

はつねの日記

Kinect, Windows 10 UWP, Windows Azure, IoT, 電子工作

30

2020-04

Unityで作成したアプリをMicrosoftストアで配布するには

[Windows 10 Unity](#)

事前準備

Unityでビルドされるexe名は、あらかじめMicrosoft Storeで名前の予約をしておきます。

例) HoloLabStoreCheck2D

Unityアプリ側での設定

Build Settingsを開いて、

Platformで「PC, Mac & Linux Standalone」を選択して、Target Platformに「Windows」を指定、Architectureは「x86_64」を選択します。

これで、配布ターゲットは、Windows 10 x64のみとなります。

必要なのはExeや実行に必要なDLLそしてデータですので、Create Visual Studio Solutionは不要です。

Player SettingでProduct Nameにストアで予約した名前にします。

「↓」タブの中のIconなど必要な設定をします。

BuildボタンをクリックしてExeの生成先をUnityのプロジェクトフォルダ配下（例:Buildフォルダ）に指定します。

以降の説明では、Unityプロジェクトフォルダ直下にBuildフォルダがあることを前提とします、

作成先のフォルダの内容を確認します。

例)

HoloLabStoreCheck2D.exe

UnityCrashHandler64.exe

UnityPlayer.dll

WinPoxEventRuntime.dll

HoloLabStoreCheck2D_Dataフォルダ

MonoBleedingEdge フォルダ

Visual StudioでWindowsアプリケーションパッケージプロジェクトを作成

Visual Studioを立ち上げて、Windowsアプリケーションパッケージプロジェクトを新規作成します。

プロジェクト名は、UnityのExe名+AppPackageを推奨します（例：HoloLabStoreCheck2DAppPackage）

場所は、Unityプロジェクトフォルダを指定します（以降の説明は、この位置を指定している前提となります）。

「ソリューションとプロジェクトを同じディレクトリに配置する」チェックは外したままです。

でパス入ります。

ターゲットバージョンは、できれば最新のバージョンを指定するのがいいでしょう。

最小バージョンはビルド14393（バージョン1607）以降を指定する必要があります。2020/04/30現在の

Windows 10サポートは1803以降なので、1809以降を指定するのも1案です。

※LTSC/LTSBを考慮するならば、ビルド14393（バージョン1607）のLTSCの延長サポート終了日が2026/10/13なのでそれを最小バージョンとするという戦略もあります。

Unityで作成したexeを直接指定できないので中間プロジェクトを作成

次に、.NET Frameworkのコンソールアプリプロジェクトを追加します。

このあたりから正攻法から外れた邪法に突入しますので、本当はWindowsアプリケーションパッケージプロジェクトのテンプレートあたりでUnityアプリの配布もサポートしてくれるといいんじゃないかと思います。プロジェクト名はUnityのExe名と必ず同じ（例：HoloLabStoreCheck2D）にします。

プロジェクトが追加されたらVisual Studioの[ビルド]-[構成マネージャー]で

- アクティブソリューション構成：Release
- アクティブソリューションプラットフォーム：x64

を選択します。

そうするとコンソールアプリのプラットフォームが「Any CPU」のままなので、x64を追加して選択しておきます。

パッケージ対象のプロジェクトとして参照

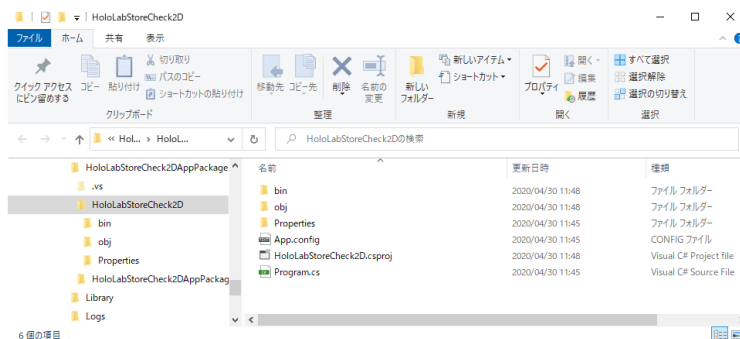
次にソリューションエクスプローラーで、Windowsアプリケーションパッケージプロジェクトの「アプリケーション」を右クリックして[参照の追加]で、追加したコンソールアプリケーションプロジェクトを参照します。

