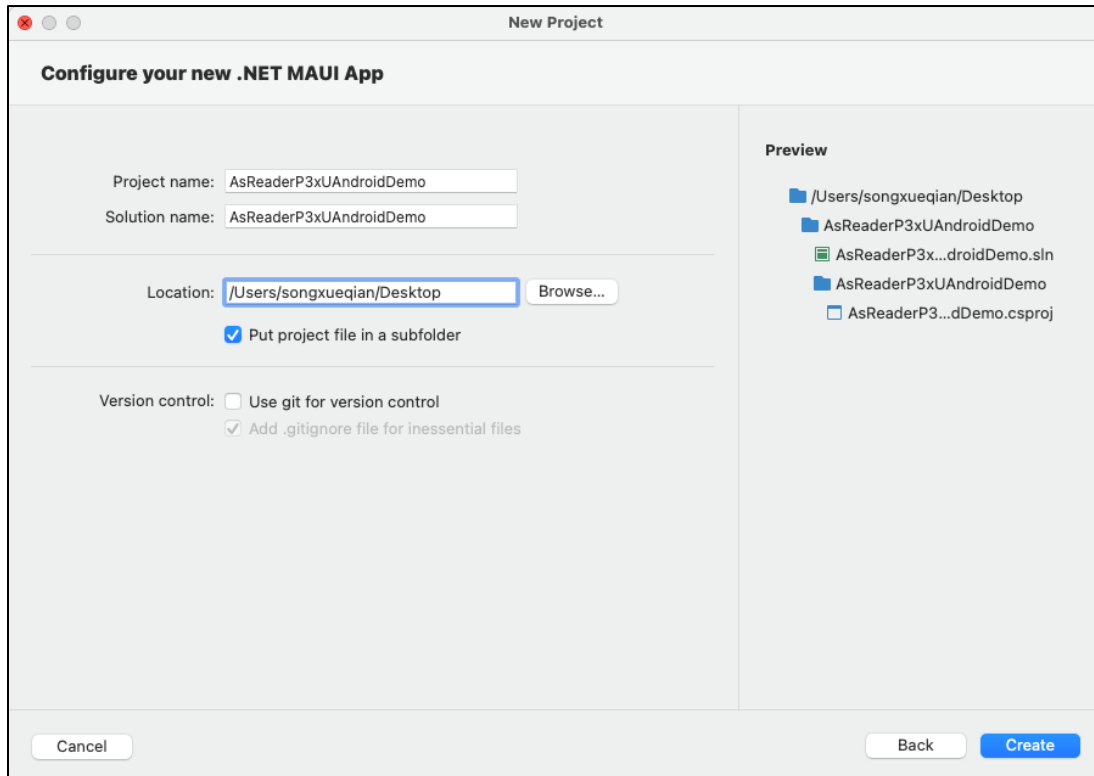
3. 「.NET MAUI App」を選択します。



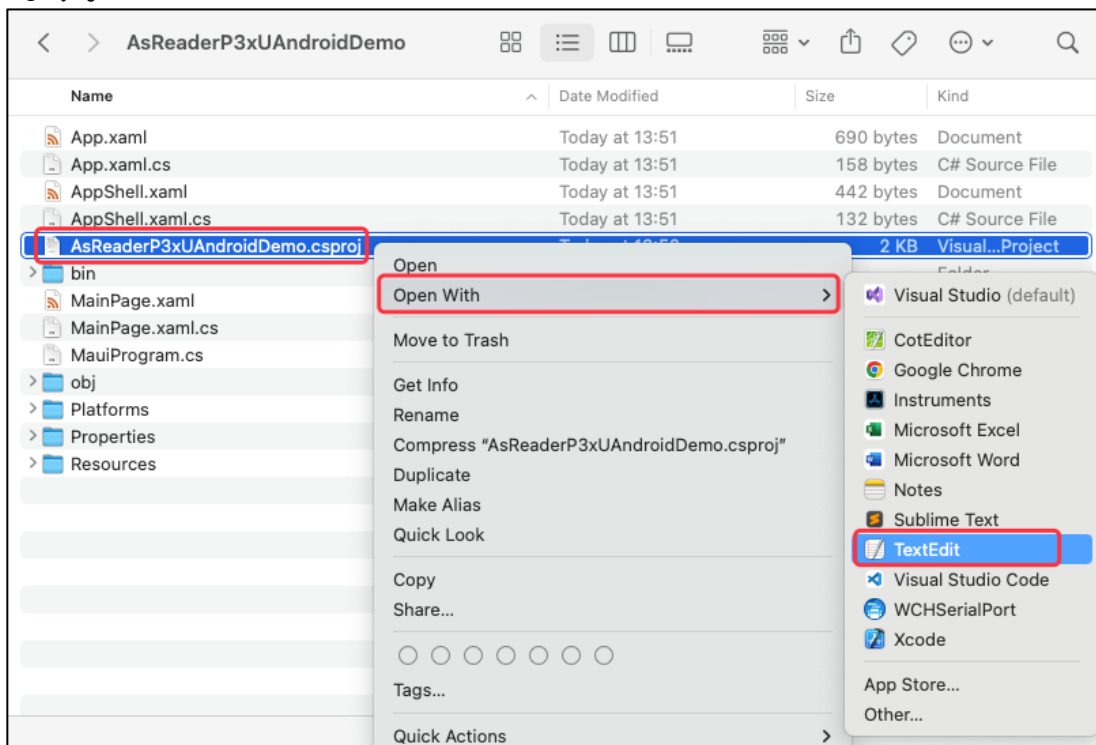
4. 「.NET 7.0」を選択します。



5. アプリ名とロケーションを作成します。



6. 「AsReaderP3xUAndroidDemo.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をご使用