

# AsReader®

.NET MAUI iOS SDK  
For ASR-032W

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

<b>1 SDK の追加</b> .....	<b>4</b>
1.1 SDK をインポートする .....	4
1.2 権限の追加 .....	11
<b>2 SDK の使用</b> .....	<b>12</b>
2.1 SDK のインポート .....	12
2.2 「AsReaderWiredBoxManagerDelegate」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 AsReader オブジェクトを初期化 .....	12
2.4 SearchingIP を実現 .....	12
2.5 ChangeNetworkState を実現 .....	12
2.6 ReceivedPCEPCData を実現 .....	12
2.7 ReceivedHIDData を実現 .....	13
2.8 InventoryStartedWithStatus を実現 .....	13
2.9 InventoryStoppedWithStatus を実現 .....	13
2.10 AsReader の検索／接続方法を設定 .....	13
2.11 AsReader の検索開始 .....	13
2.12 AsReader の検索停止 .....	13
2.13 接続 .....	13
2.14 切断 .....	13
2.15 スキャン開始 .....	14
2.16 スキャン停止 .....	14
2.17 HID データの受信開始 .....	14
2.18 HID データの受信停止 .....	14
2.19 RFID タグデータの受信開始 .....	14
2.20 RFID タグデータの受信停止 .....	14
<b>3 ビルドする.NETバージョンの変更</b> .....	<b>15</b>
<b>4 API の使用</b> .....	<b>18</b>

この文書の説明には、下記開発環境を使用しています。

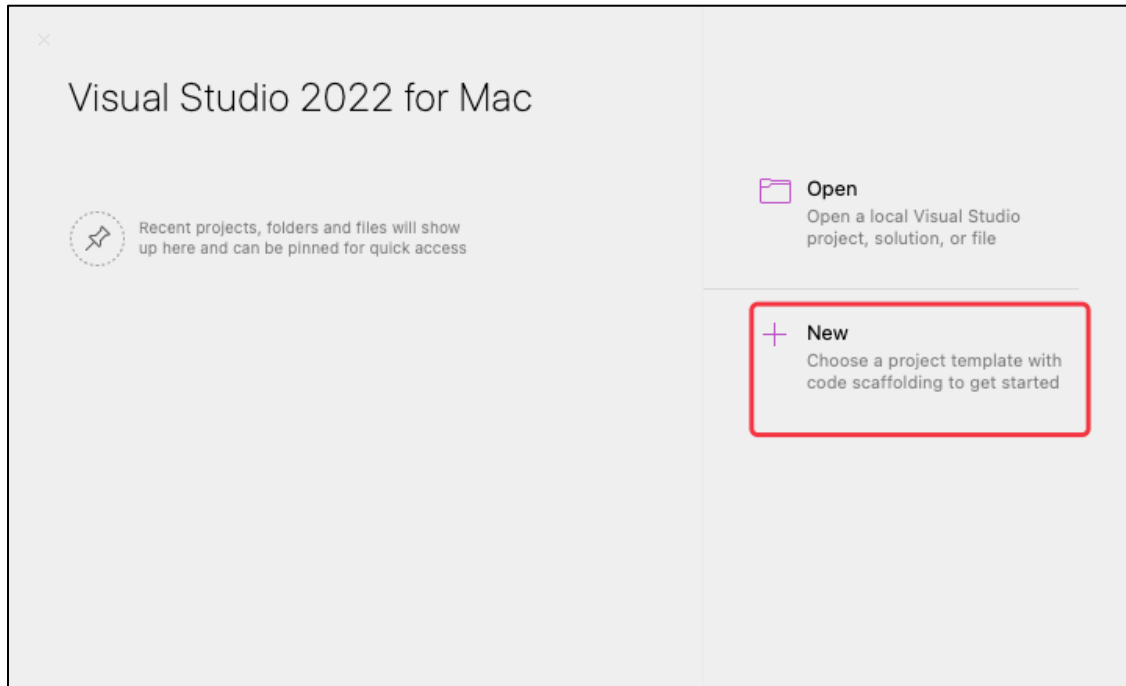
**開発環境：**

- ・ .NET バージョン : .NET 7.0
- ・ Visual Studio: Visual Studio Community 2022 for Mac 17.6.7 (build 417)
- ・ Xcode : 15.2

