

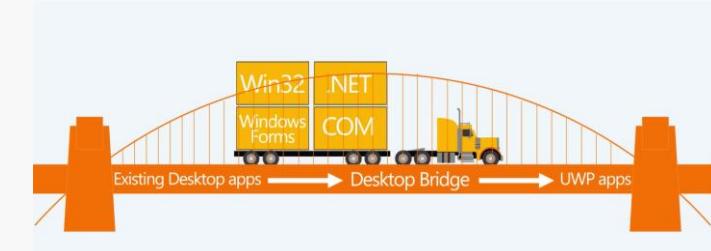


Desktop Bridge: ハンズオントレーニング

日本マイクロソフト株式会社
デベロッパー & エバンジェリズム統括本部

目次

- 準備
- ハンズオン環境
- 演習 1 DAC を使った変換
- 演習 2 Appx への署名
- 演習 3 アプリのインストール
- 演習 4 PRI リソースの作成（オプション）
- 演習 5 開発環境でのアプリのインストール
- 演習 6 Appx パッケージの作成 1
- 演習 7 Appx パッケージの作成 2



はじめに

目的

この手順書は、Desktop App Converter でインストーラーを変換するためのものです。

実習内容

変換

テスト

Appx の作成

インストールしてテスト

準備（ハンズオン環境構築）

必要なソフトウェア

Windows 10 Version 1607 Professional 以上

Desktop App Converter

<http://aka.ms/converter>

Windows SDK

<https://developer.microsoft.com/windows/downloads/windows-10-sdk>

ベースイメージ

<http://aka.ms/converterimages>

必要なハードウェア

ハードウェアの仮想化機構 (Second Level Address Translation) が必要

Coreinfo ユーティリティで確認します

<http://technet.microsoft.com/ja-jp/sysinternals/cc835722>

```
C:\$Sysinternals>coreinfo -v
Coreinfo v3.4 - Dump information on system CPU and
省略
HYPERVISOR * Hypervisor is present
VMX      - Supports Intel hardware-assisted virtualization
EPT      - Supports Intel extended page tables (SLAT)
```

ベース イメージのインストール

```
C:> DesktopAppConverter.exe -Setup -BaseImage  
BaseImage-14393.wim
```

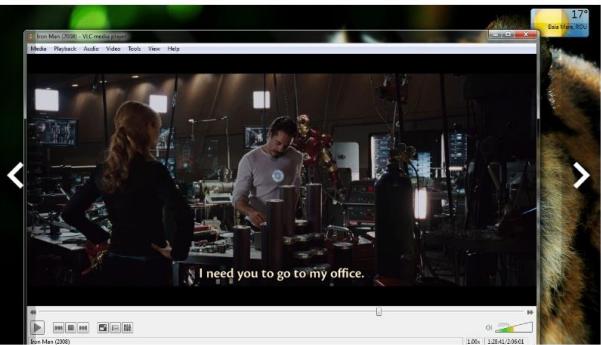
- ホスト OS のビルドと同じイメージを使用
- -Setup オプションは 1 度だけ実施すればよい

変換するインストーラー

VLC メディア プレーヤー

<https://www.videolan.org/vlc/>

よりダウンロードしておきます



VideoLAN, このプロジェクト、および **非営利の組織について**。

VLC media player

VLCはフリーなマルチプラットフォーム対応のマルチメディアプレイヤーであり、DVD、オーディオCD、VCDや様々なストリーミングプロトコルを再生可能なフレームワークです。

ダウンロード VLC

Version 2.2.4 • Windows • 28 MB
141,094,215 までダウンロード

Windows Mac Linux Android

ハンズオン 環境

ハンズオンのフォルダ構造

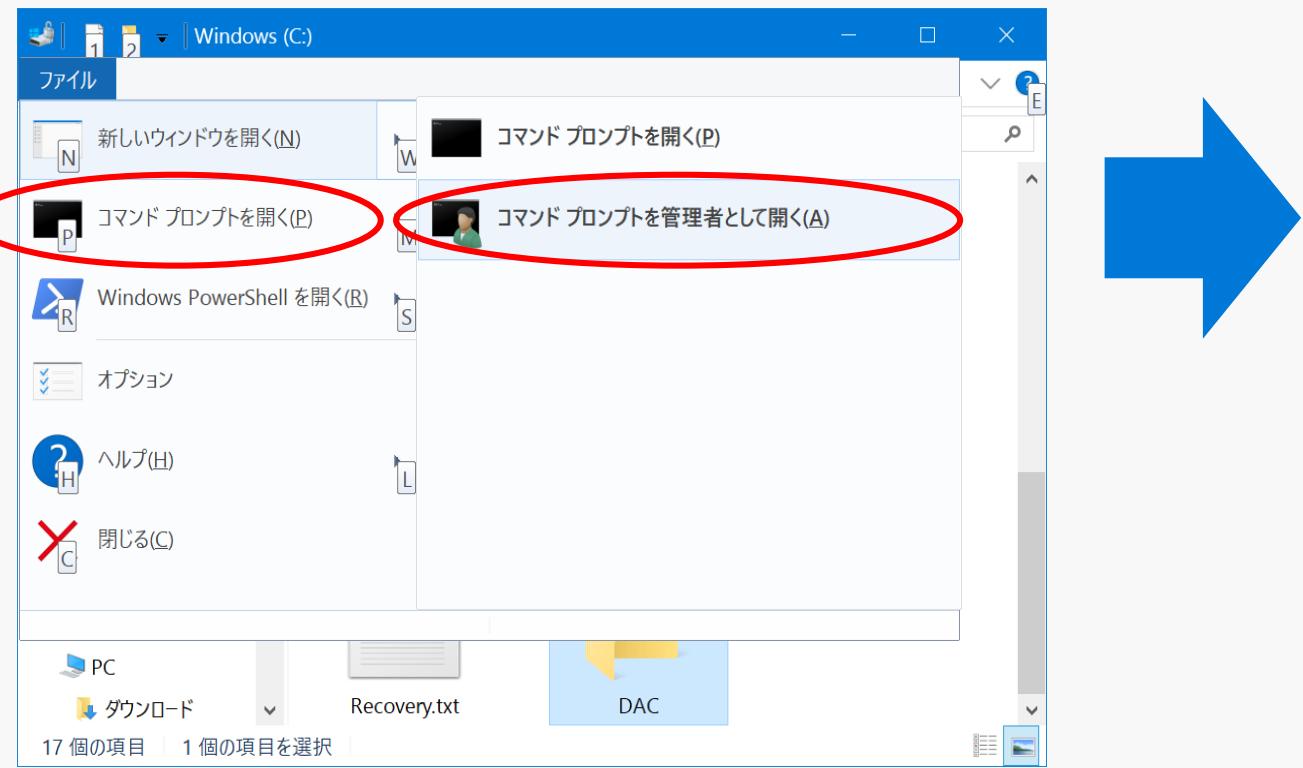
次のフォルダ構造を想定しています。

C:\DAC ━━━━━━ コマンド操作のフォルダ
 └ \Input ━━ vlc-2.2.2-win32.exe

演習 1 DAC を使った変換

コマンドプロンプトを開く

管理者権限でコマンドプロンプトを開き、
C:¥DAC をカレントフォルダにします



The screenshot shows a command prompt window with the title 'Administrator: C:\WINDOWS\system32\cmd.exe'. The window displays the text 'C:¥DAC>', indicating the current working directory.

