MAUI iOS For AsReader251G SDK

SDK マニュアル　V1.0

Asterisk,Inc.
2024-03-15

AsReader®

変更履歴

No.	バージョン	変更内容	日付
1	1.0	新規作成	2024/03/15

AsReader

目 次

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)

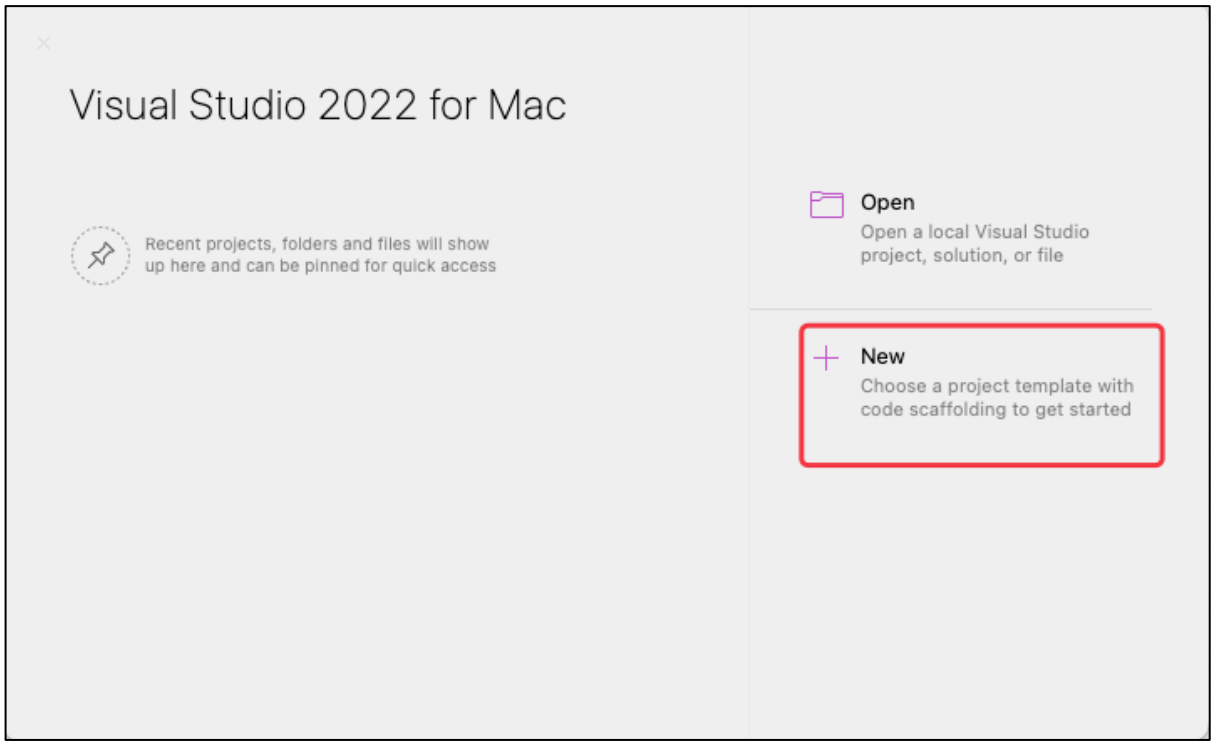