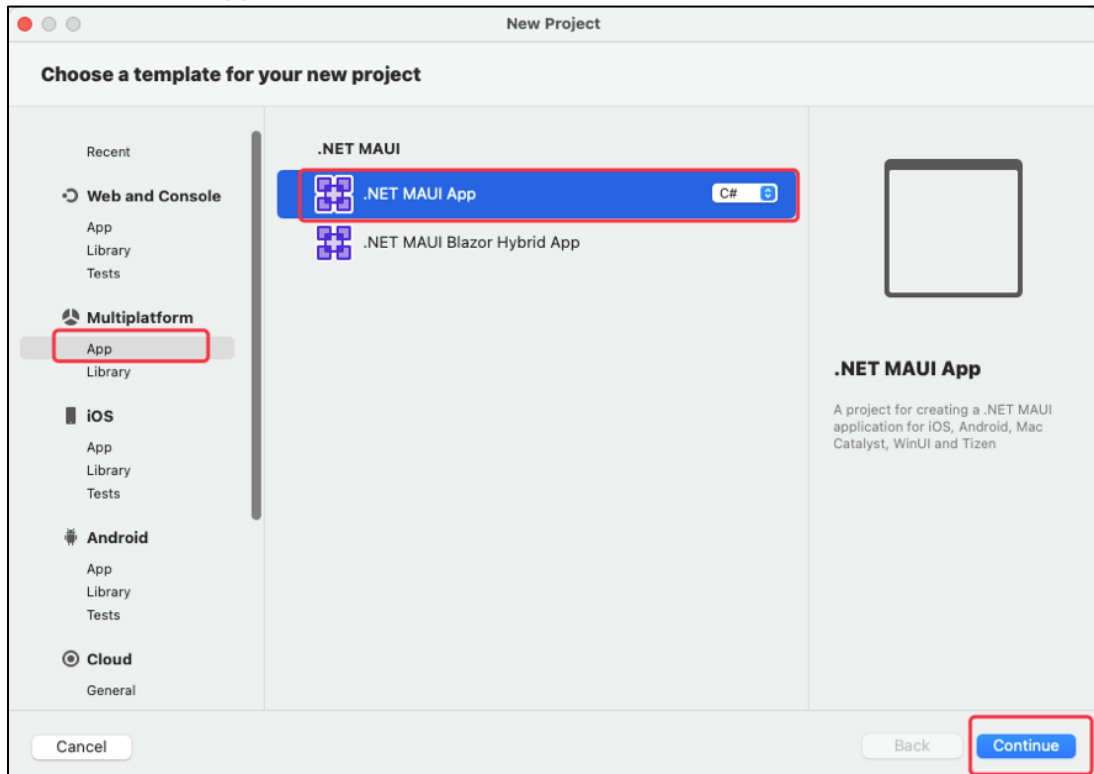
# 1 SDK の追加

## 1.1 SDK をインポートする

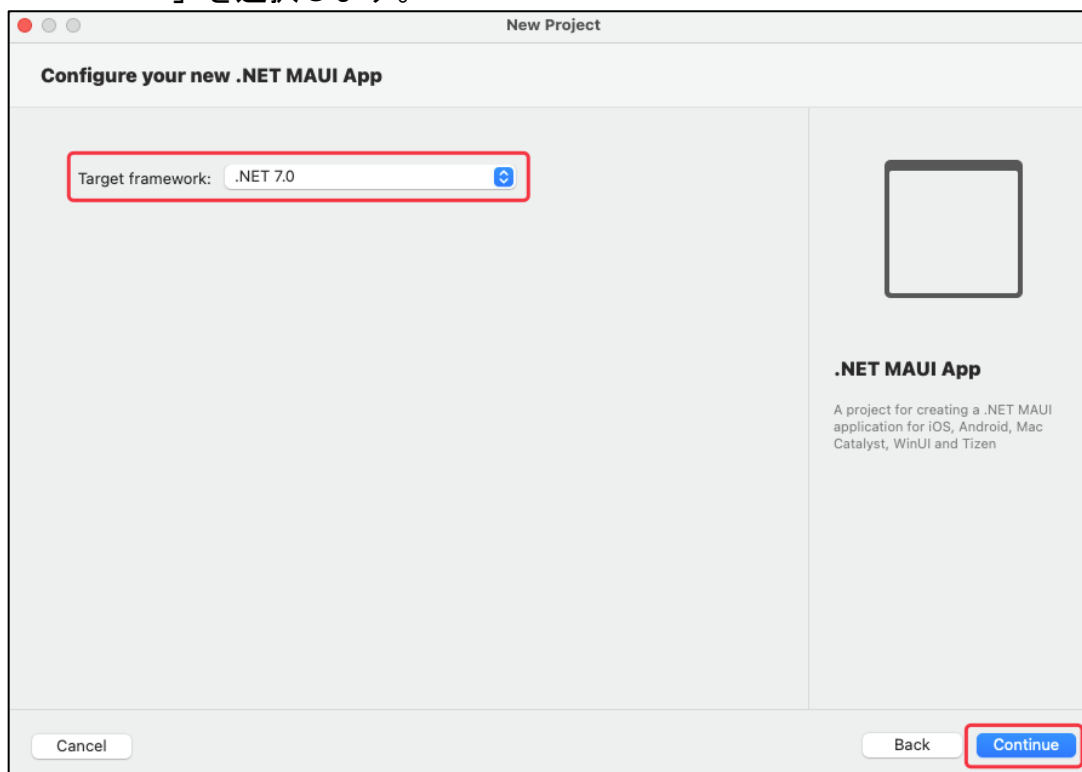
1. AsReaderWiredBoxTypeIOSSDK をローカルにダウンロードします。
2. プロジェクトを新規作成します。