上記が完了したら、このあたりで、[ファイル]-[すべて保存]で、一度保存しておきましょう。

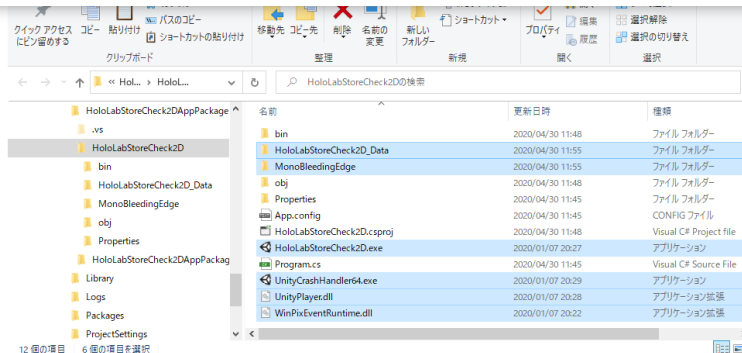
中間プロジェクトにUnityアプリの実行ファイルをコピー

コンソールアプリケーションプロジェクト名を右クリックして、[エクスプローラーでフォルダを開く]をします。

エクスプローラーでの表示例：



このフォルダにUnityでビルドした中身を丸ごとコピーします。



中間プロジェクトビルド時にUnityアプリの実行ファイルが使われるように調整

ソリューションエクスプローラーでコンソールアプリケーションプロジェクト名を指定して[すべてのファイルを表示]します。

さらに

HoloLabStoreCheck2D_Dataフォルダ

MonoBleedingEdge フォルダ

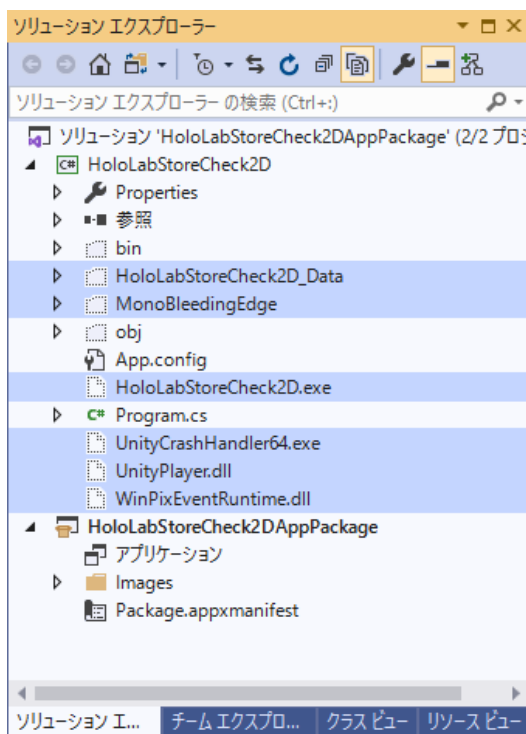
HoloLabStoreCheck2D.exe

UnityCrashHandler64.exe

UnityPlayer.dll

WinPixEventRuntime.dll

を選択して右クリックメニューで[プロジェクトに含める]をクリックします。



これでUnityアプリが必要としているファイルやフォルダ構造がプロジェクトに取り込まれました。

更にプロジェクトに含めたexeやdll、フォルダ内のファイル全てのプロパティを次のように指定します。

- ビルドアクション：なし
- 出力ディレクトリにコピー：常にコピーする

コンソールアプリをビルドします。

もし、bin\x64\Releaseフォルダではなくbin\Releaseフォルダに出力されていたらプラットフォームがAny CPUのままなので再確認をしてください。