される場合、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>AsReaderP3xUAndroidDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderP3xUAndroidDemo</ApplicationTitle>
  
```

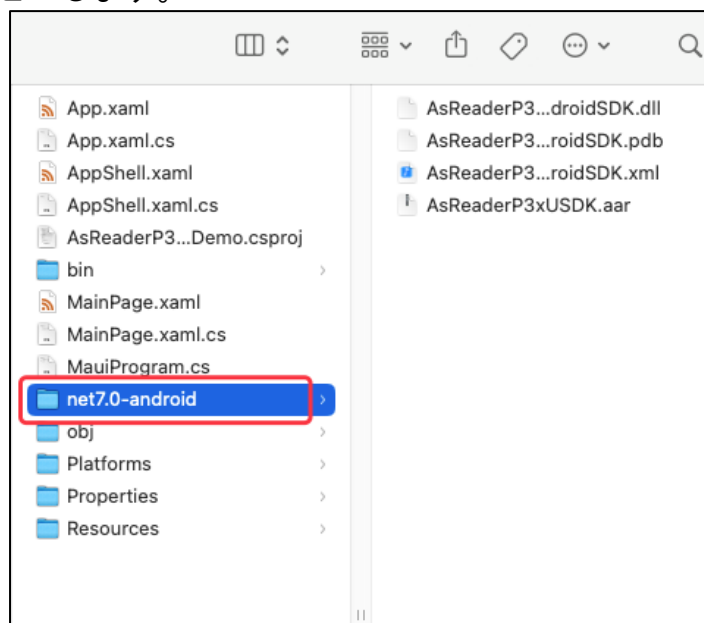
8. 削除すると、下図の赤枠内ようになります。