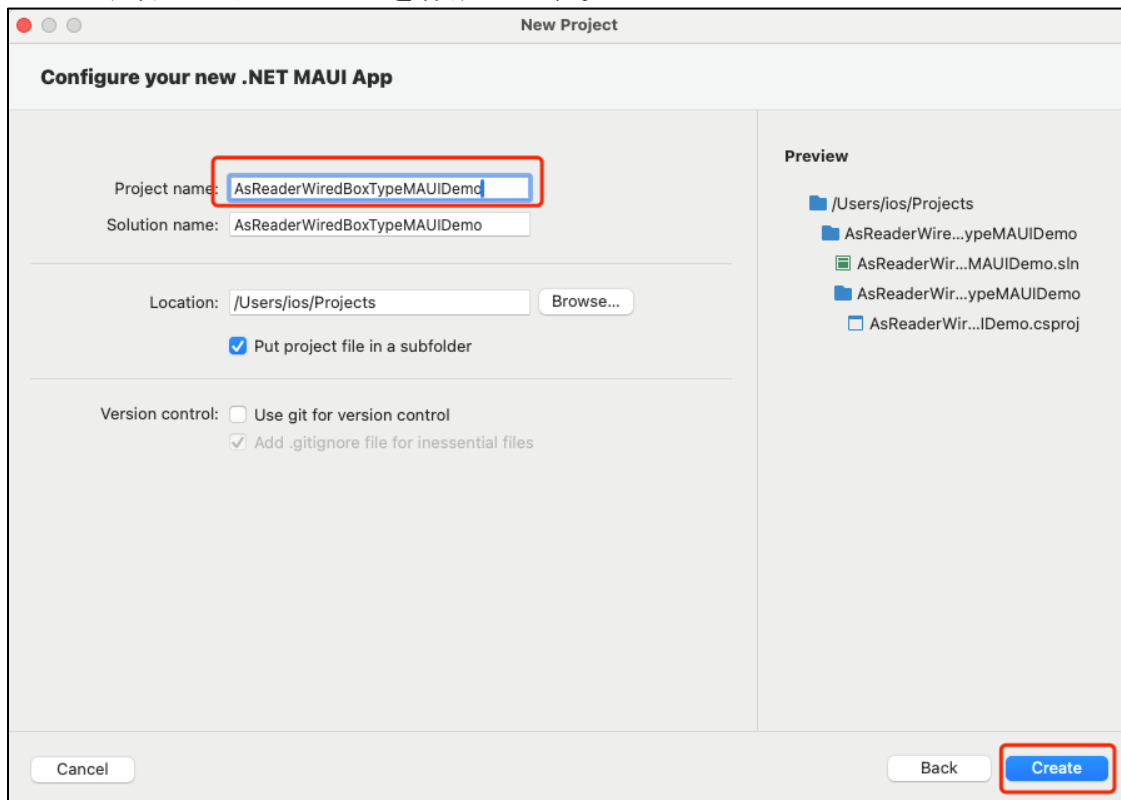
3. 「.NET MAUI App」を選択します。



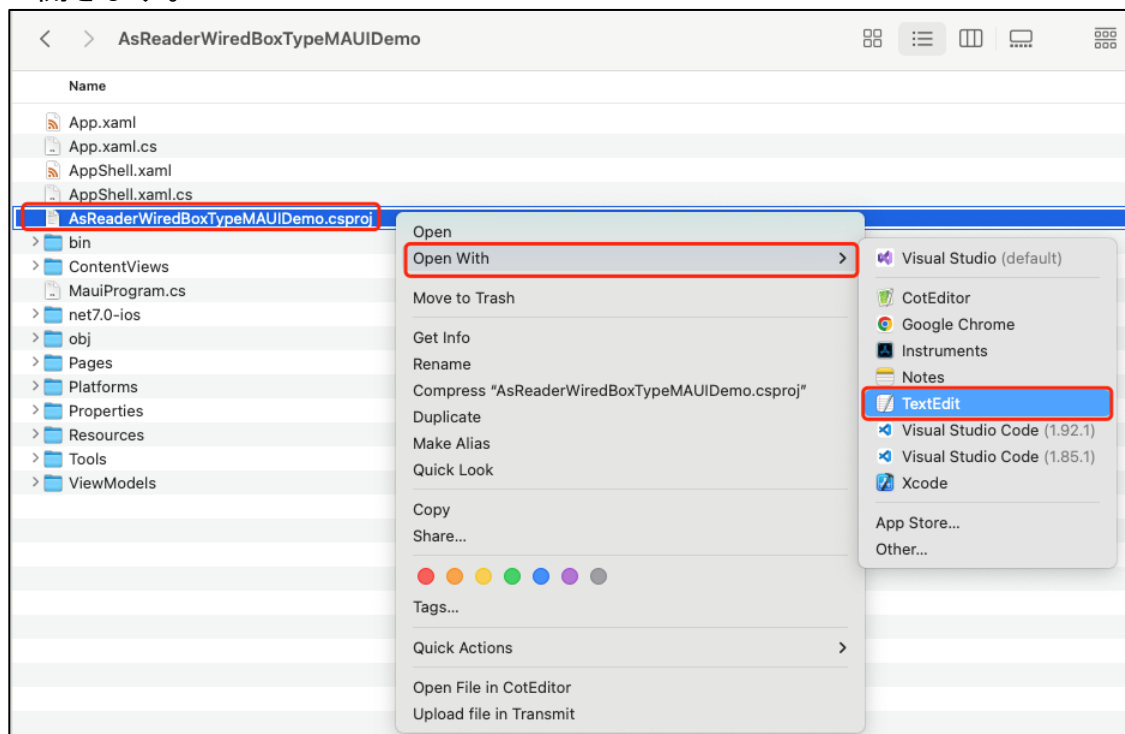
4. 「.NET 7.0」を選択します。



## 5. アプリ名とロケーションを作成します。



## 6. 「AsReaderWiredBoxTypeMAUIDemo.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ございません。)

```

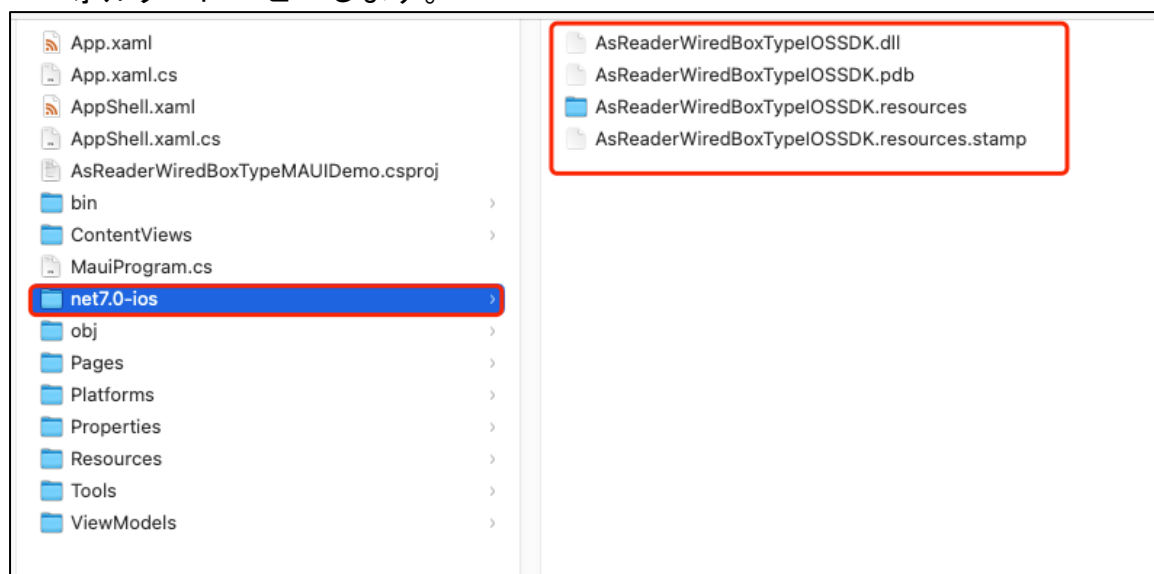
1 <Project Sdk="Microsoft.NET.Sdk">
2
3 <PropertyGroup>
4 <TargetFrameworks>net7.0-android;net7.0-ios;net7.0-maccatalyst</TargetFrameworks>
5 <TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows'))">$(TargetFrameworks);net7.0-windows10.0.19041.0</TargetFrameworks>
6 <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->
7 <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
8 <OutputType>Exe</OutputType>
9 <RootNamespace>AsReaderWiredBoxTypeMAUIDemo</RootNamespace>
10 <UseMaui>true</UseMaui>
11 <SingleProject>true</SingleProject>
12 <ImplicitUsings>enable</ImplicitUsings>
13
    
```