この状態でexeファイルをダブルクリックして実行してもコンソールアプリが実行されます。つまり、Unityで作成した実行ファイルとコンソールアプリケーションプロジェクトでビルドされるExe名が同じなので出力フォルダでは上書きされてしまうのです。

これを次に解決します。

中間プロジェクトビルド時で作成されたexeファイルをUnityアプリのexeファイルに変更

そこで、ソリューションエクスプローラーからコンソールアプリケーションプロジェクトのプロパティを開いて、[ビルドイベント]-「ビルド後イベントのコマンドライン」に以下をいれて、上書きされた本来のExeを更に本来のものに上書きし直します。

```
xcopy /Y "$(ProjectDir)$(TargetFileName)" "$(TargetDir)"
del /Q "$(TargetDir)$(TargetName).pdb"
```

再度、bin\x64\Releaseフォルダ上のexeを起動してみましょう。

作成したUnityアプリが起動するはずです。

無事に起動出来たら、Windowsアプリケーションパッケージプロジェクトに情報を伝えるためのダミープロジェクトの構成がまずは完了しました。

Microsoft Storeに登録

ここから先は、WPFアプリなどをAppPackage化してMicrosoft Storeに登録するのと同じ流れになります。身近に、Windows Phoneアプリ、Windows Storeアプリ、UWPアプリのストア配布経験者がいらっしゃるようであれば、ぜひ、その方に以降の手順を一読いただいて以降の説明をショートカットするチートルートを進んでいただければと思います。

Package.appxmanifestの編集（AppPackage化特有事項）

WindowsアプリケーションパッケージプロジェクトにあるPackage.appxmanifestをテキストエディタやVisual Studio Codeで開くと下記の部分があります。

```
<Dependencies>
<TargetDeviceFamily Name="Windows.Universal" MinVersion="10.0.0.0" MaxVersionTested="10.0.0.0" />
<TargetDeviceFamily Name="Windows.Desktop" MinVersion="10.0.14393.0"
MaxVersionTested="10.0.14393.0" />
</Dependencies>
こちらからWindows.Universalの行を削除して保存しましょう。
<Dependencies>
<TargetDeviceFamily Name="Windows.Desktop" MinVersion="10.0.14393.0"
MaxVersionTested="10.0.14393.0" />
</Dependencies>
```

Package.appxmanifestの内容設定（UWPアプリと同等項目）

次にソリューションエクスプローラーでPackage.appxmanifestをダブルクリックして内容を編集します。表示名は、元々とのExe名にAppPackageが付いた形になっているので適切な名前にします。

例：HoloLabStoreCheck2DAppPackage→HoloLabStoreCheck2D for Desktop

説明もきちんと記載しましょう。

パッケージ化のところにもパッケージ表示名があるのでそちらも同様（例：

HoloLabStoreCheck2DAppPackage→HoloLabStoreCheck2D）に編集します。

[ビジュアル資産]のところにアプリのアイコン画像など必要な解像度のアイコンを作成します。

大きめのアイコンファイルを作成して、[生成]ボタンで一気に必要な解像度の画像を自動生成するのも良いでしょう。

アイコンは背景を透過pngにしておき、タイルの背景やスプラッシュスクリーンの背景に色を指定してください。

もしくは、アイコンのイメージ部分を白で作成し、背景指定色をtransparentを指定してシステムカラーを使うようにするのも、それっぽい感じの仕上げになります。

試しにローカルインストールを確認

コンソールアプリケーションプロジェクトを右クリックして[リビルド]します。

Windowsアプリケーションパッケージプロジェクトを右クリックして[配置]を実行します。

うまく配置がいけば、Windowsのスタートの中にアプリが出現します（例：HoloLabStoreCheck2D）。

配置したアプリを実行してUnityアプリが起動すれば、ストア登録前の確認は終了です。

忘れずにアンインストールして置きましょう。

Microsoft Storeに登録

Windowsアプリケーションパッケージプロジェクトを右クリック[公開]-[アプリパッケージの作成]をクリックします。

配布方法は、アプリ名は予約済ですが[新しいアプリ名でMicrosoft Storeに]を選択します。

ストアに登録するアカウントを追加して、[既存のアプリケーション名]リストから、事前に予約した名前を選択します。

バージョンは[自動的に増加]チェックは外して、アーキテクチャはx64のみチェックします。ソリューション構成も「Release (x64)」になっていることを確認して、[作成]をクリックします。