```

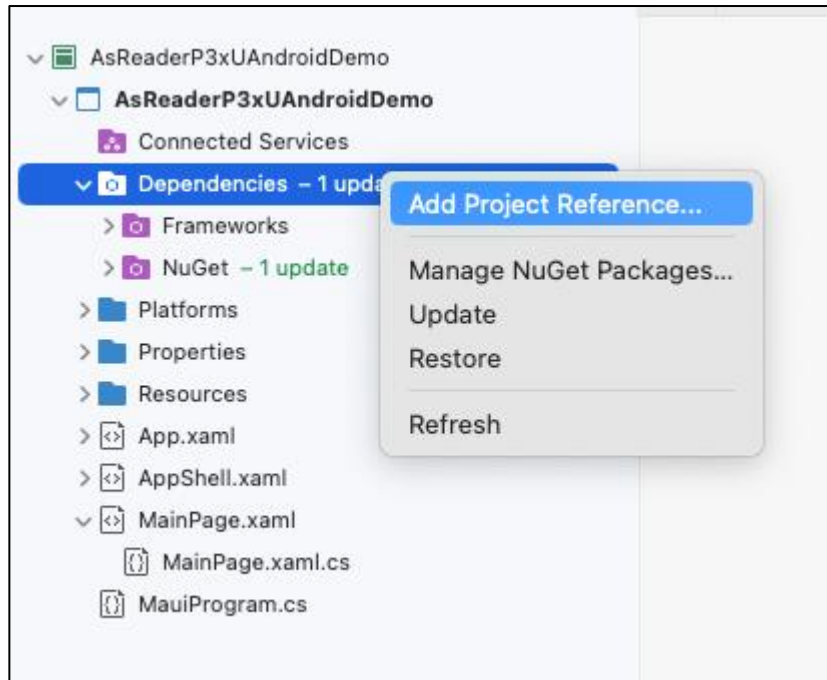
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net7.0-android</TargetFrameworks>
    <TargetFrameworks Condition="$(MSBuild::IsOSPlatform('windows'))">$(TargetFrameworks);net7.0-
windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://
github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>AsReaderP3xUAndroidDemo</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>AsReaderP3xUAndroidDemo</ApplicationTitle>
  
```

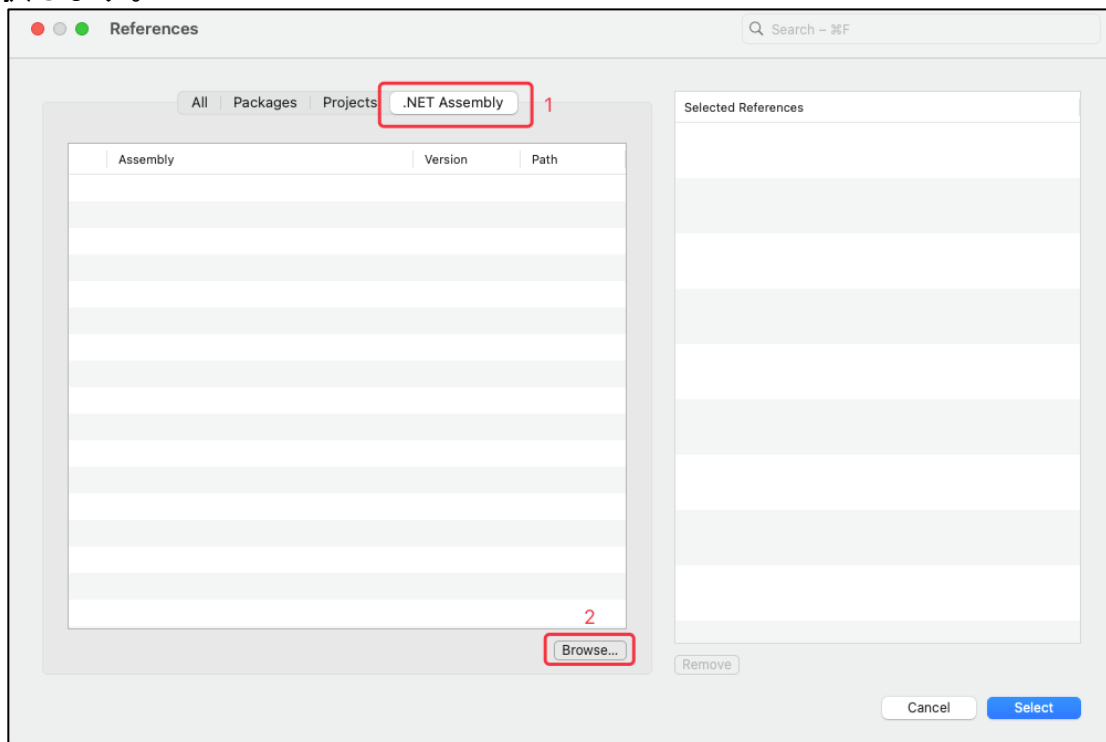
9. ローカルにダウンロードした「AsReaderP3xUAndroidSDK」をプロジェクトフォルダーにコピーします。



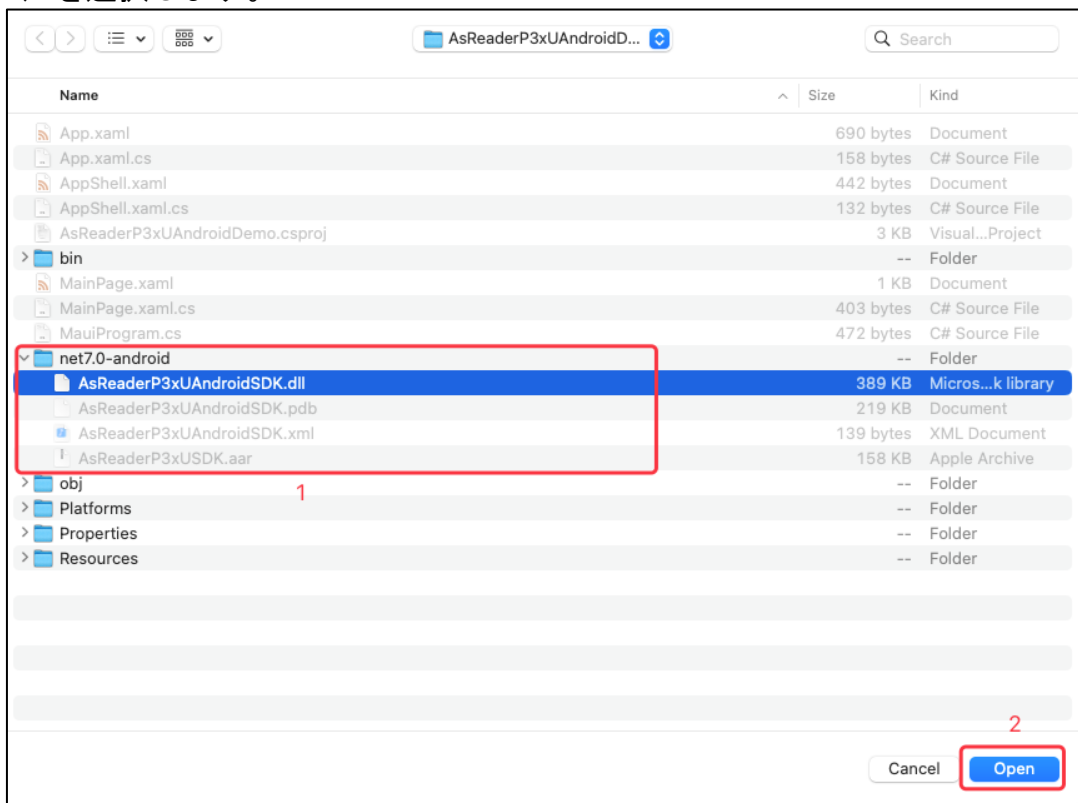
10. Dependencies -> 右クリック -> Add Project Reference...を選択します。