8. 削除すると、下図の赤枠内ようになります。

```

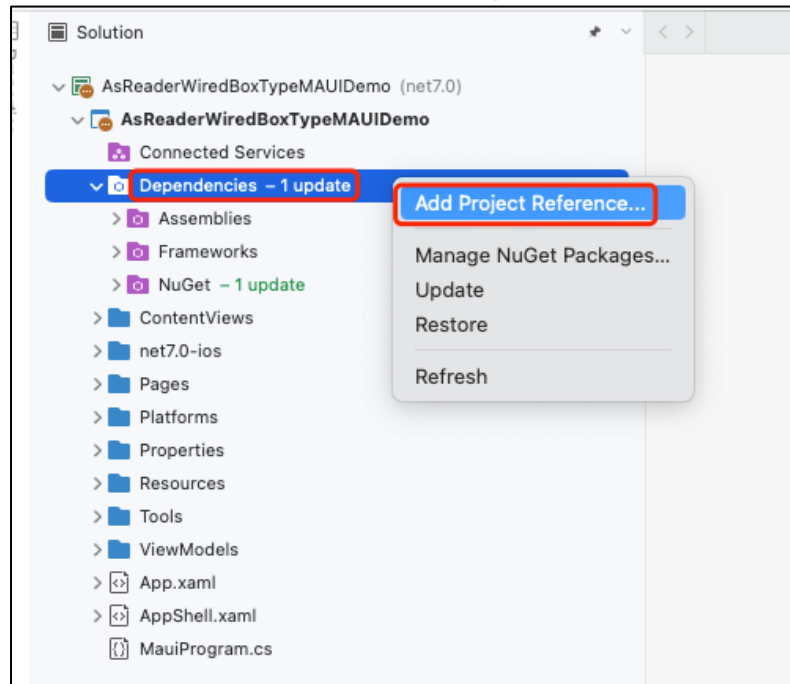
1 <Project Sdk="Microsoft.NET.Sdk">
2
3 <PropertyGroup>
4 <TargetFrameworks>net7.0-ios</TargetFrameworks>
5 <TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows'))">$(TargetFrameworks);net7.0-windows10.0.19041.0</TargetFrameworks>
6 <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->
7 <!-- <TargetFrameworks>$(TargetFrameworks);net7.0-tizen</TargetFrameworks> -->
8 <OutputType>Exe</OutputType>
9 <RootNamespace>AsReaderWiredBoxTypeMAUIDemo</RootNamespace>
10 <UseMaui>true</UseMaui>
11 <SingleProject>true</SingleProject>
12 <ImplicitUsings>enable</ImplicitUsings>
    
```