変換コマンドを入力します

```
C:\¥DAC> DesktopAppConverter.exe -Installer  
".¥input¥vlc-2.2.4-win32.exe" -InstallerArguments  
"/S /L=1041" -Destination "." -PackageName "VLC" -  
Publisher "CN=videoLAN" -Version 2.2.4.0 -MakeAppx  
-Verbose
```

Installer	インストーラー (EXE や MSI など)
InstallerArguments	サイレントインストールのためのパラメータ
Destination	出力先フォルダ
PackageName	パッケージ名で、Appx のファイル名になります
Publisher	必ず CN=xxxxx 形式 (証明書の組織名)
Version	Major . Minor . Build . Revision 形式
MakeAppx	DAC v 1.0.6 では指定しておきます

変換コマンドの実行後

変換が成功すると、次のようなログが表示されます

Output Paths:

Name

CertPath

PackageFilesPath

PackagePath

LogFilePath

Value

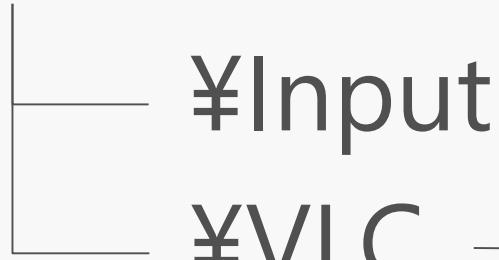
.¥VLC¥PackageFiles

.¥VLC¥VLC.appx

.¥VLC¥logs

成功時のフォルダ構造

C:¥DAC



¥VLC ————— Appx が 出力 さ れ ま す



変換した Appx レイアウトになります

演習 1 のまとめ

Desktop App Converter を使った、基本的な変換方法を学習しました

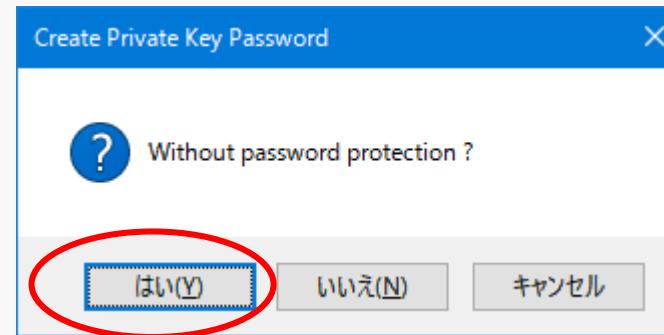
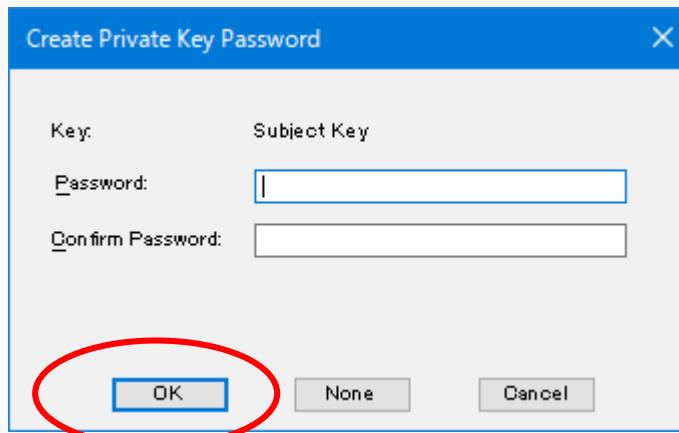
演習で使用していないパラメータについては、ドキュメントを参照してください

<https://msdn.microsoft.com/ja-jp/windows/uwp/porting/desktop-to-uwp-run-desktop-app-converter>

演習 2 Appx への署名

証明書を作成します

```
C:\$DAC> CD VLC  
C:\$DAC\$VLC> SET PATH="C:\Program Files  
(x86)\Windows Kits\10\bin\x86";%PATH%  
C:\$DAC\$VLC> makecert.exe -r -h 0 -n "CN=videoLAN" -  
eku 1.3.6.1.5.5.7.3.3 -pe -sv VLC.pvk VLC.cer  
Succeeded
```



MakeCert コマンドによるパスワードは、空欄にします

証明書 (pfx) より鍵のコピーを作成します

```
C:\¥DAC¥VLC> pvk2pfx.exe -pvk VLC.pvk -spc VLC.cer -  
pfx VLC.pfx
```

C:\¥DAC¥VLC フォルダには、
VLC.pfx , VLC.cer , VLC.pvk の 3 種類ができます

Appx に署名します

```
C:\¥DAC¥VLC> signtool.exe sign -f VLC.pfx -fd SHA256  
-v VLC.appx
```

The following certificate was selected:

Issued to: videoLAN

Issued by: videoLAN

Expires: Sun Jan 01 08:59:59 2040

SHA1 hash: 3E79B5CC8CA6289F17BC5BF6E436815C18090E1C

Done Adding Additional Store

Successfully signed: VLC.appx

Number of files successfully Signed: 1

Number of warnings: 0

Number of errors: 0

演習 2 のまとめ

コード署名用の証明書の作成方法を学習しました。

Appx パッケージへの署名方法を学習しました

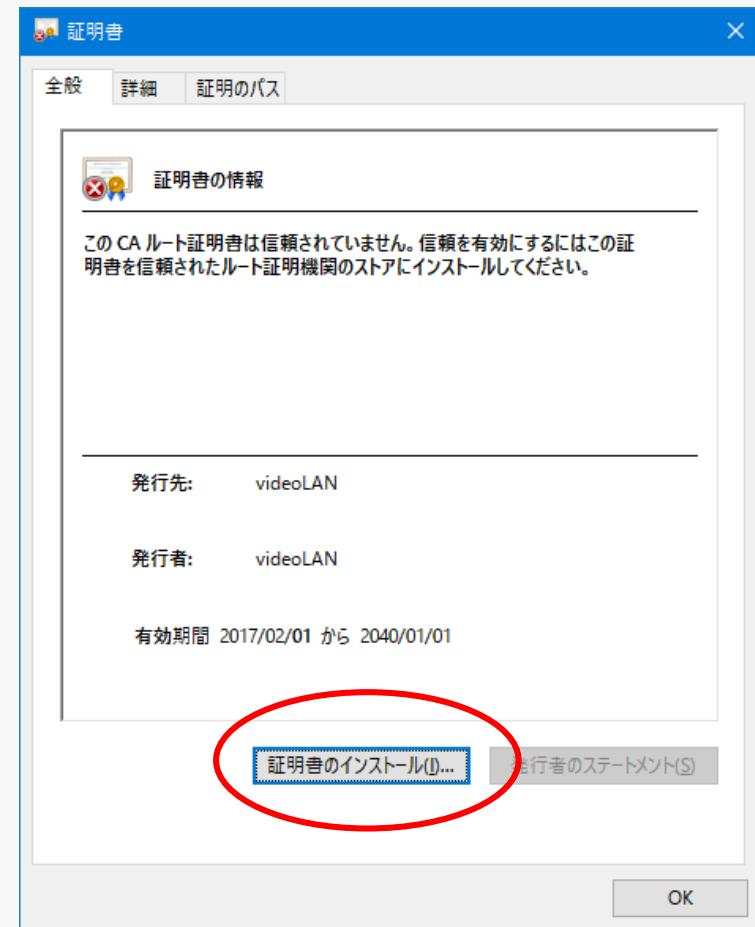
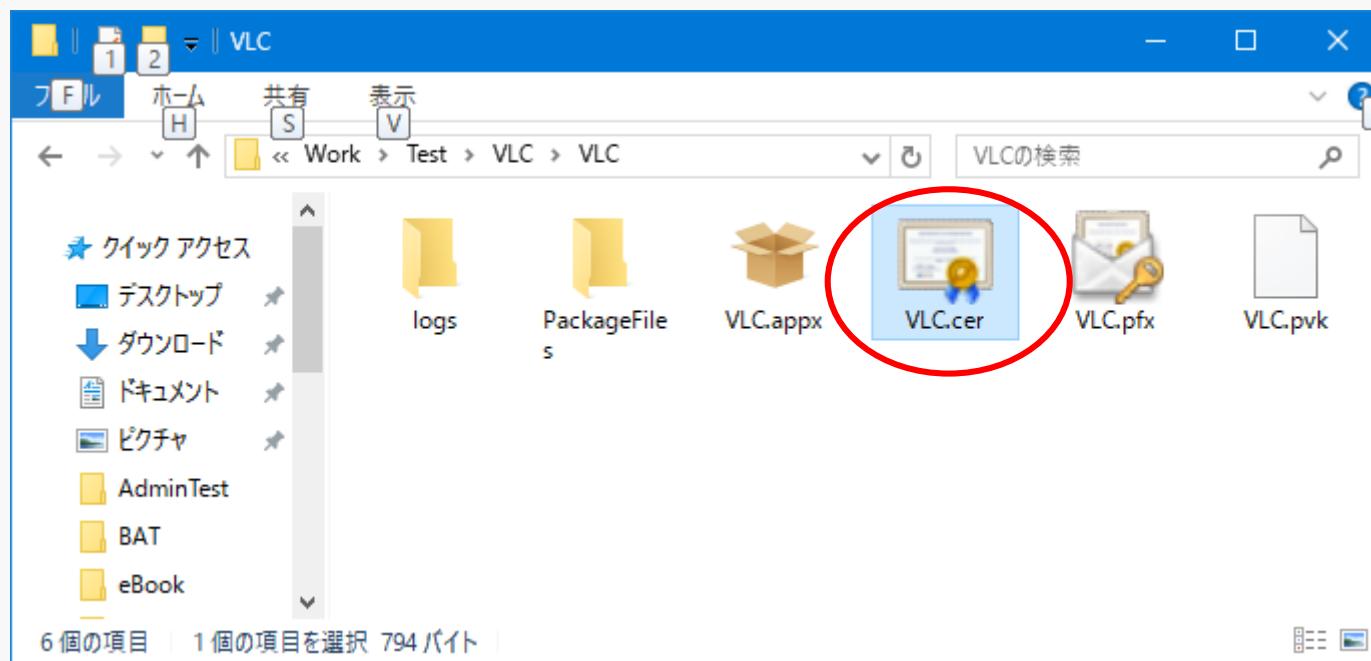
演習 3 で証明書のインストールを学習しますが、管理する必要のある証明書は、 pfx と cer の 2 種類になります