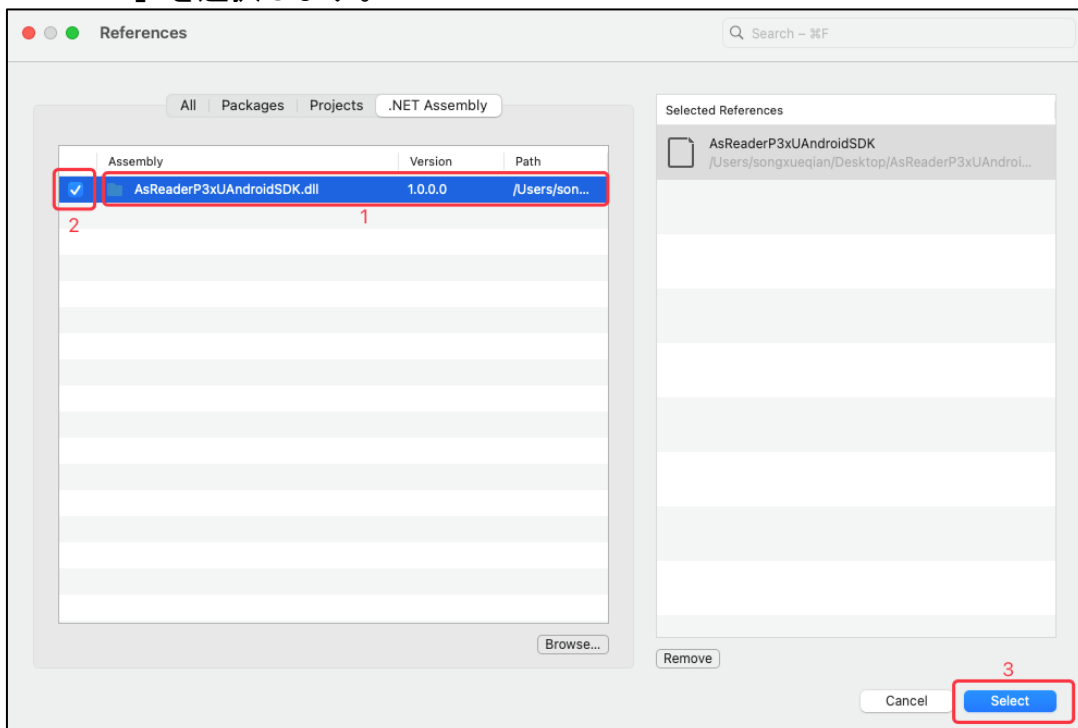
11. その後、「.NET Assembly」（下図1）を選び、「Browse...」（下図2）を選択します。



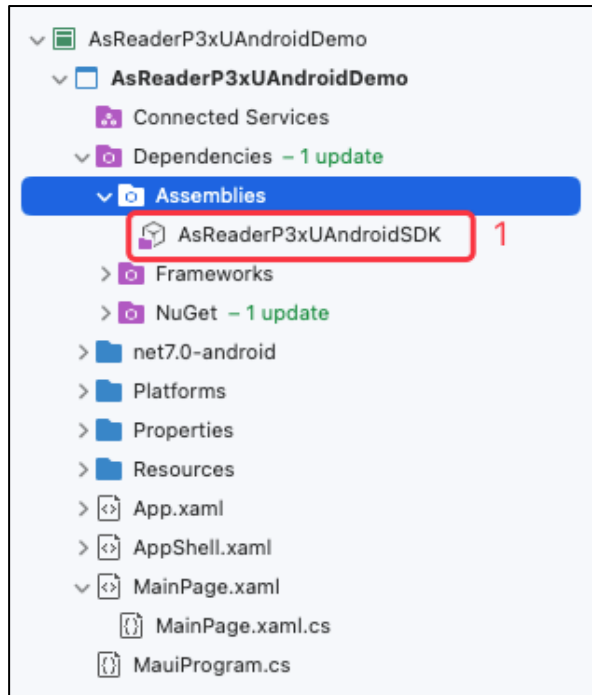
12. 「AsReaderP3xUAndroidSDK.dll」 (下図 1) を選択し「Open」ボタン (下図 2) を選択します。



13. 「References」ウィンドウ内のリストにある「AsReaderP3xUAndroidSDK.dll」左側のチェックボックスを選択、「Select」を選択します。



14. 「Assemblies」に「AsReaderP3xUAndroidSDK」（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="AsReaderP3xUAndroidSDK">
    <HintPath>net7.0-android\AsReaderP3xUAndroidSDK.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" package="com.asreader.p3xu.maui.android.demo" android:versionCode="1" android:versionName="1.0.0">
3   <application android:allowBackup="true" android:supportRtl="true" android:label="AsReaderP3xUAndroidDemo"></application>
4   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
5   <uses-permission android:name="android.permission.INTERNET" />
6   <uses-permission android:name="android.permission.WAKE_LOCK" />
7   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
8   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
9   <uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
10  <uses-permission android:name="android.permission.READ_MEDIA_VIDEO" />
11  <uses-sdk android:minSdkVersion="29" android:targetSdkVersion="33" />
12 </manifest>
13