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

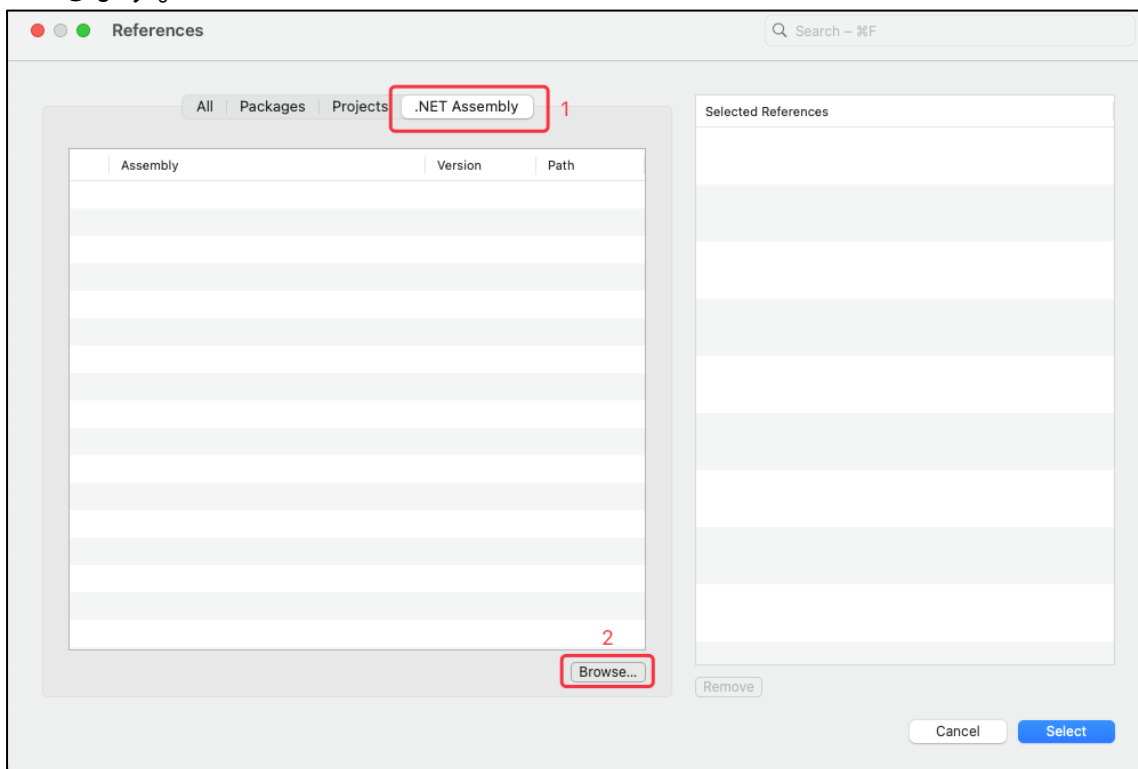




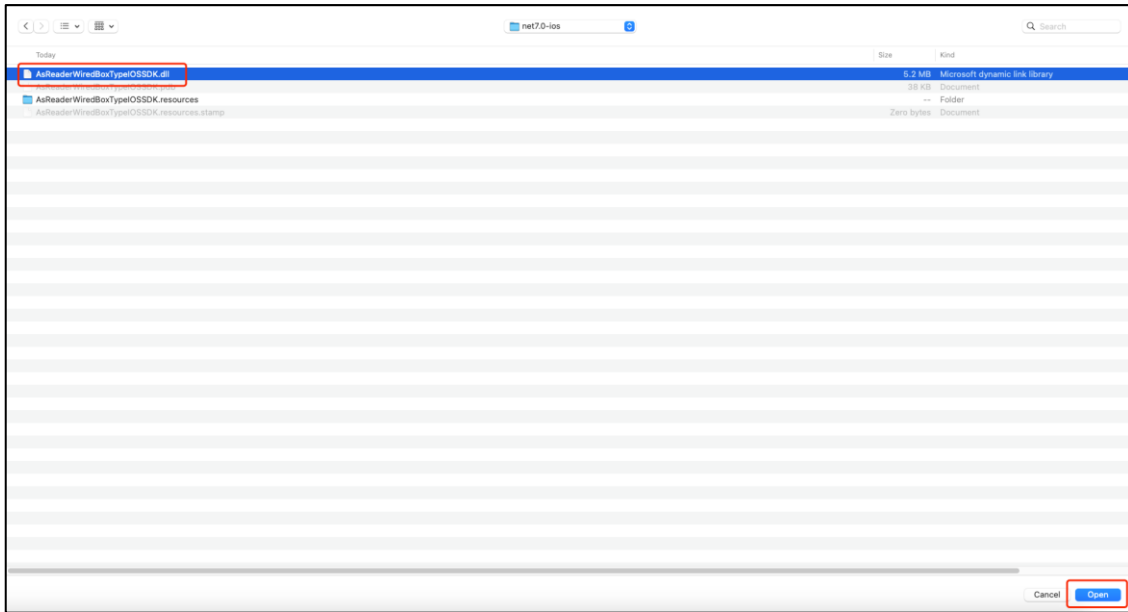
10. Dependencies -> 右クリック -> Add Project Reference...を選択します。



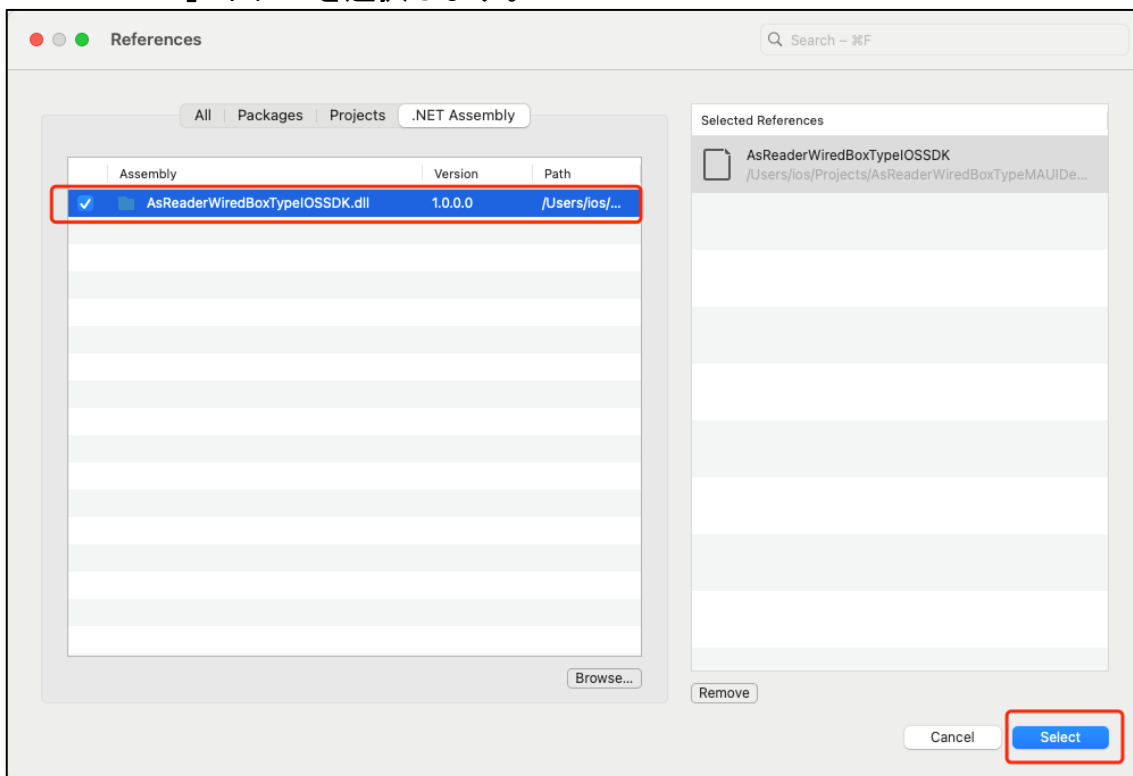
11. その後「.NET Assembly」（下図1）を選び、「Browse...」（下図2）を選択します。



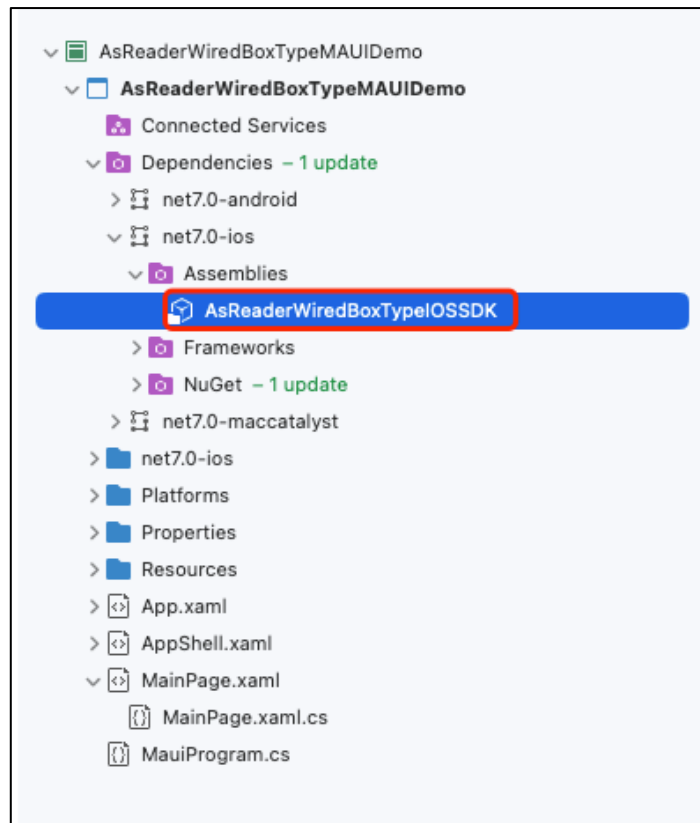
12. 「AsReaderWiredBoxTypeIOSSDK.dll」 (下図 1) を選び、「Open」 (下図 2) を選択します。