成功すれば次のような出力となります。

```
1>パッケージが正常に作成されました。
===== 公開: 1 正常終了、0 失敗、0 スキップ =====
パッケージが Release (x64) 向けに正常にビルドされました。
===== パッケージ: 1 正常終了、0 失敗 =====
アプリケーション バンドルが Release (x64) 向けに正常にビルドされました。
===== アプリケーション バンドル: 1 正常終了、0 失敗 =====
```

出力先フォルダはプロジェクトフォルダ内のAppPackagesフォルダになります。この中にあるmsixuploadフォルダをストアに登録します。

例) HoloLabStoreCheck2DAppPackage_1.0.0.0_x64_bundle.msixupload

ストア登録時の注意表示について（AppPackage化特有事項）

Windowsアプリケーションパッケージプロジェクトは自動的にrunFullTrustがついているので下記の注意書き（次の制限された機能は、アプリで使用する前に承認が必要です）が表示されますが、気にしないで大丈夫です。

runFullTrust.

HoloLabStoreCheck2D

アプリケーション

アプリケーションの概要

申請

アドオン

分析

製品管理

サービス

HoloLabStoreCheck2DAppPackage_1.0.0.0_x64_bundle.msixupload 21.8 MB

パッケージ受領の検証に関する警告: The following restricted capabilities require approval before you can use them in your app: runFullTrust.

デバイス ファミリの利用可否

この表は、特定の Windows 10 デバイス ファミリ (該当する場合は以前の OS バージョンも含む) に対して提供されるパッケージを、ランク付けされた順序で示しています。デバイス ファミリのチェック ボックスがオフになっている場合、その種類のデバイスを使う新しいユーザーは、アプリを入手することはできません (ただし、アプリを既に所有しているユーザーは引き続きそのアプリを使用でき、今後申請される更新プログラムも取得できます)。

☒ このアプリを将来のデバイス ファミリで使用できるようにするかどうかを Microsoft が決定できる

☒

Windows 10 デスクトップ

☐

Windows 10 Mobile

☐

Windows 10 Xbox

☐

Windows 10 Team

☐

Windows 10 Holographic

パッケージ				
HoloLabStoreCheck2DAppPackage_1.0.0.0_x64_bundle.msixupload v1.0.0.0, Neutral	1			

1

最初にこのデバイス ファミリに提供されます。

2

デバイスが上位パッケージをサポートできない場合に提供されます。

このデバイス ファミリでパッケージを利用できるようにするには、チェック ボックスをオンにします。

ストア登録自体は一般的なものと同じですが、申請オプションの項目にある制限付き機能については、下記のようなコメントを記載します。

デスクトップアプリのブリッジを使ったAppPackage化によるストア登録なのでrunFullTrustが必要です。
Because it is a store registration by AppPackage using the bridge of the desktop application, runFullTrust is necessary.

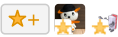
あとは申請に提出してストアに公開されるのを待つだけです。

以前とは異なり数時間くらいで審査通過してストア公開フェーズになります。

ストア公開フェーズになってから実際に公開されるまで数時間待たされるので翌朝には公開されているかなくらいの感覚でいるとイライラしないでしょう。

もちろんストア公開なので有料アプリもサクッと指定ができてアプリ費用もMSさんが回収してくれる（マージンは取られるけれど）ので、これはっ！というUnityアプリを登録してみるのもいいでしょう！

初音玲 (id:hatsune_a) 251日前



0

8

ツイート

シェア