```

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.P3xu;
using Com.Asreader.P3xu.Rfid.Event;
using Com.Asreader.P3xu.Rfid.Type;
using Com.Asreader.P3xu.Rfid.Params;
using Com.Asreader.P3xu.Device.Type;
using Com.Asreader.P3xu.Device;
#endif
```

2.2 「MyAsReaderDeviceListener」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 MyAsReaderDeviceListener : Java.Lang.Object, IAsReaderP3xUEventListener
{
    public MyAsReaderDeviceListener()
    {
    }

    public void OnAccessResult(AsReaderP3xU p0, AsReaderP3xUResultCode p1, AsReaderP3xUActionState p2, string p3, string p4)
    {
    }

    public void OnActionChanged(AsReaderP3xU p0, AsReaderP3xUActionState p1)
    {
    }

    public void OnKeyEvent(AsReaderP3xU p0, AsReaderP3xUKeyType p1, AsReaderP3xUKeyState p2)
    {
    }

    public void OnReadTag(AsReaderP3xU p0, AsReaderP3xUActionState p1, string p2, float p3)
    {
    }

    public void OnStateChanged(AsReaderP3xU p0, AsReaderP3xUConnectionState p1)
    {
    }
}
```

2.3 AsReaderP3xUMAUIManager オブジェクトを初期化

```
AsReaderP3xU mAsReaderP3xU = AsReaderP3xUManager.Instance;
MyAsReaderDeviceListener myAsReaderDeviceListener = new MyAsReaderDeviceListener();
mAsReaderP3xU.SetEventListener(myAsReaderDeviceListener);

AsReaderP3xUDeviceUsbCdc usbCdc = new AsReaderP3xUDeviceUsbCdc(MainActivity.Instance);
```

2.4 OnStateChanged を実現

```
public void OnStateChanged(AsReaderP3xU p0, AsReaderP3xUConnectionState p1)
{
    ...
}
```

2.5 OnReadTag を実現

```
public void OnReadTag(AsReaderP3xU p0, AsReaderP3xUActionState p1, string p2, float p3)
{
    ...
}
```

2.6 OnKeyEvent を実現

```
public void OnKeyEvent(AsReaderP3xU p0, AsReaderP3xUKeyType p1, AsReaderP3xUKeyState p2)
{
    ...
}
```

2.7 OnActionChanged を実現

```
public void OnActionChanged(AsReaderP3xU p0, AsReaderP3xUActionState p1)
{
    ...
}
```

2.8 OnAccessResult を実現

```
public void OnAccessResult(AsReaderP3xU p0, AsReaderP3xUResultCode p1, AsReaderP3xUActionState p2, string p3, string p4)
{
    ...
}
```

2.9 USB 接続

```
mDevice = mAsReaderP3xU.CurrentDevice;
if (mDevice == null)
{
    mDevice = usbCdc;
}
mAsReaderP3xU.ConnectDevice(usbCdc);
```