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



14. 「Assemblies」に「AsReaderWiredBoxTypeIOSSDK」（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ください。

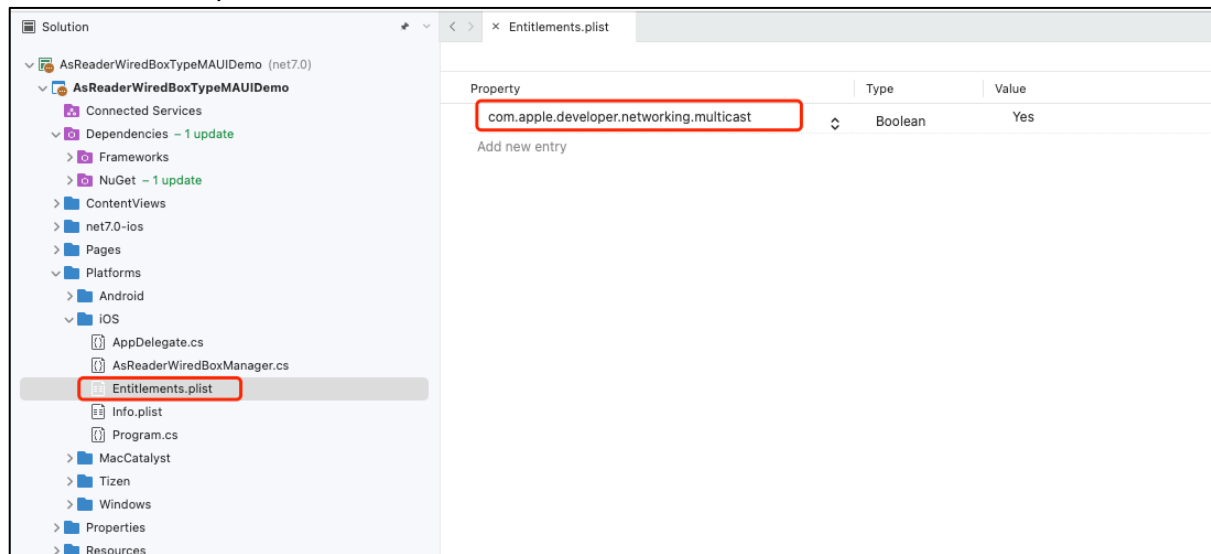
```

54     <CreatePackage>+false</CreatePackage>
55 </PropertyGroup>
56 <ItemGroup>
57     <!-- App Icon -->
58     <MauiIcon Include="Resources\AppIcon\appicon.svg" ForegroundFile="Resources\AppIcon\appiconfg.
59
60     <!-- Splash Screen -->
61     <MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />
62
63     <!-- Images -->
64     <MauiImage Include="Resources\Images\*" />
65     <MauiImage Update="Resources\Images\dotnet_bot.svg" BaseSize="168,208" />
66
67     <!-- Custom Fonts -->
68     <MauiFont Include="Resources\Fonts\*" />
69
70     <!-- Raw Assets (also remove the "Resources\Raw" prefix) -->
71     <MauiAsset Include="Resources\Raw\*" LogicalName="%(RecursiveDir)%(Filename)%(Extension)" />
72 </ItemGroup>
73
74 <ItemGroup>
75     <PackageReference Include="Microsoft.Extensions.Logging.Debug" Version="7.0.0" />
76 </ItemGroup>
77
78 <ItemGroup Condition="'$(TargetFramework)' == 'net7.0-ios'>
79     <Reference Include="AsReaderWiredBoxTypeIOSSDK">
80       <HintPath>net7.0-ios\AsReaderWiredBoxTypeIOSSDK.dll</HintPath>
81     </Reference>
82 </ItemGroup>
83 </Project>
84

```

## 1.2 権限の追加

Entitlements.plist 内にあるプロトコルを追加します。



## 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。

```
using AsReaderWiredBoxTypeIOSSDK;
```

### 2.2 「AsReaderWiredBoxManagerDelegate」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 iOS フォルダに作成することをお勧めします。

```
public class AsReaderWiredBoxManagerDelegate : AsReaderWiredBoxTypeDelegate
{
    public AsReaderWiredBoxManagerDelegate()
    {
    }

    public override void AsReaderWiredBoxTypeNetworkStateChanged(AsReaderWiredBoxType AsReaderWiredBoxType, AsReaderWiredBoxTypeSDKBOXNetworkState state, NSError error)
    {
        AsReaderWiredBoxManager.Shared().AsReaderWiredBoxTypeNetworkStateChanged?.Invoke(state, error);
    }

    public override void AsReaderWiredBoxTypeSearchingIP(AsReaderWiredBoxType AsReaderWiredBoxType, string ip, bool isFinish)
    {
        Console.WriteLine("SearchingIP ip:" + ip);
        AsReaderWiredBoxManager.Shared().searchingIP?.Invoke(ip, isFinish);
    }

    public override void AsReaderWiredBoxTypeReceivedPCEPCData(AsReaderWiredBoxType AsReaderWiredBoxType, NSData pcEpc, AsReaderWiredBoxTypeSDKBOXSelectAntenna antenna)
    {
        byte[] bytes = new byte[pcEpc.Length];
        bytes = pcEpc.ToArray();
        AsReaderWiredBoxManager.Shared().receivedPCEPCData?.Invoke(bytes, antenna);
    }

    public override void AsReaderWiredBoxTypeHIDDataReceived(AsReaderWiredBoxType AsReaderWiredBoxType, NSData HIDData)
    {
        byte[] bytes = new byte[HIDData.Length];
        bytes = HIDData.ToArray();
        AsReaderWiredBoxManager.Shared().receivedHIDData?.Invoke(bytes);
    }

    public override void AsReaderWiredBoxTypeInventoryStartedWithStatus(AsReaderWiredBoxType AsReaderWiredBoxType, AsReaderWiredBoxTypeSDKBOXStatusCode statusCode)
    {
        AsReaderWiredBoxManager.Shared().inventoryStartedWithStatus?.Invoke(statusCode);
    }

    public override void AsReaderWiredBoxTypeInventoryStoppedWithStatus(AsReaderWiredBoxType AsReaderWiredBoxType, AsReaderWiredBoxTypeSDKBOXStatusCode statusCode)
    {
        AsReaderWiredBoxManager.Shared().inventoryStoppedWithStatus?.Invoke(statusCode);
    }
}
```