演習 3

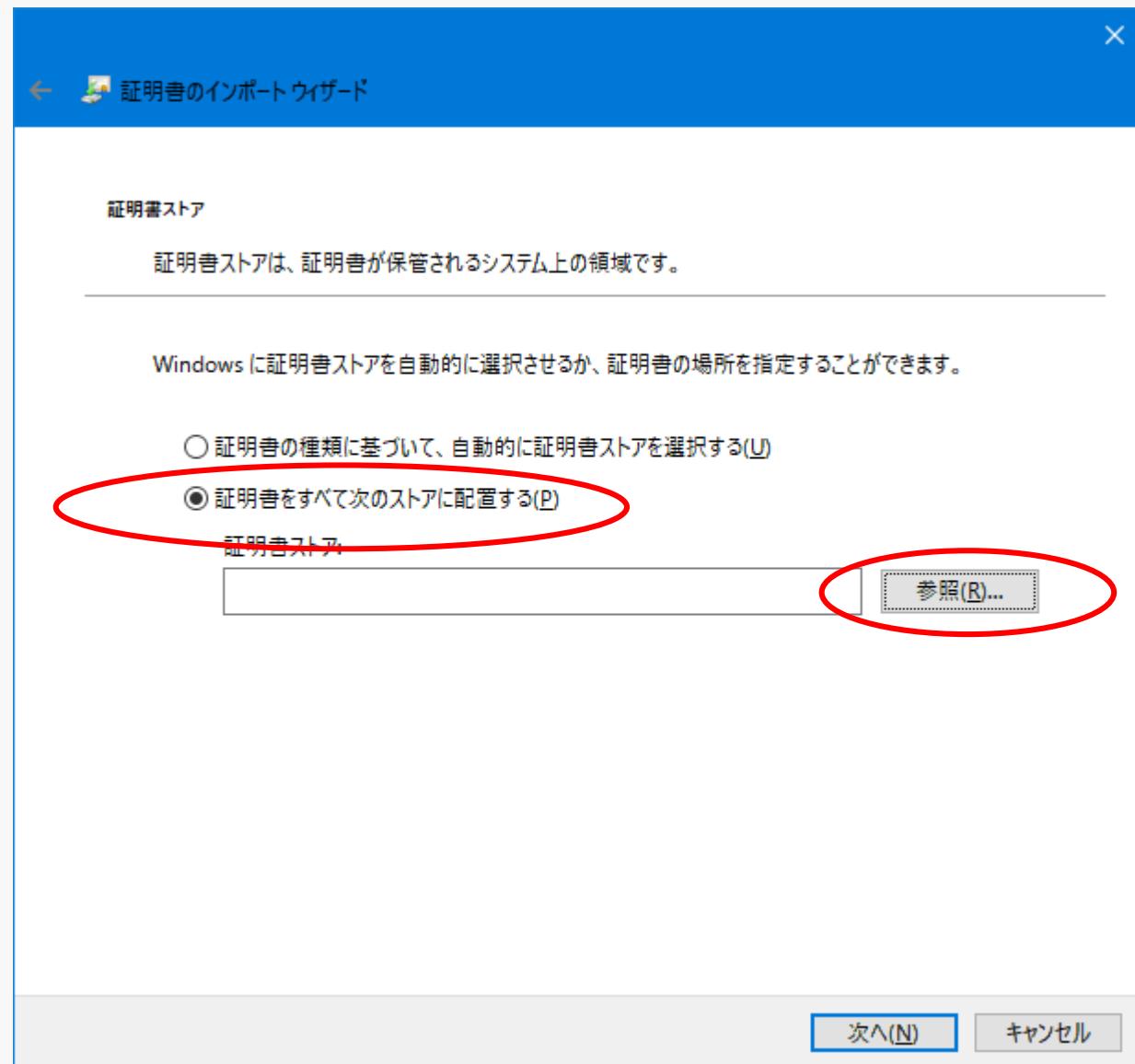
アプリのインストール

証明書をインストール 1/4

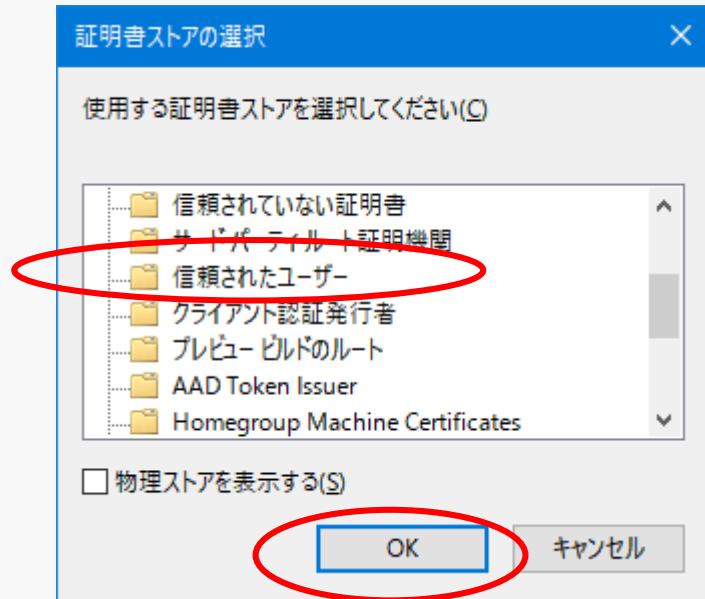
Windows エクスプローラで証明書をダブルクリックします



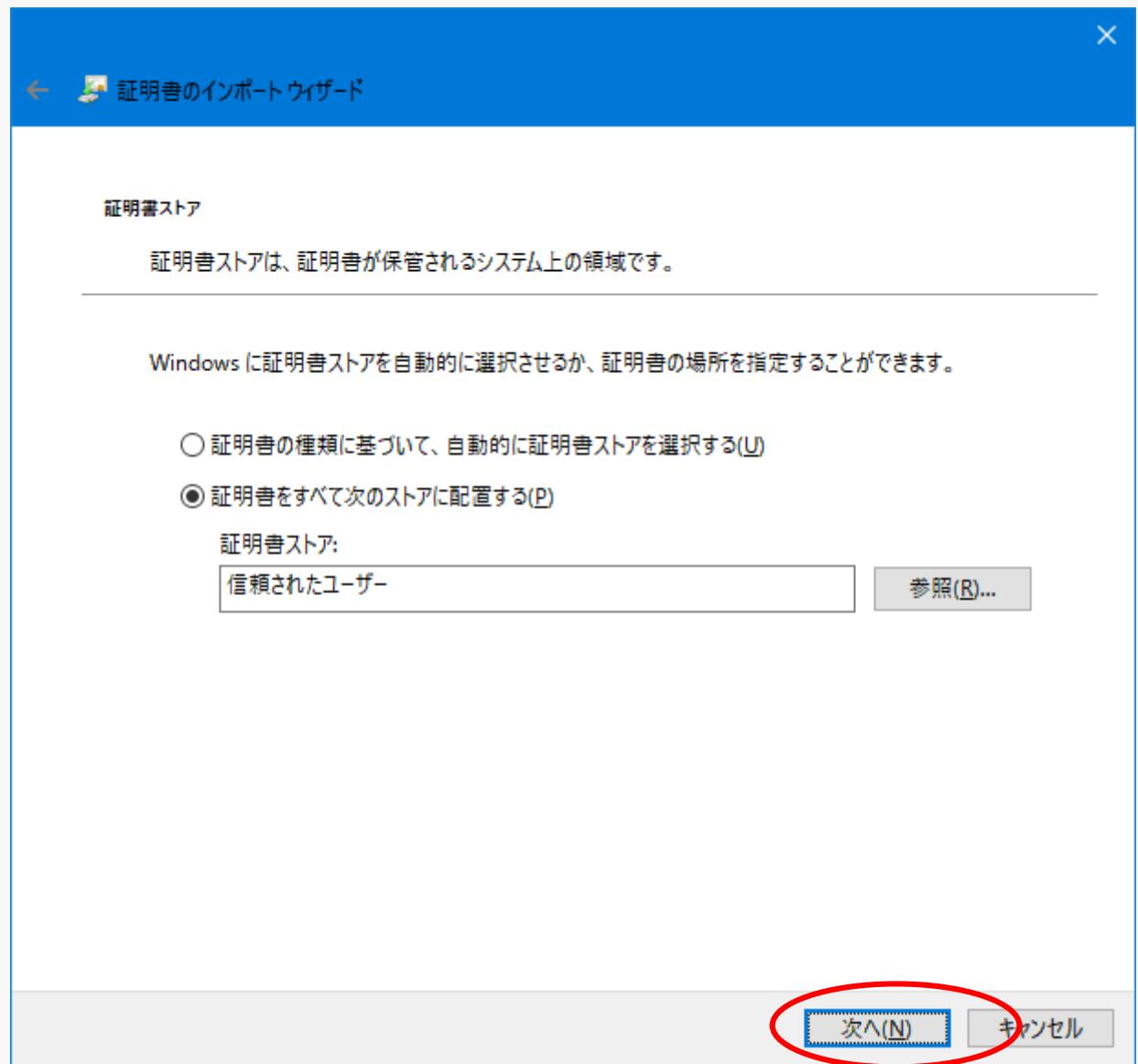
証明書をインストール 2/4



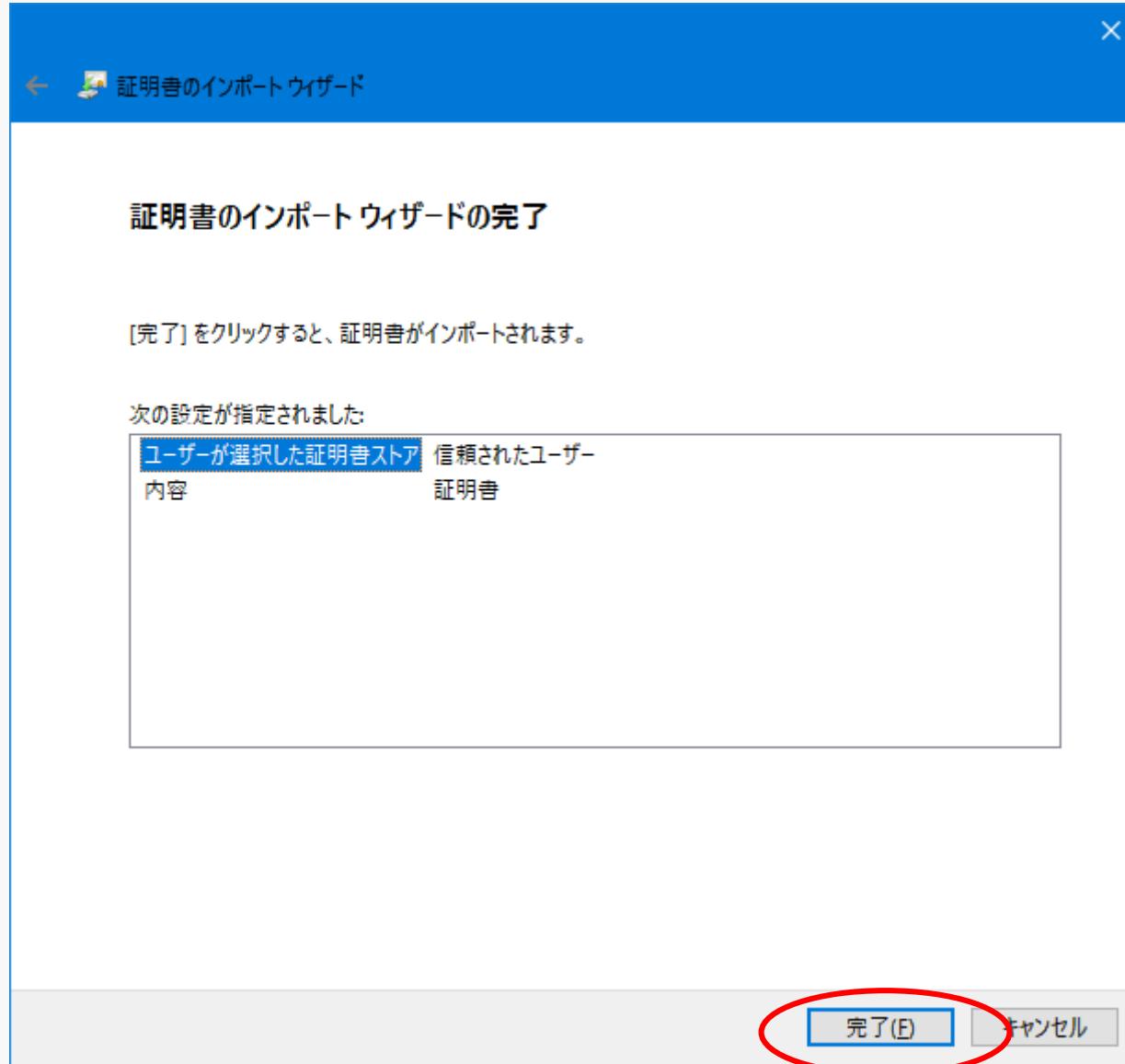
証明書をインストール 3/4