Xcode：15.2

この文書の説明には、上記開発環境を使用しています。

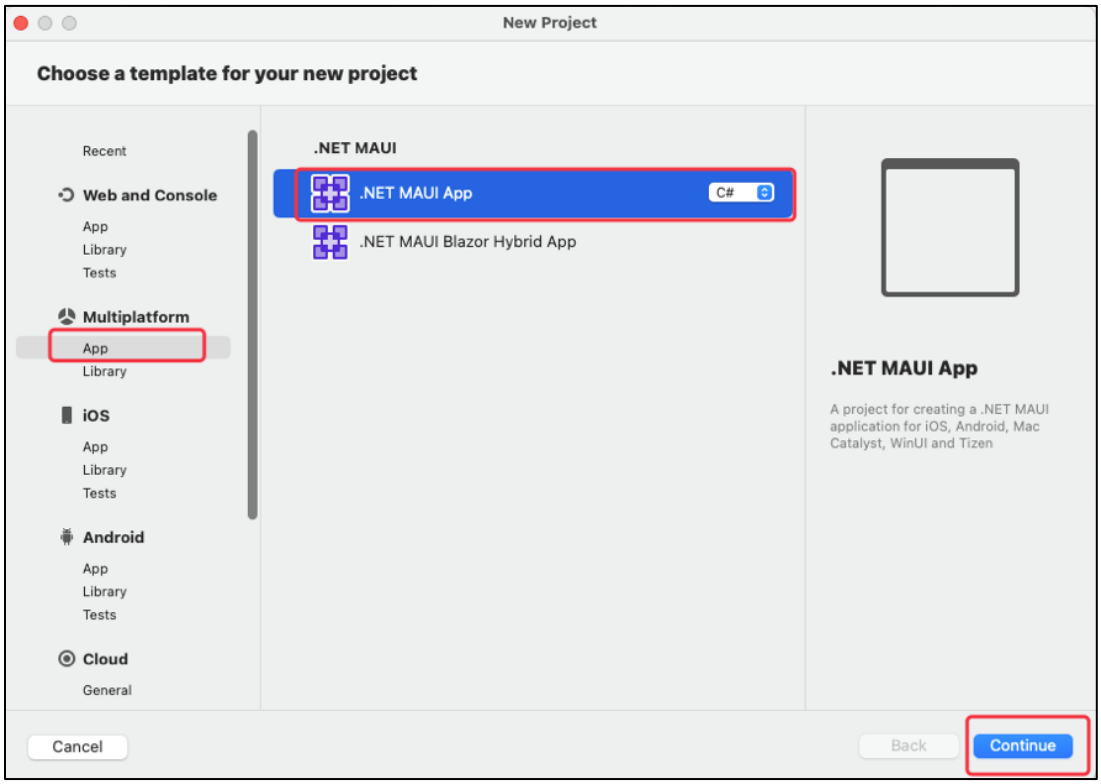
1 SDK の追加

1.1 SDK をインポートする

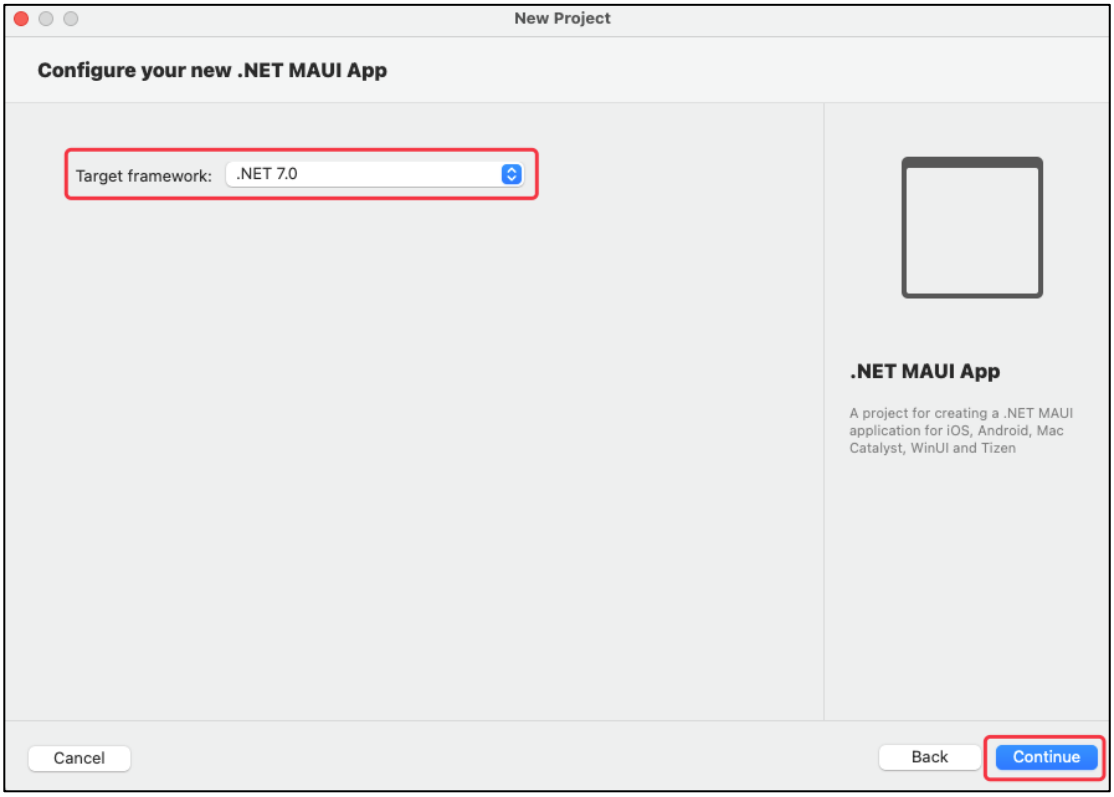
1. AsReader251GSDK をローカルにダウンロードします。
2. プロジェクトを新規作成します。