### 2.3 AsReader オブジェクトを初期化

```
asReaderWiredBoxType = AsReaderWiredBoxType.SharedInstance;
asReaderWiredBoxManagerDelegate = new AsReaderWiredBoxManagerDelegate();
asReaderWiredBoxType.Delegate = asReaderWiredBoxManagerDelegate;
```

### 2.4 SearchingIP を実現

```
public override void AsReaderWiredBoxTypeSearchingIP(AsReaderWiredBoxType AsReaderWiredBoxType, string ip, bool isFinish)
{
}
}
```

### 2.5 ChangeNetworkState を実現

```
public override void AsReaderWiredBoxTypeNetworkStateChanged(AsReaderWiredBoxType AsReaderWiredBoxType, AsReaderWiredBoxTypeSDKBOXNetworkState state, NSError error)
{
}
```

### 2.6 ReceivedPCEPCData を実現

```
public override void AsReaderWiredBoxTypeReceivedPCEPCData(AsReaderWiredBoxType AsReaderWiredBoxType, NSData pcEpc, AsReaderWiredBoxTypeSDKBOXSelectAntenna antenna)
{
}
```

## 2.7 ReceivedHIDData を実現

```
public override void AsReaderWiredBOXTYPEHIDDataReceived(AsReaderWiredBOXTYPE AsReaderWiredBOXTYPE, NSData HIDData)
{
}
```

## 2.8 InventoryStartedWithStatus を実現

```
public override void AsReaderWiredBOXTYPEInventoryStartedWithStatus(AsReaderWiredBOXTYPE AsReaderWiredBOXTYPE, AsReaderWiredBOXTYPESDKBOXStatusCODE statusCode)
{
}
```

## 2.9 InventoryStoppedWithStatus を実現

```
public override void AsReaderWiredBOXTYPEInventoryStoppedWithStatus(AsReaderWiredBOXTYPE AsReaderWiredBOXTYPE, AsReaderWiredBOXTYPESDKBOXStatusCODE statusCode)
{
}
```

## 2.10 AsReader の検索／接続方法を設定

```
AsReaderWiredBOXTYPEConnectMethod method = AsReaderWiredBOXTYPEConnectMethod.Tcp;
asReaderWiredBOXTYPE.ConnectMethod = method;
```

asReaderWiredBOXTYPE : AsReaderWiredBOXTYPE クラスのインスタンスオブジェクト

## 2.11 AsReader の検索開始

```
asReaderWiredBOXTYPE.StartSearchDevice;
```

## 2.12 AsReader の検索停止

```
asReaderWiredBOXTYPE.StopSearchDevice;
```

## 2.13 接続

```
asReaderWiredBOXTYPE.ConnectServer(ip, port);
```

## 2.14 切断

```
asReaderWiredBOXTYPE.DisconnectServer;
```

## 2.15 スキャン開始

```
asReaderWiredBOXType.StartInventoryWithTagsMaxTagCount(  
    mtNu,  
    mtime,  
    repeatCycle,  
    enable1,  
    enable2,  
    enable3,  
    enable4,  
    enable5,  
    enable6,  
    enable7,  
    enable8);
```

## 2.16 スキャン停止

```
asReaderWiredBOXType.StopInventory;
```

## 2.17 HID データの受信開始

```
asReaderWiredBOXType.StartReceiveHIDData;
```

## 2.18 HID データの受信停止

```
asReaderWiredBOXType.StopReceiveHIDData;
```

## 2.19 RFID タグデータの受信開始

```
asReaderWiredBOXType.StartReceiveRFIDData;
```

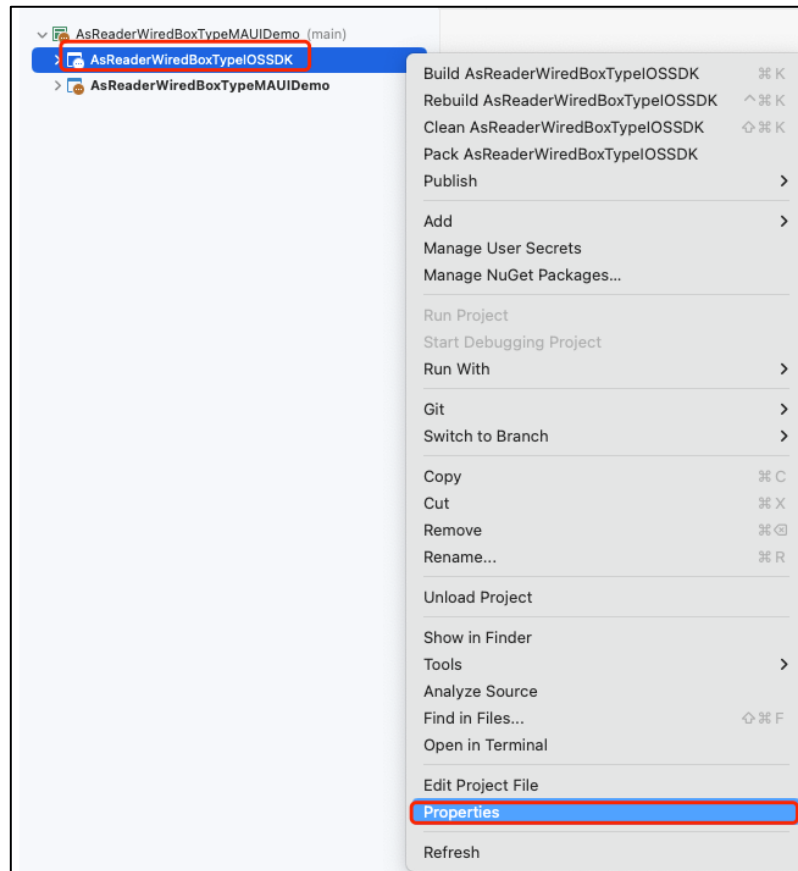
## 2.20 RFID タグデータの受信停止

```
asReaderWiredBOXType.StopReceiveRFIDData;
```