mDevice : AsReaderP3xUDevice クラスのインスタンスオブジェクト

mAsReaderP3xU : AsReaderP3xU クラスのインスタンスオブジェクト

usbCdc : AsReaderP3xUDevice クラスのインスタンスオブジェクト

2.10 RFID スキャン開始

```
mAsReaderP3xU.Inventory();
```

2.11 RFID スキャン停止

```
mAsReaderP3xU.Stop();
```

2.12 切断

```
if (mDevice != null)
{
    mAsReaderP3xU.DisconnectDevice();
}
```

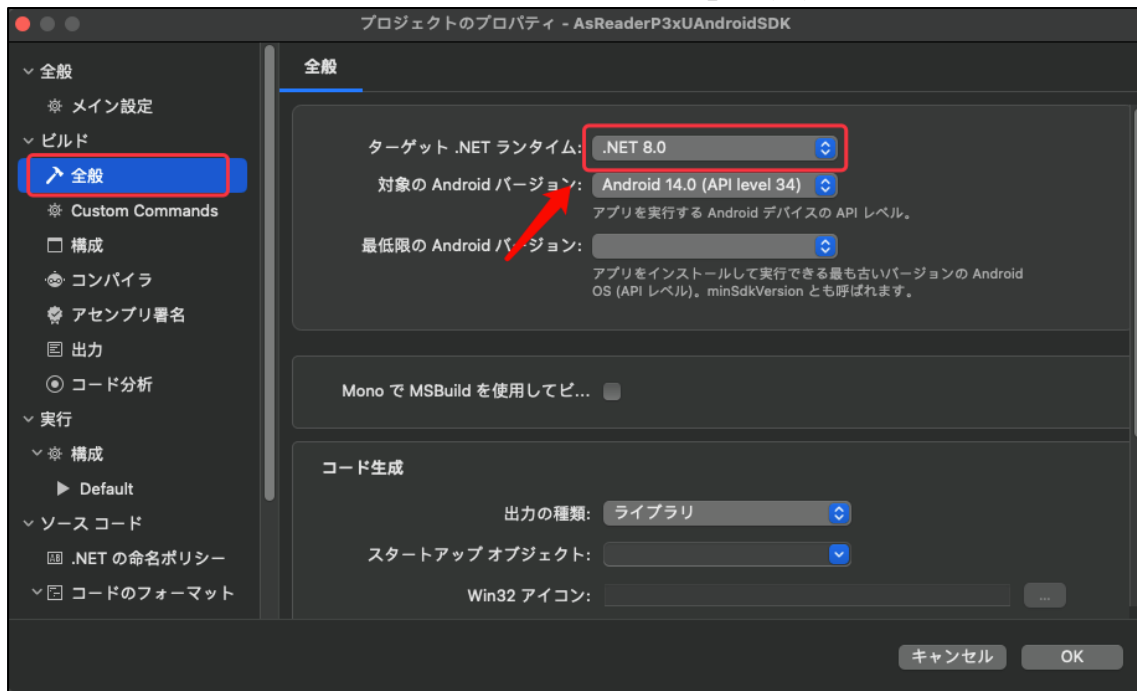
3 ビルドする.NET バージョンの変更

「.NET 7.0」を「.NET 8.0」に変更する方法です。

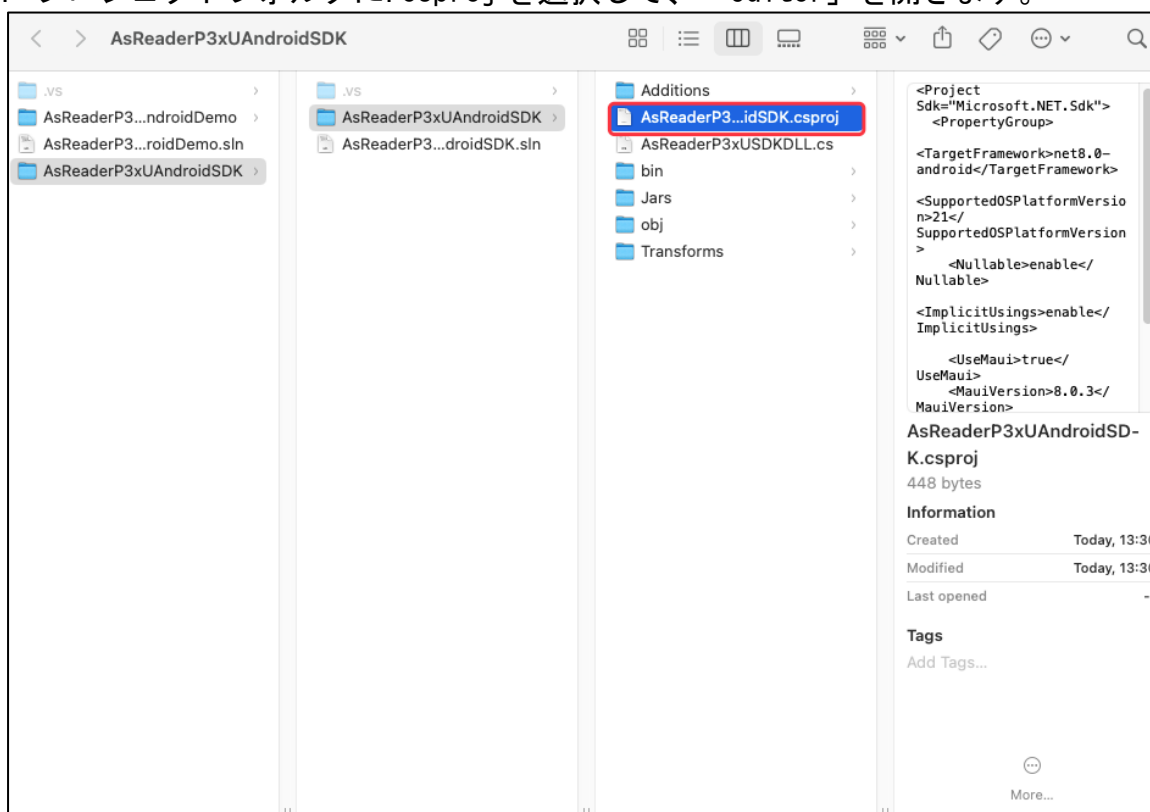
1. プロジェクトを右クリック→プロパティ



2. 全般->ターゲット.NET ランタイムを「.NET 8.0」に変更します。



3. プロジェクトフォルダに .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
8   </PropertyGroup>
9 </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
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>

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 マニュアルを参照してください。

「AsReaderP3xU_SDK_Reference_Guide_for_Android_Developers_JP_x_x」