[ローカルコンピュータ]-
[信頼されたユーザー]
ストアを選択します

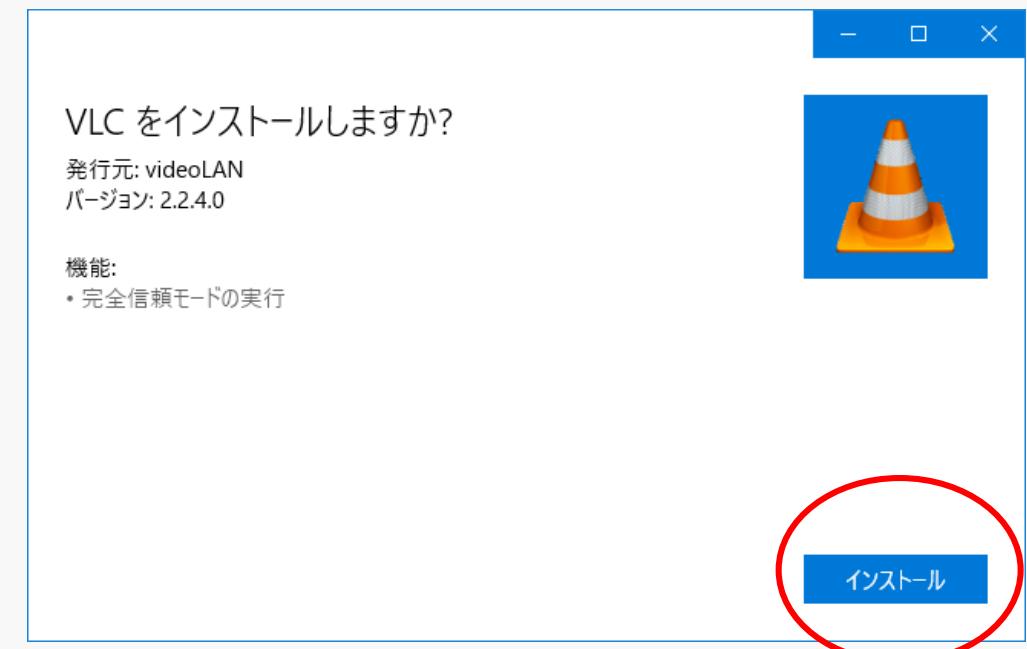
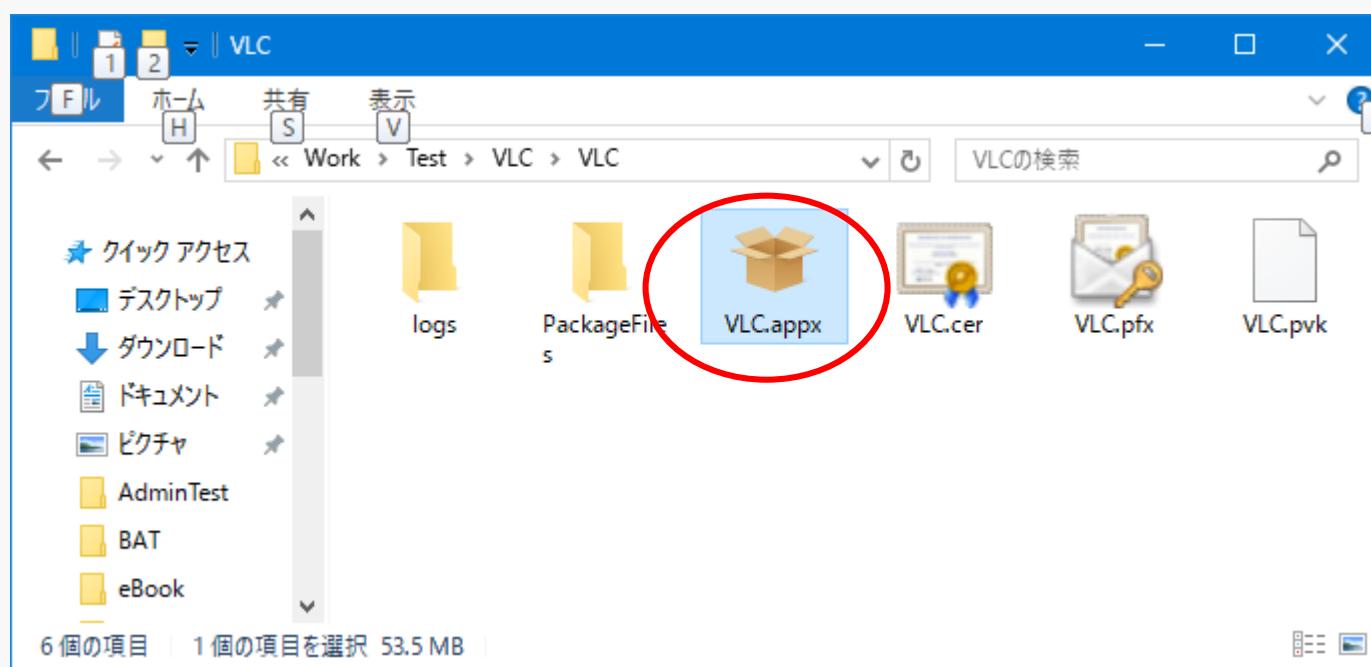


証明書をインストール 4/4



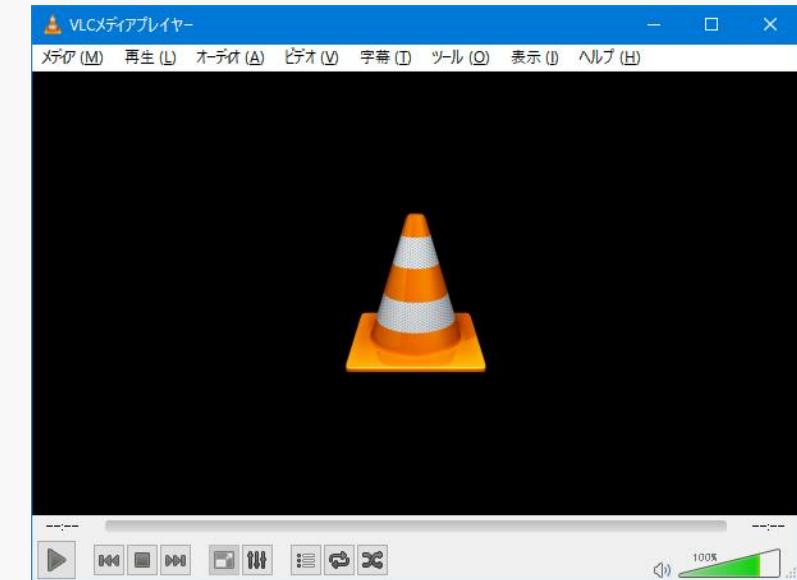
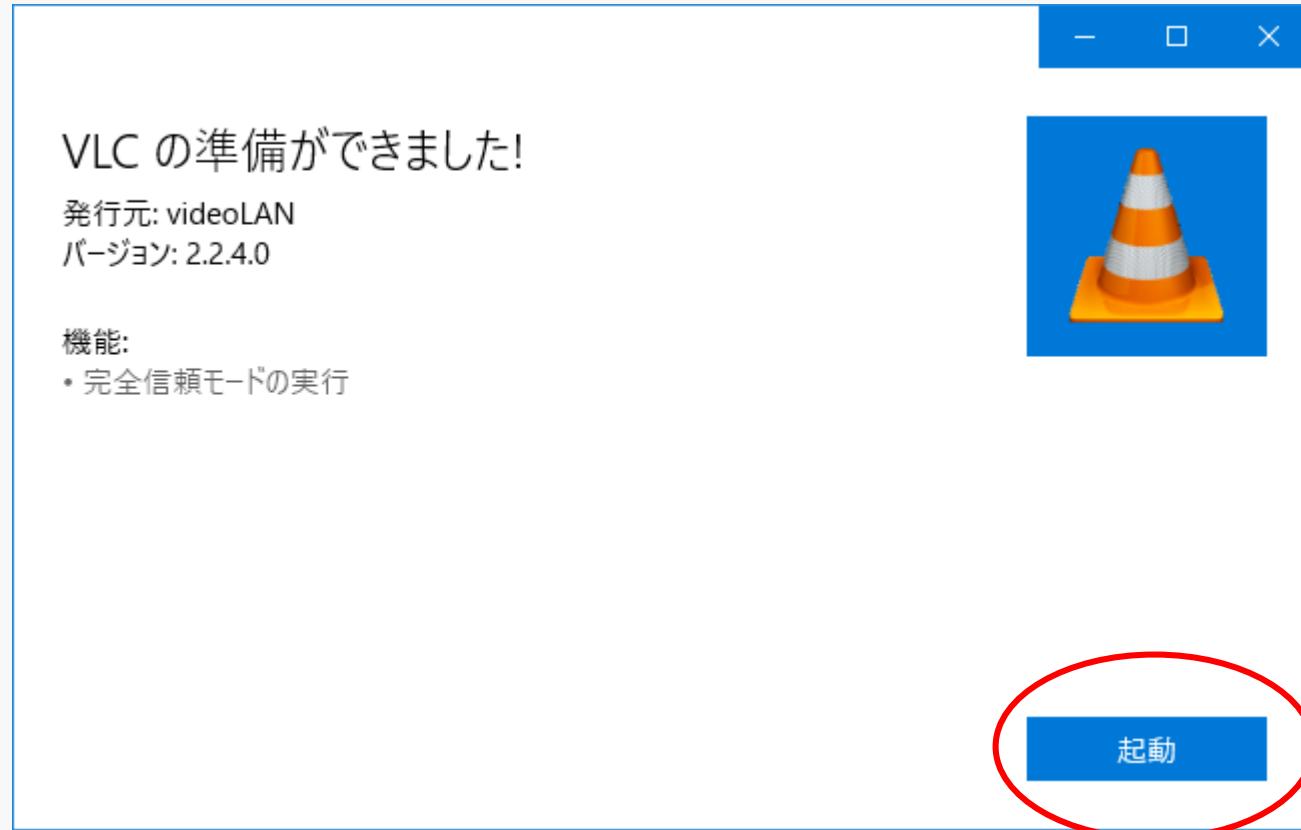
アプリのインストール

Windows エクスプローラで Appx をダブルクリックします



アプリの起動

アプリを起動して、動作を確認します



アプリのインストール（参考）

Appx のダブル クリックでインストールできない場合は、PowerShell を使用します

```
C:¥DAC¥VLC> powershell  
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All  
rights reserved.
```

```
PS C:¥DAC¥VLC> Add-AppxPackage .¥VLC.appx
```