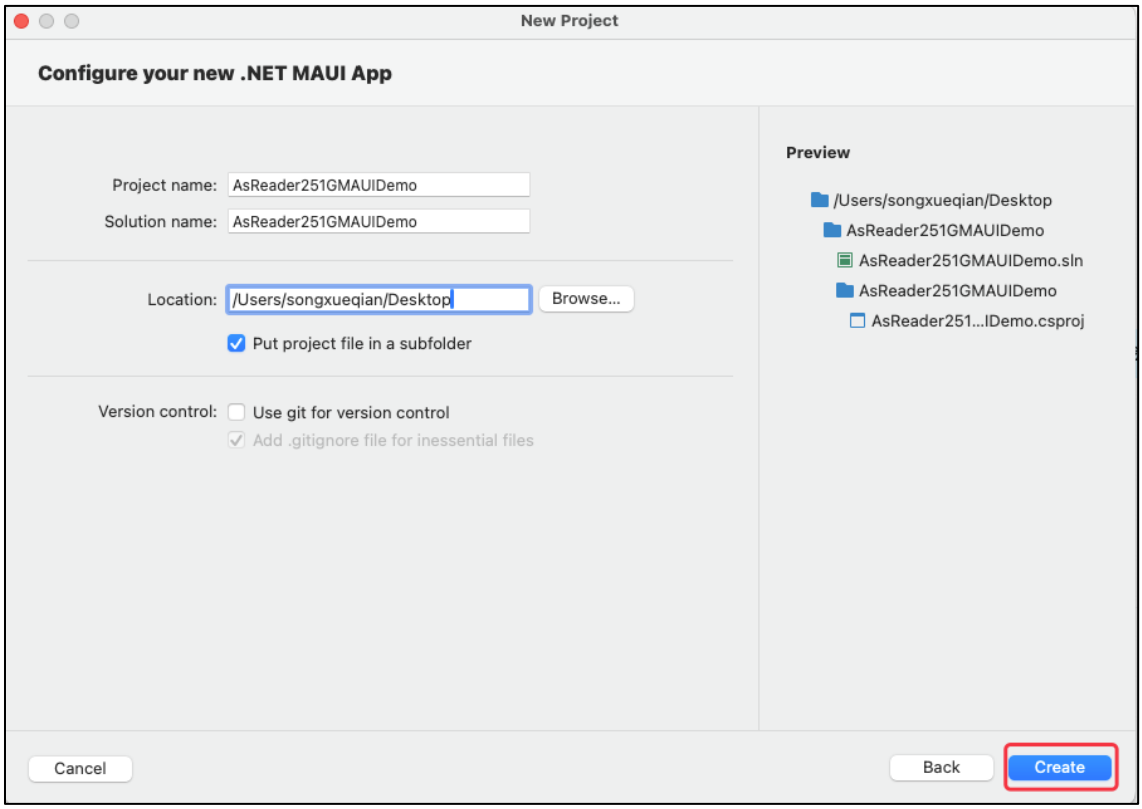
3.「.NET MAUI APP」を選択します。



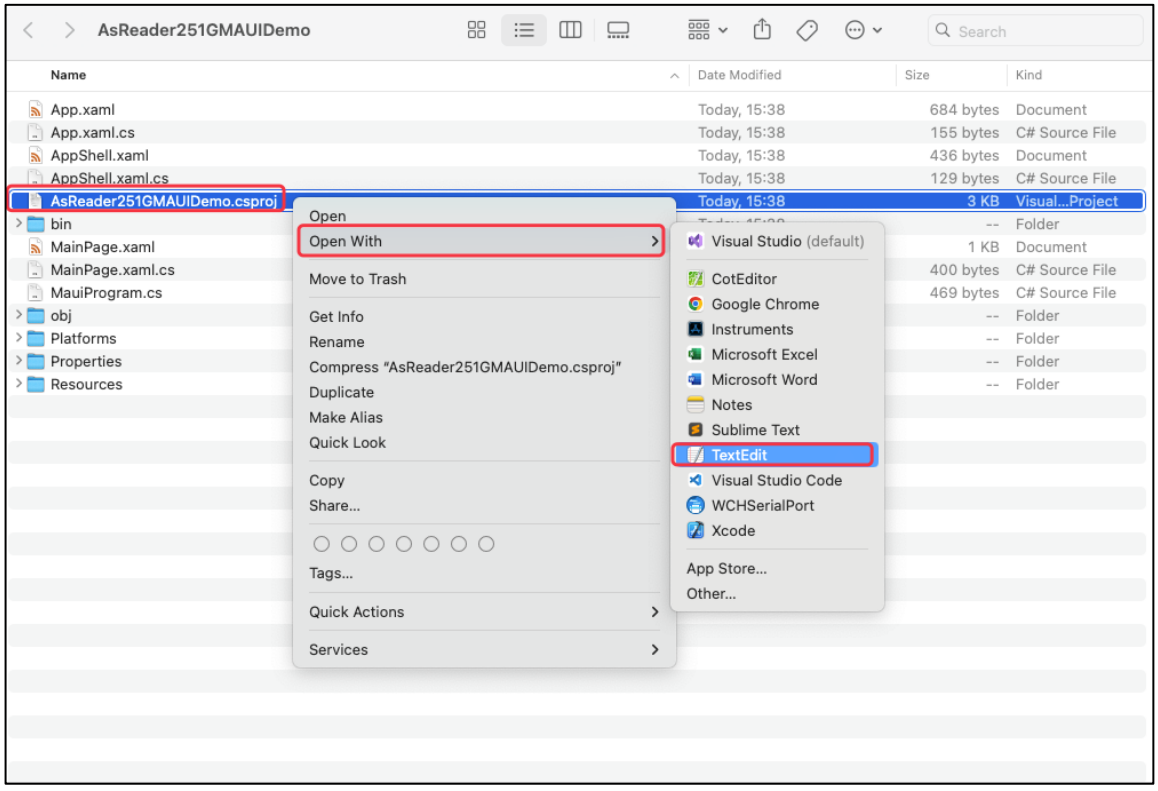
4.「.NET 7.0」を選択します。



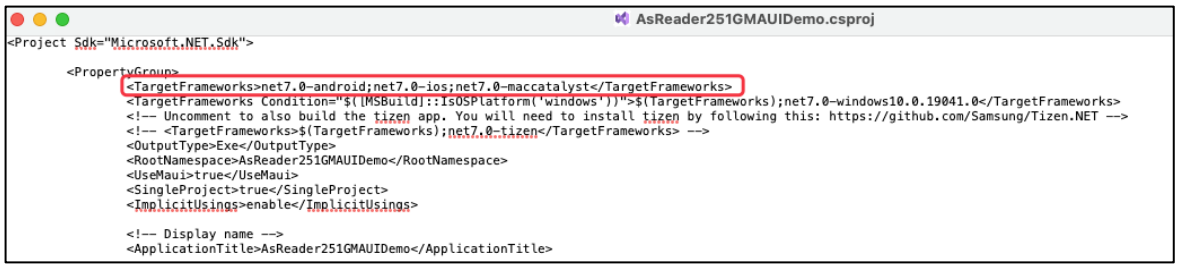
5. アプリ名とロケーションを作成します。