### 3 ビルドする.NET バージョンの変更

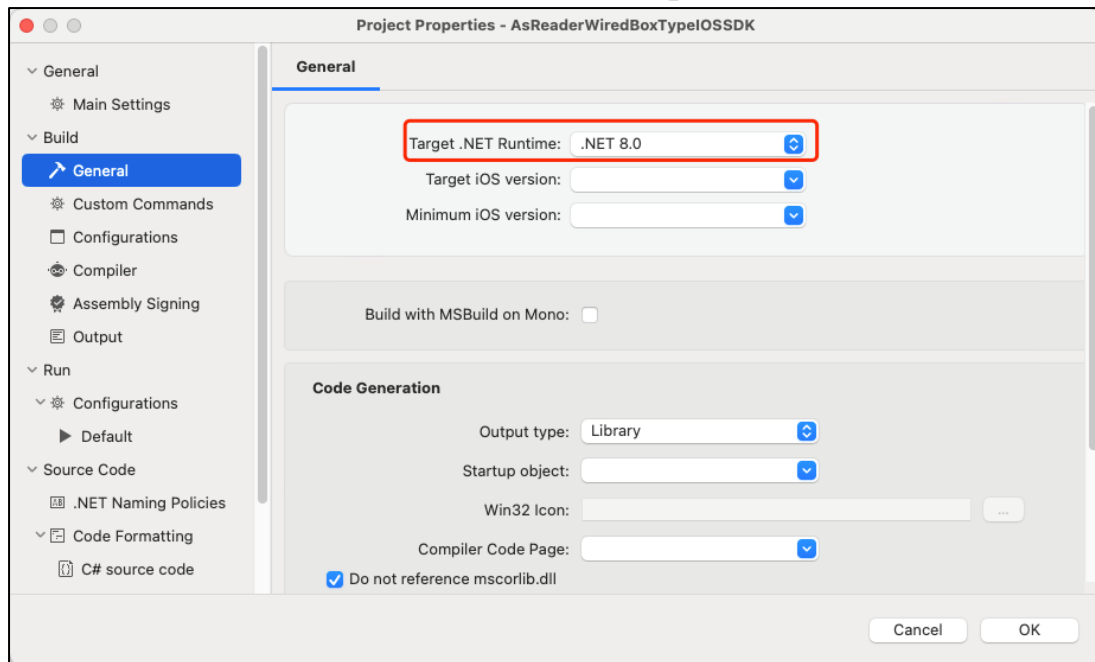
「.NET 7.0」を「.NET 8.0」に変更する方法です。

#### 1. プロジェクトを右クリック→プロパティ

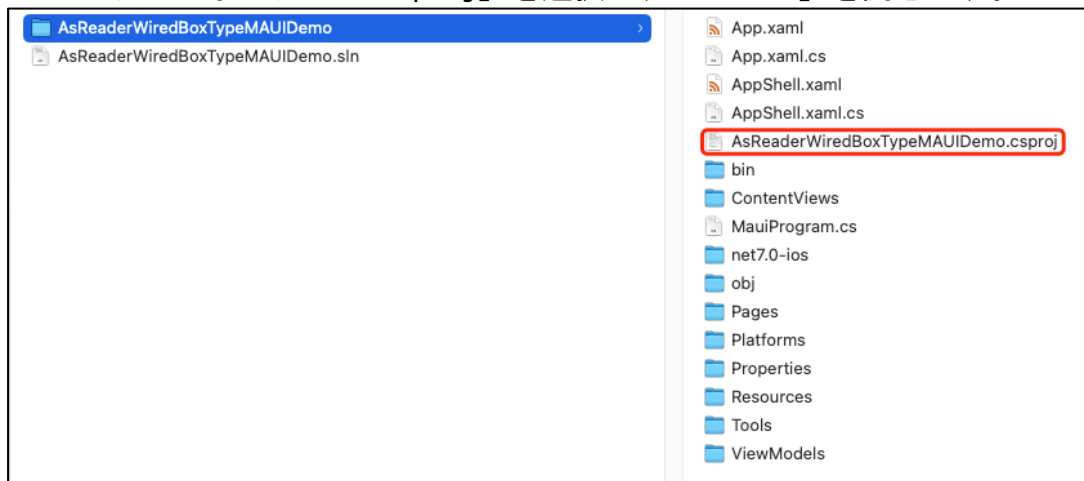




2. 全般->ターゲット.NET ランタイムを「.NET 8.0」に選択します。



3. プロジェクトフォルダの「.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。

```
× AsReaderWiredBoxTypeMAUIDemo.csproj
1  <Project Sdk="Microsoft.NET.Sdk">
2
3      <PropertyGroup>
4          <TargetFrameworks>net8.0-ios</TargetFrameworks>
5          <OutputType>Exe</OutputType>
6          <RootNamespace>AsReaderWiredBoxTypeMAUIDemo</RootNamespace>
7          <UseMaui>true</UseMaui>
8          <SingleProject>true</SingleProject>
9          <ImplicitUsings>enable</ImplicitUsings>
10         <Nullable>enable</Nullable>
11     </PropertyGroup>
</Project>
```

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

↓

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

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、Objective-C の SDK リファレンスガイドを参照してください。

「WiredBOX-Type\_SDK\_Reference\_Guide\_ObjectiveC\_JP\_x\_x」