演習 3 のまとめ

コード署名用の証明書ファイル (cer ファイル) の
インストール 方法を学習しました

署名済みの Appx パッケージのインストール方法
を学習しました

Appx パッケージは、コード署名がされていないとインス
トールができません

インストールに失敗する場合は、PowerShell を使用します

演習4 PRIリソースの作成 (オプション)

演習5と6を行うには演習4を実施しておきます
または、演習7でも構いません

PRIについて

UWP アプリ環境では、デスクトップと違って新しいリソース マネージャが提供されています

新しいリソース マネージャが取り扱うのが、
Package Resource Index (PRI) になります

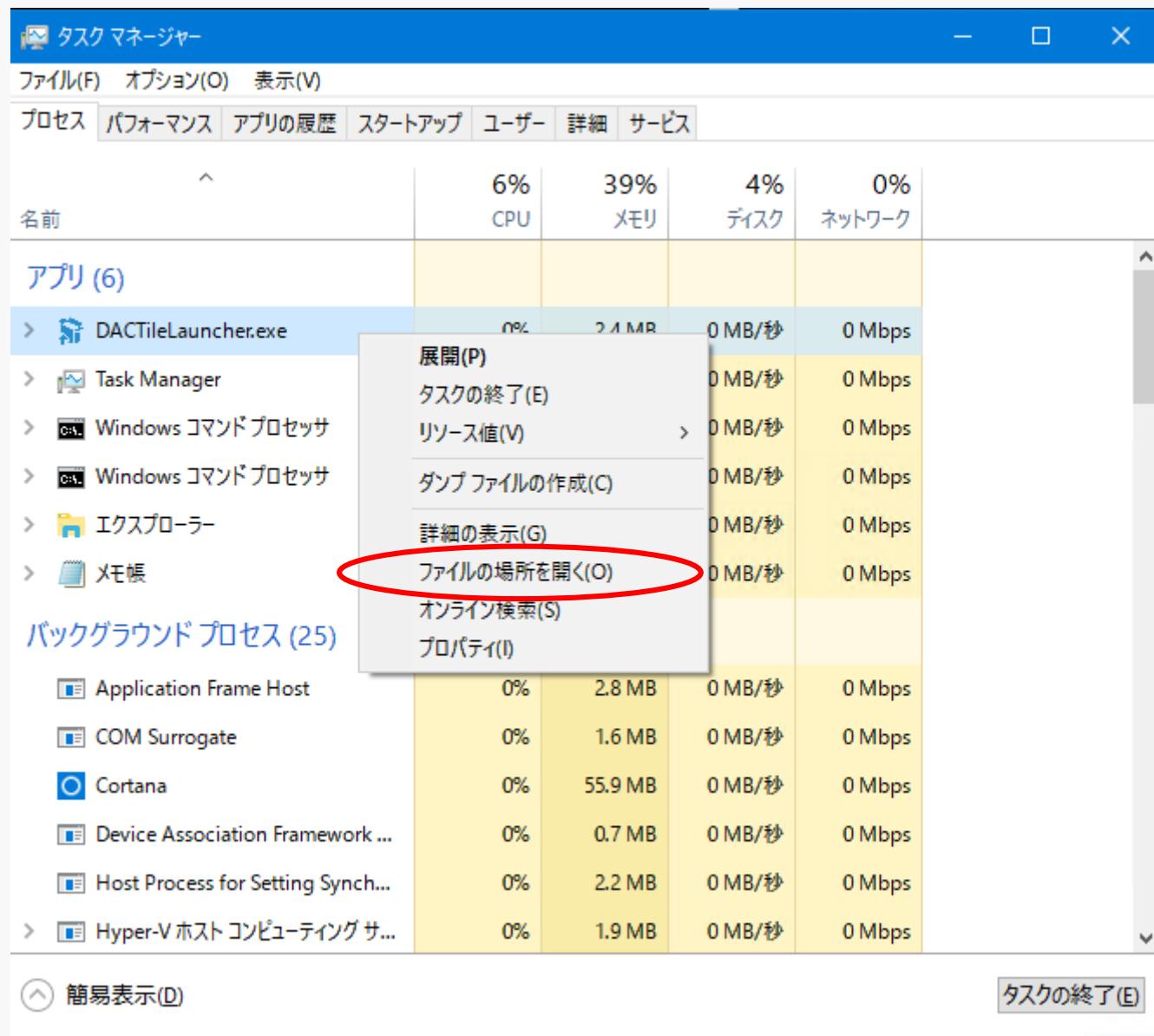
タイルアイコンなどのファイル名を変更したり、タイルアイコンを追加した場合に PRI ファイルを作り直す必要があります。

この演習は、PRI ファイルの作成方法になります

PRI 構成ファイルなどを入手します

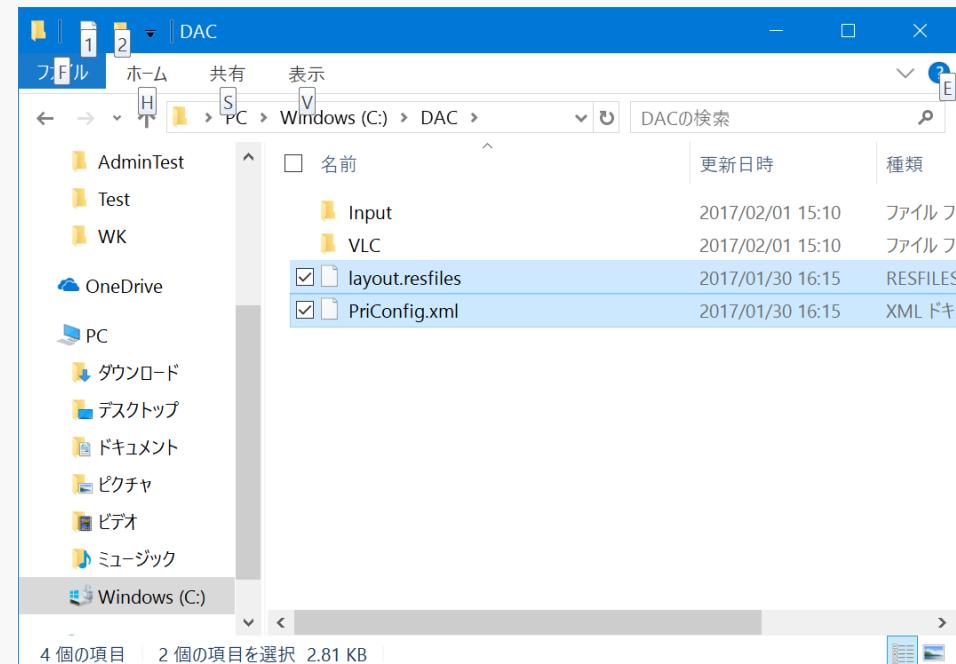
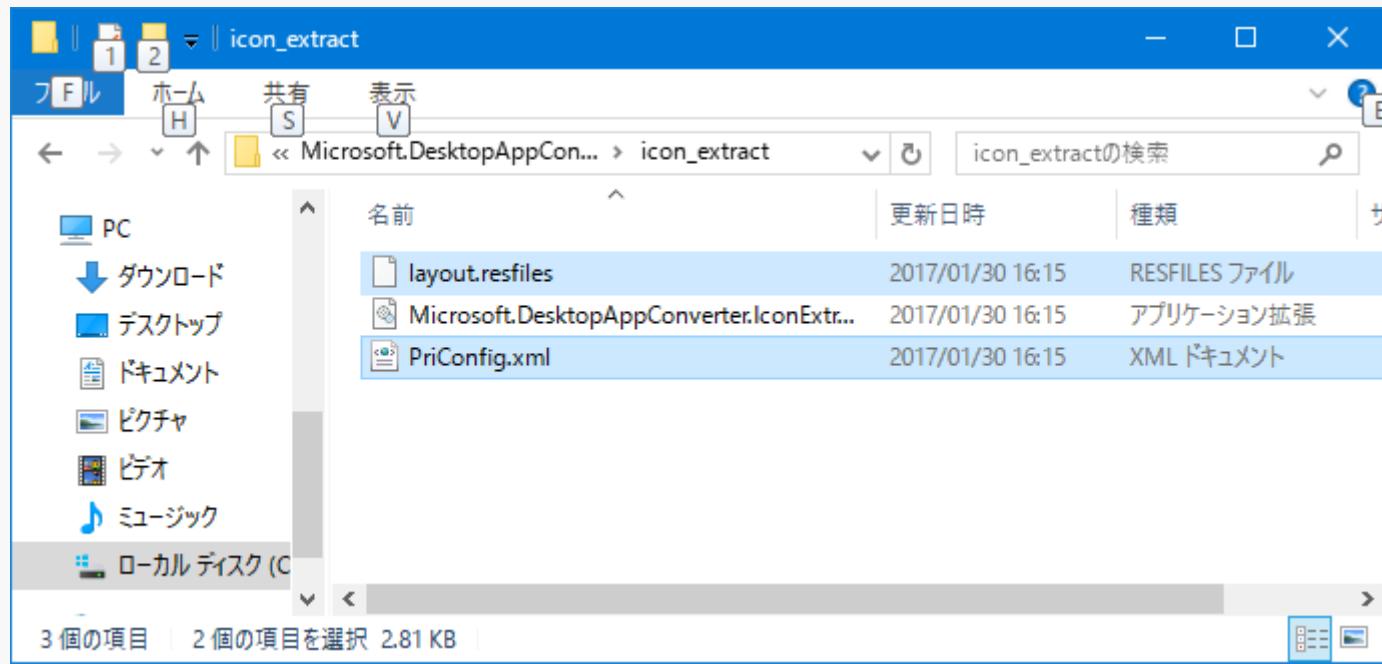
スタートメニューより、
Desktop App Conveter を
起動します。