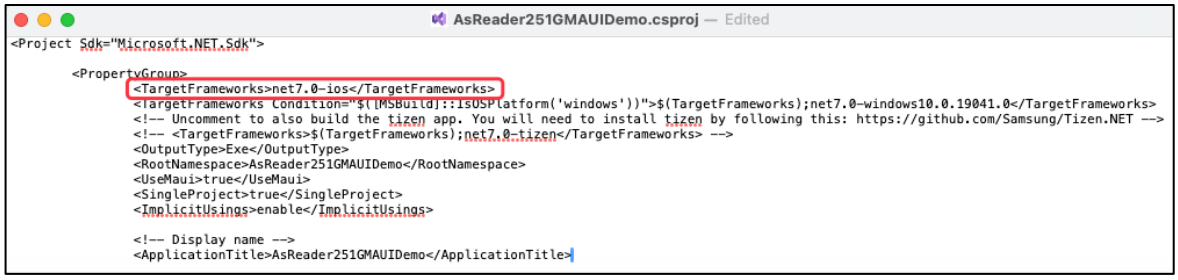
6.「AsReader251GMAUIDemo.csproj」を右クリックして、TextEdit で開きます。



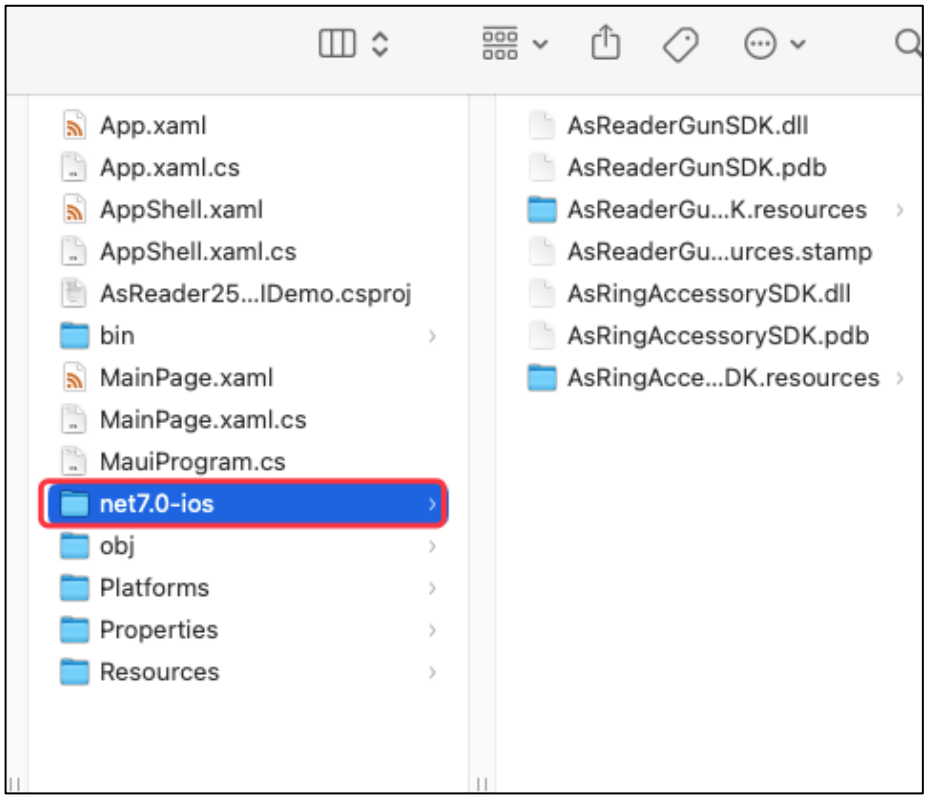
7. 下図赤枠内（net7.0-ios）で、必要としない全てのフレームを削除します。（iOS プラットフォームのみ使用した場合の操作です。マルチプラットフォームの場合、この操作は不要です。）



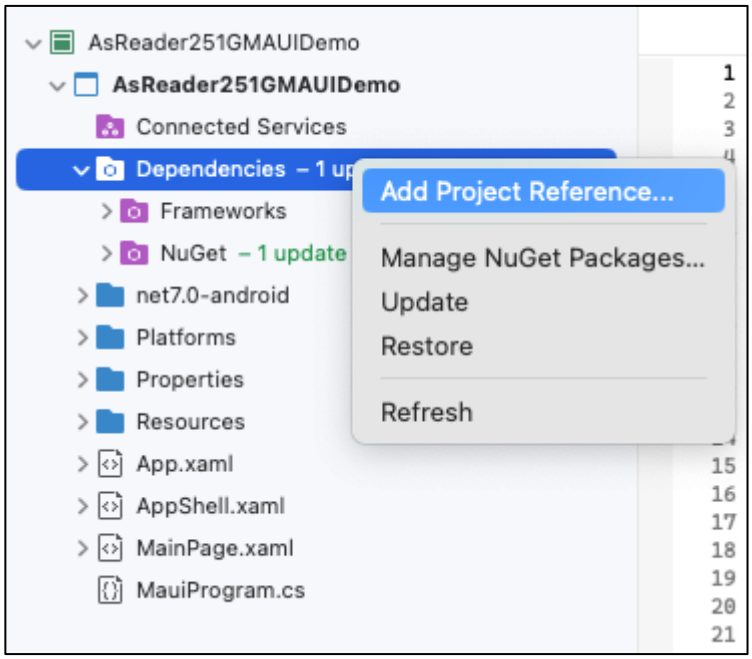
8. 削除完了後は下図の通りです。



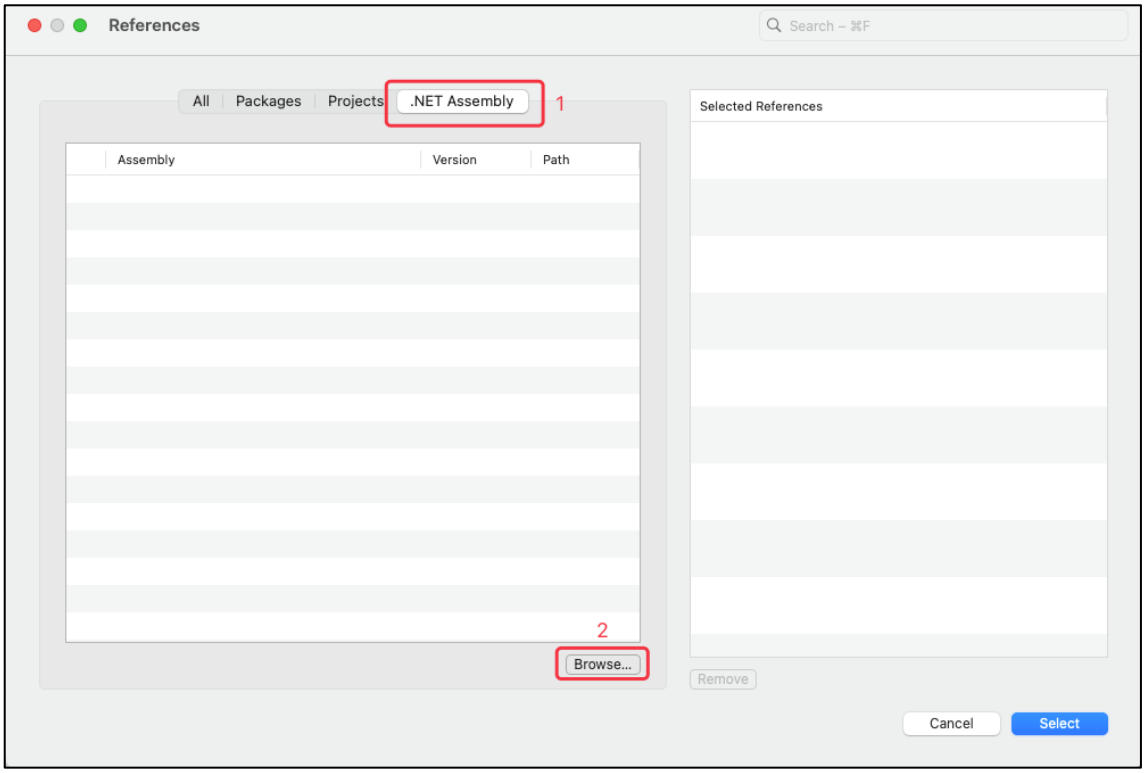
9. ローカルにダウンロードした AsReader251GSDK をプロジェクトフォルダーにコピーします。