タスクマネージャを
起動して、
「DacTileLauncher.exe」
のコンテキストメニュー
より、「ファイルの場所を
開く」を選択します。



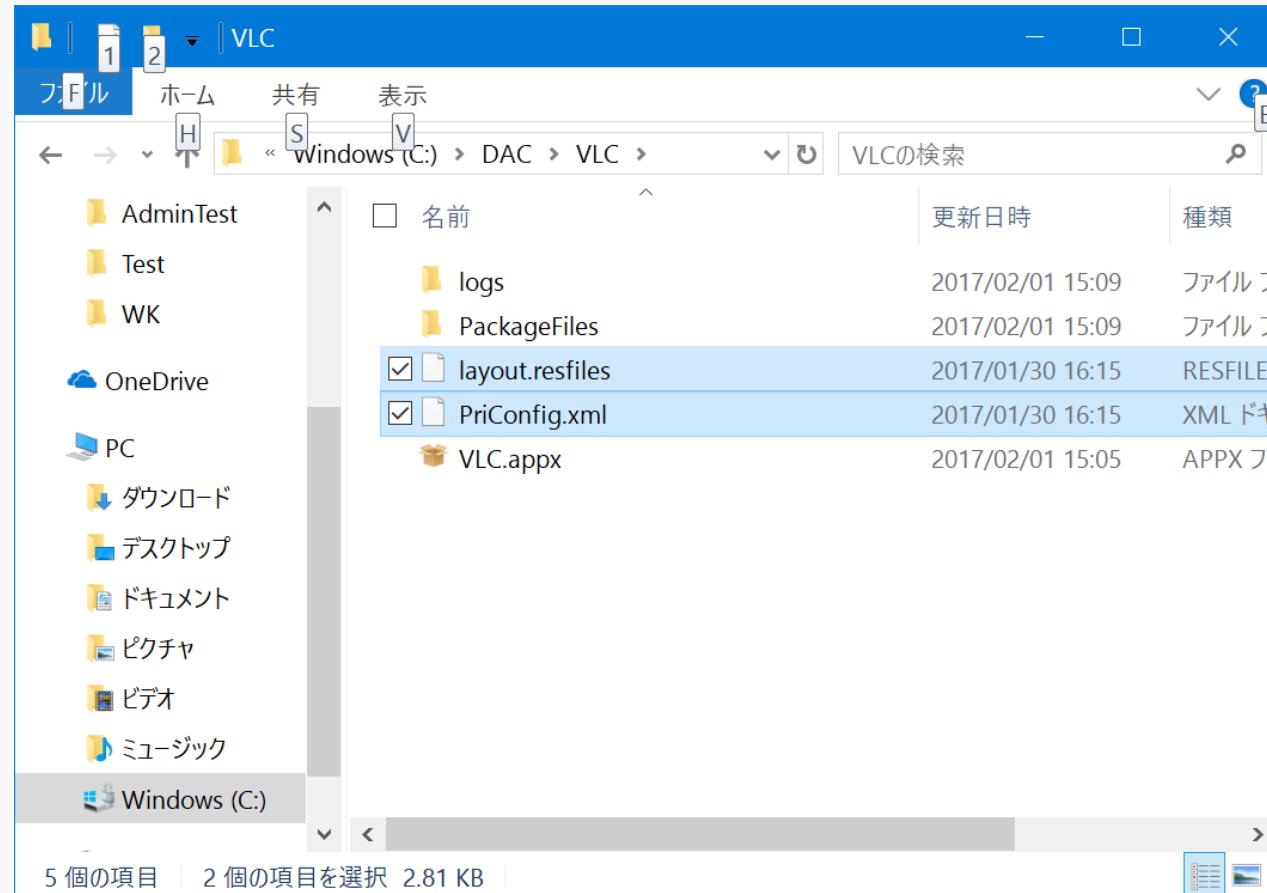
PriConfig.xml と layout.resfiles をコピー

Icon_extract フォルダを開いて、PriConfig.xml と layout.resfiles を C:\DAC フォルダへコピーします



2つのファイルを VLC へコピーします

PriConfig.xml と layout.resfiles を C:¥DAC¥VLC
へコピーします。



PriConfig.xml を編集します

C:\DAC\VLV\PriConfig.xml の layout.resfiles 定義を変更します (8 行目)。

変更前

```
<index root="\\" startIndexAt="layout.resfiles">
```

変更後

```
<index root="\\" startIndexAt="..\layout.resfiles">
```

必要に応じて layout.resfiles を編集します

layout.resfiles は、VLC¥PackageFiles¥Assets
フォルダ内のファイル名が記録されています。
Assets フォルダ内のファイル名を変更したり、
ファイルを追加した場合にのみ layout.resfiles を
編集します。