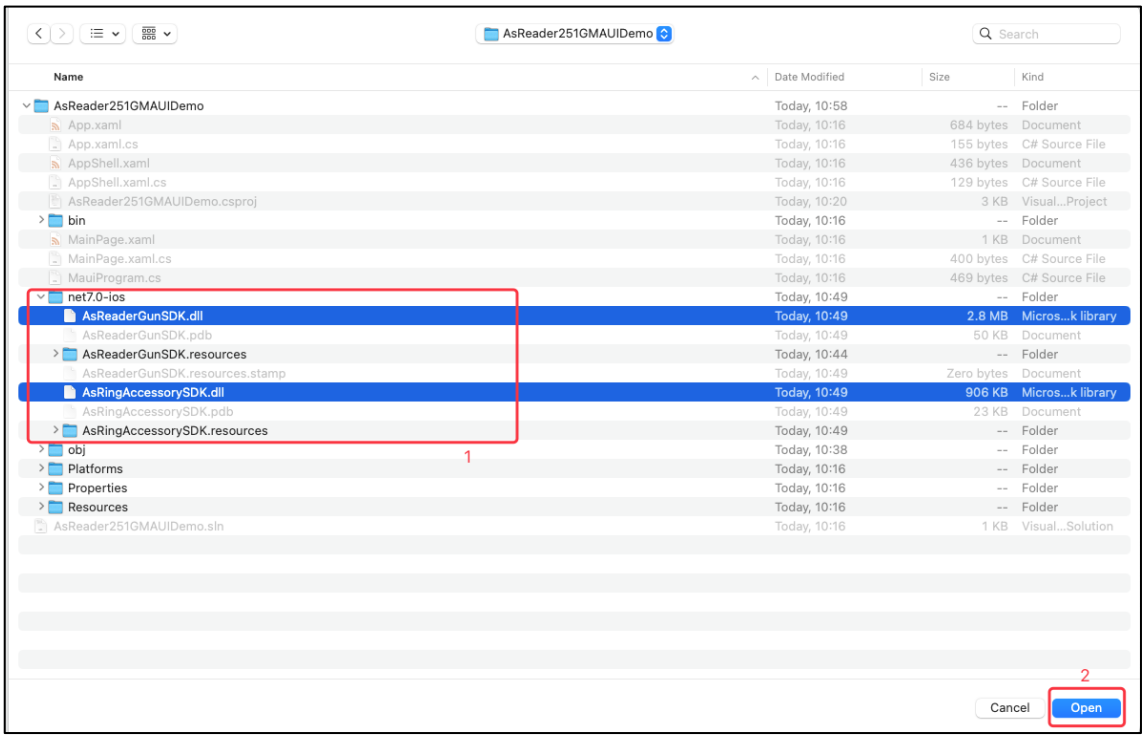
10. Dependencies -> 右クリック -> Add Project Reference…



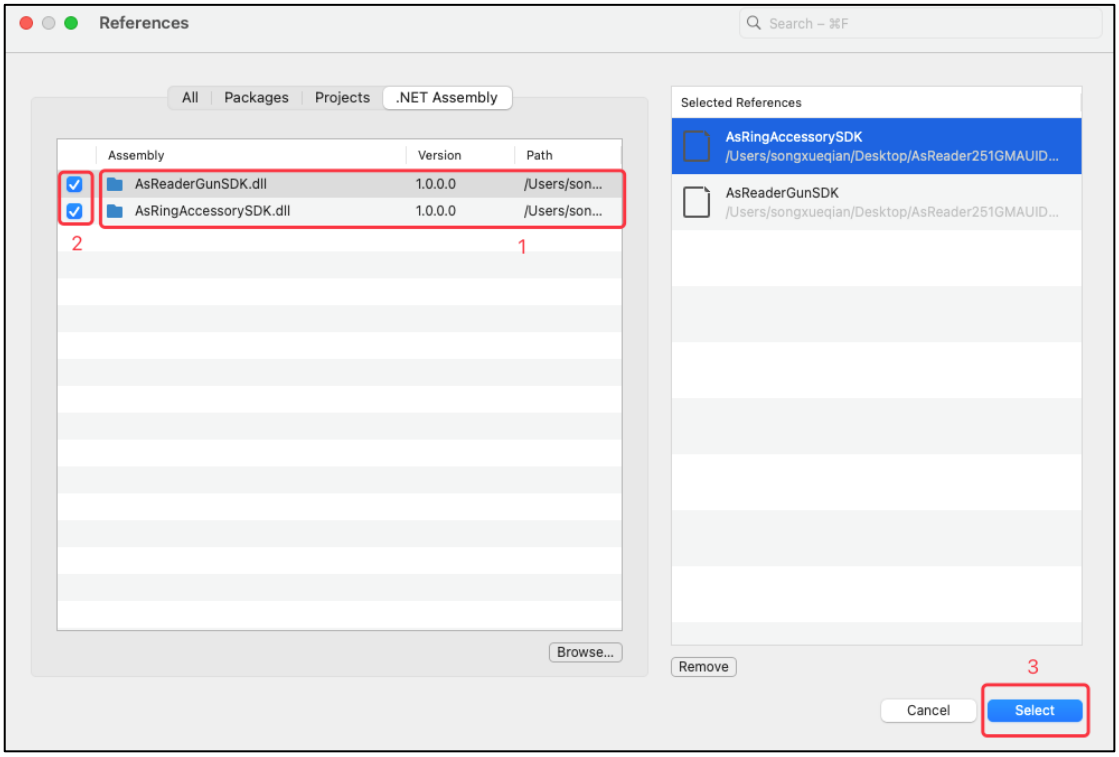
11. ポップアップしたウィンドウに「.Net Assembly」（下図 1）をクリックして、「Browse…」（下図 2）を選択します。



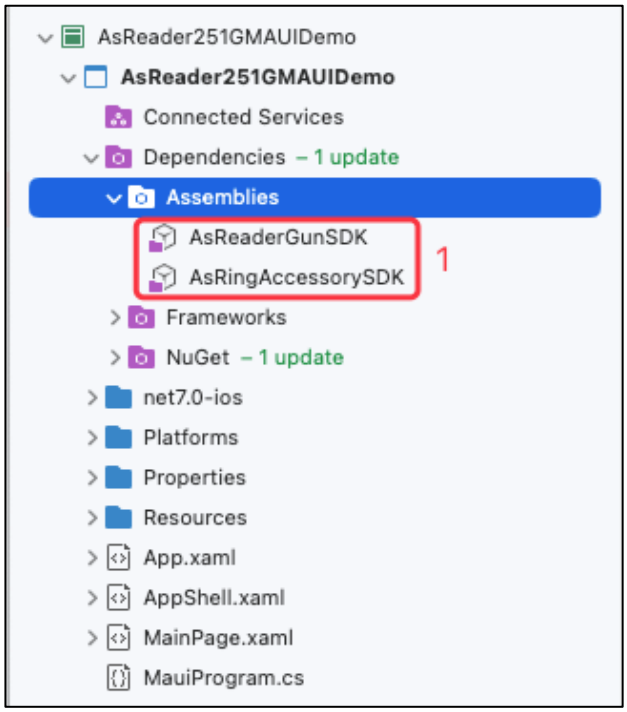
12. ポップアップしたウィンドウに「AsReaderGunSDK.dll」と「AsRingAccessorySDK.dll」（下図 1）を選択して、「Open」ボタン（下図 2）をクリックします。



13. 「References」ウィンドウのリストにある「AsReaderGunSDK.dll」と「AsRingAccessorySDK.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「AsReaderGunSDK.dll」と「AsRingAccessorySDK.dll」（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="AsRingAccessorySDK">
    <HintPath>net7.0-ios\AsRingAccessorySDK.dll</HintPath>
  </Reference>
  <Reference Include="AsReaderGunSDK">
    <HintPath>net7.0-ios\AsReaderGunSDK.dll</HintPath>
  </Reference>
</ItemGroup>
```

1.2 権限の追加

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	(2 items)
	String	jp.co.asx.asreader.gun
	String	jp.co.asx.asring.plus

AsReader®

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 IOS
using AsRingAccessorySDK;
using AsReaderGunSDK;
#endif
```

2.2「AsReaderDelegate」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 MyAsreaderDelegate : AsReaderDelegate
{
    public override void ReaderInitialized(AsReader reader)
    {
        AsReader251GManager.sharedInstance().setAsReaderDelegate();
    }
    public override void DetectBarcode(BarcodeType barcodeType, string codeId, NSData barcodeData)
    {
    }
    public override void ReadTag(string tag, float rssi, float phase, float frequency)
    {
    }
}
```

2.3 AsReader オブジェクトを初期化する

mAsReaderGUN は AsReaderGUN クラスのインスタンスオブジェクトです。

mAsReaderDelegate は MyAsReaderDelegate クラスのインスタンスオブジェクトです。

mReader は AsReader クラスのインスタンスオブジェクトです。

```
public AsReader251GManager()
{
    mAsReaderGUN = new AsReaderGUN("com.asreader.gun");
    mAsreaderDelegate = new MyAsreaderDelegate();
    NSNotificationCenter center = NSNotificationCenter.DefaultCenter;
    center.AddObserver(new NSString("AsReaderGUNConnected"), AsReaderGUNConnected);
    center.AddObserver(new NSString("AsReaderGUNDisconnected"), AsReaderGUNDisconnected);
}
void AsReaderGUNConnected(NSNotification notification)
{
    mReader = new AsReader(mAsReaderGUN, mAsreaderDelegate);
}
void AsReaderGUNDisconnected(NSNotification notification)
{
    if (mReader != null)
    {
        mReader = null;
    }
}
public void setAsReaderDelegate()
{
    if (mReader != null)
    {
        mReader.SetDelegate(mAsreaderDelegate);
    }
}
```

2.4 ReaderInitialized を実現する

```
public override void ReaderInitialized(AsReader reader)
{
    AsReader251GManager.sharedInstance().setAsReaderDelegate();
}
```

2.5 DetectBarcode を実現する

```
public override void DetectBarcode(BarcodeType barcodeType, string codeId, NSData barcodeData)
{
}
```

2.6 ReadTag を実現する

```
public override void ReadTag(string tag, float rssi, float phase, float frequency)
{
}
```

2.7 Barcode スキャン開始

mReader は AsReader クラスのインスタンスオブジェクトです。

```
ResultCode code = mReader.StartDecode;
```

2.8 Barcode スキャン停止

mReader は AsReader クラスのインスタンスオブジェクトです。

```
ResultCode code = mReader.StopSync;
```

2.9 RFID インベントリ開始

mReader は AsReader クラスのインスタンスオブジェクトです。

```
ResultCode code = mReader.Inventory;
```

2.10 RFID インベントリ停止

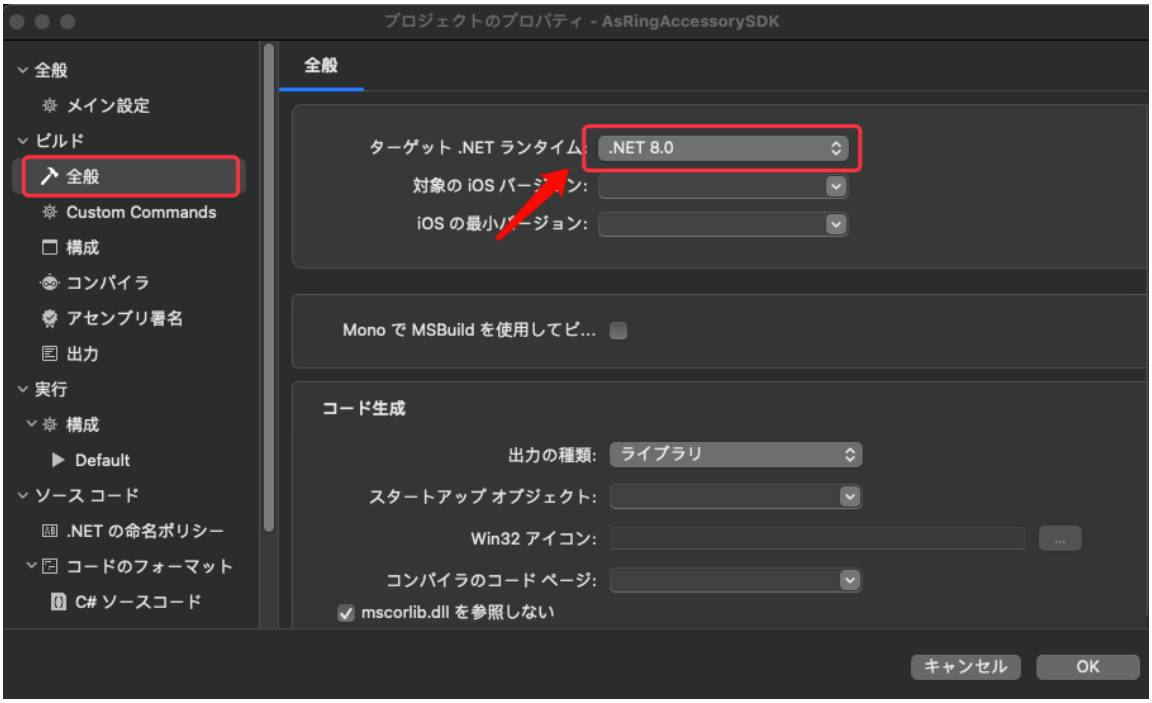
mReader は AsReader クラスのインスタンスオブジェクトです。

```
ResultCode code = mReader.Stop;
```

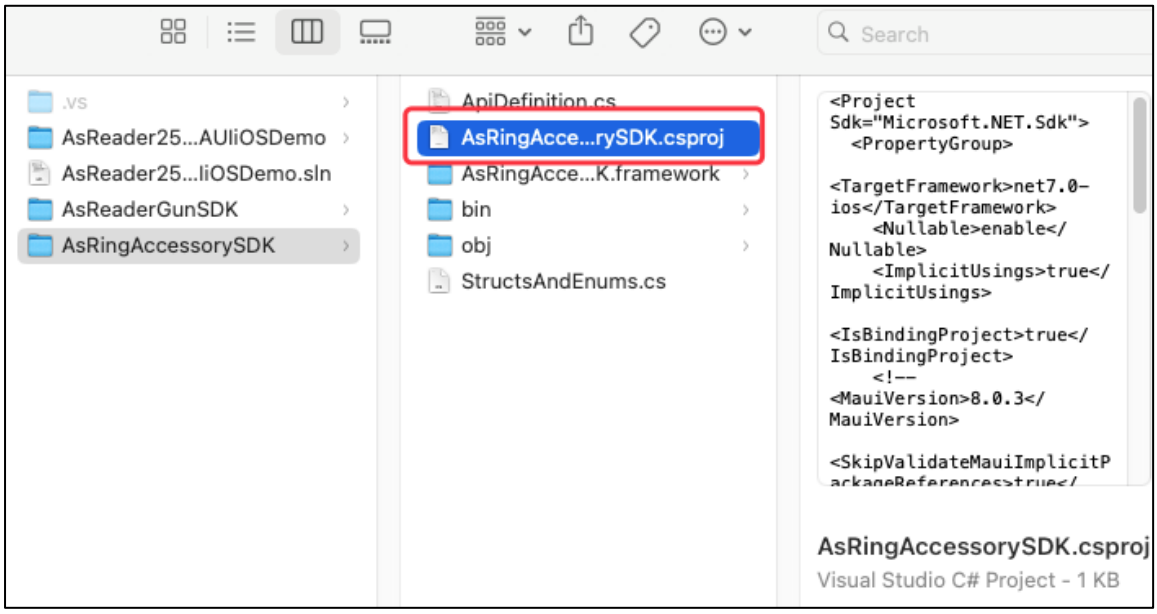
3 AsReader 251G SDK MAUI .Net の変更

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 ランタイムで「.NET 8.0」を選択します。



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



4. 以下の修正を行います。

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0-ios</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>true</ImplicitUsings>
    <IsBindingProject>true</IsBindingProject>
    <UseMaui>true</UseMaui>
    <MauiVersion>8.0.3</MauiVersion>
    <SkipValidateMauiImplicitPackageReferences>true</SkipValidateMauiImplicitPackageReferences>
  </PropertyGroup>
```

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

↓

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

以下の配置を追加します。

<UseMaui>true</UseMaui>

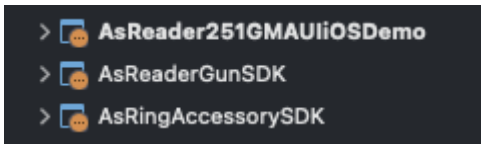
<MauiVersion>8.0.3</MauiVersion>

<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。
「AsReaderGun_SDK_Reference_Guide_for_iOS_Developers_ObjectiveC_JP_1_3」