Assets フォルダ内の変更を行わない場合は、編集
は必要ありません。

既存の PRI ファイルを削除します

VLC¥PackageFiles フォルダに
ある PRI ファイルを削除しま
す。

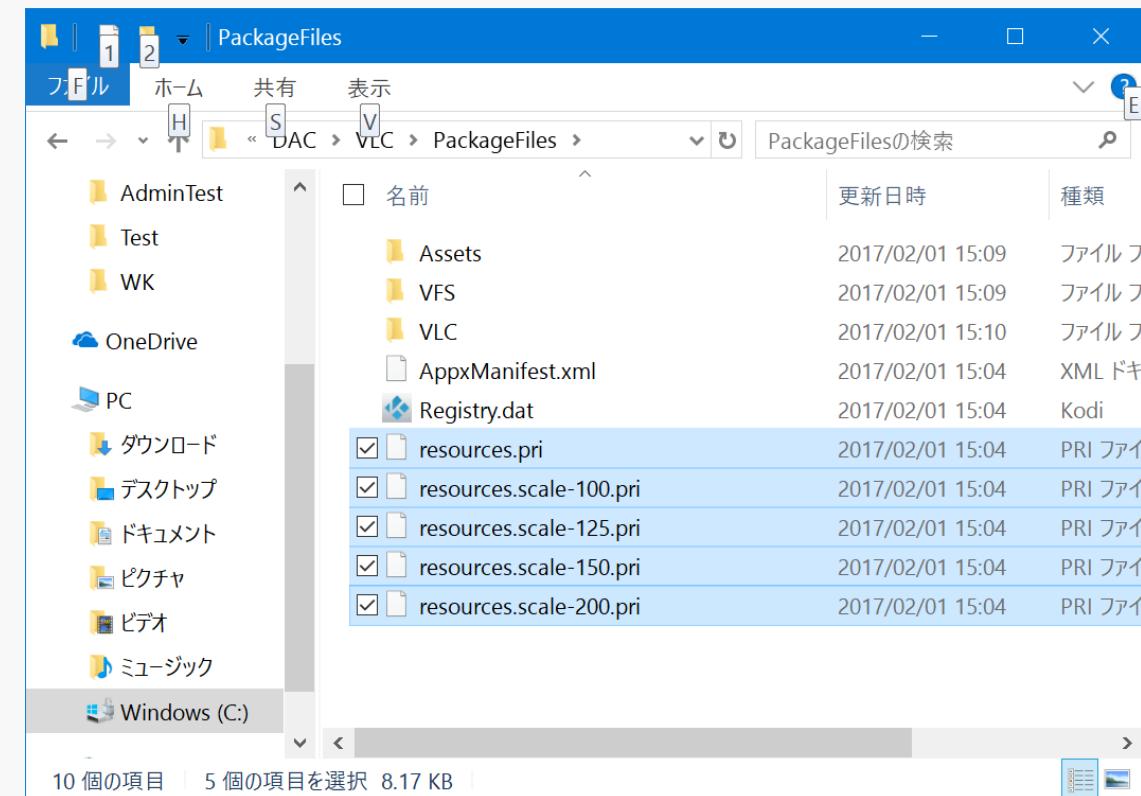
resources.pri

resources.scale-100.pri

resources.scale-125.pri

resources.scale-150.pri

resources.scale-200.pri



PRI ファイルを作成します

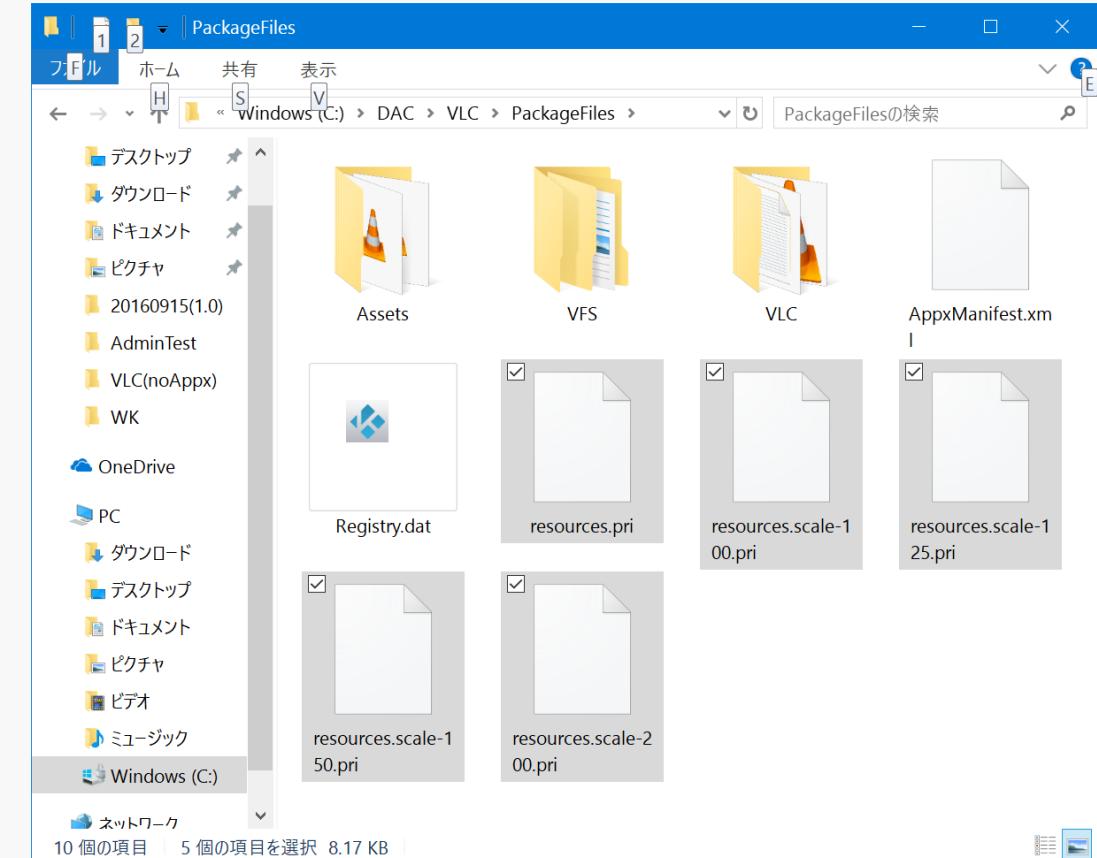
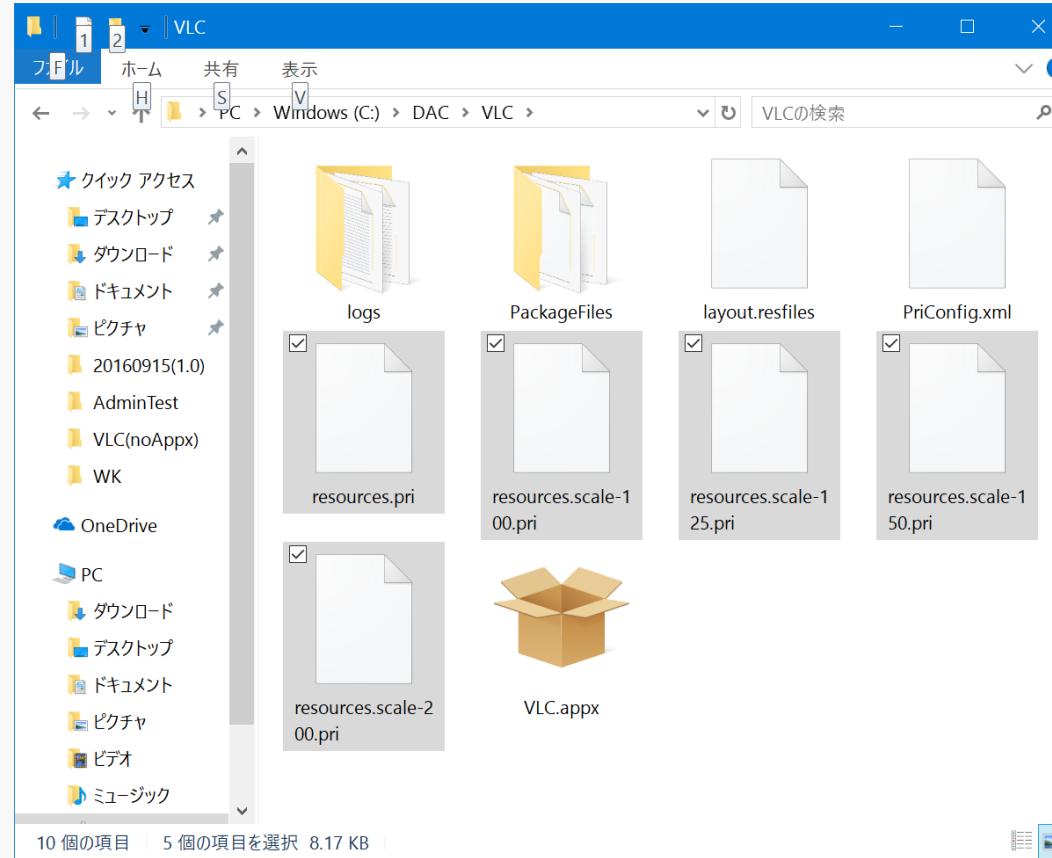
```
C:¥DAC¥VLC> SET PATH="C:¥Program Files  
(x86)¥Windows Kits¥10¥bin¥x86";%PATH%  
PS C:¥DAC¥VLC> makepri.exe new /pr ".¥PackageFiles"  
/cf ".¥PriConfig.xml" /o
```

```
.....  
Resource File: resources.scale-200.pri  
Version: 1.0  
Resource Candidates: 6  
Scale Qualifiers: 200  
Successfully Completed
```

Successfully Completed と表示されれば完了です

作成した PRI ファイルをコピーします

作成したPRI ファイルを VLC¥PackageFiles フォルダーへコピーします。



演習 4 のまとめ

PRI ファイルの作成方法を学習しました。

PRI ファイルは、UWP アプリ用のリソース マネージャが使用するリソース ファイルです。

Assets フォルダ内のファイル名の変更やファイルの追加があった場合に作成します。

既存の PRI ファイルは削除しておきます。この理由は、makepri ユーティリティの既定の動作がマージになっているためです。

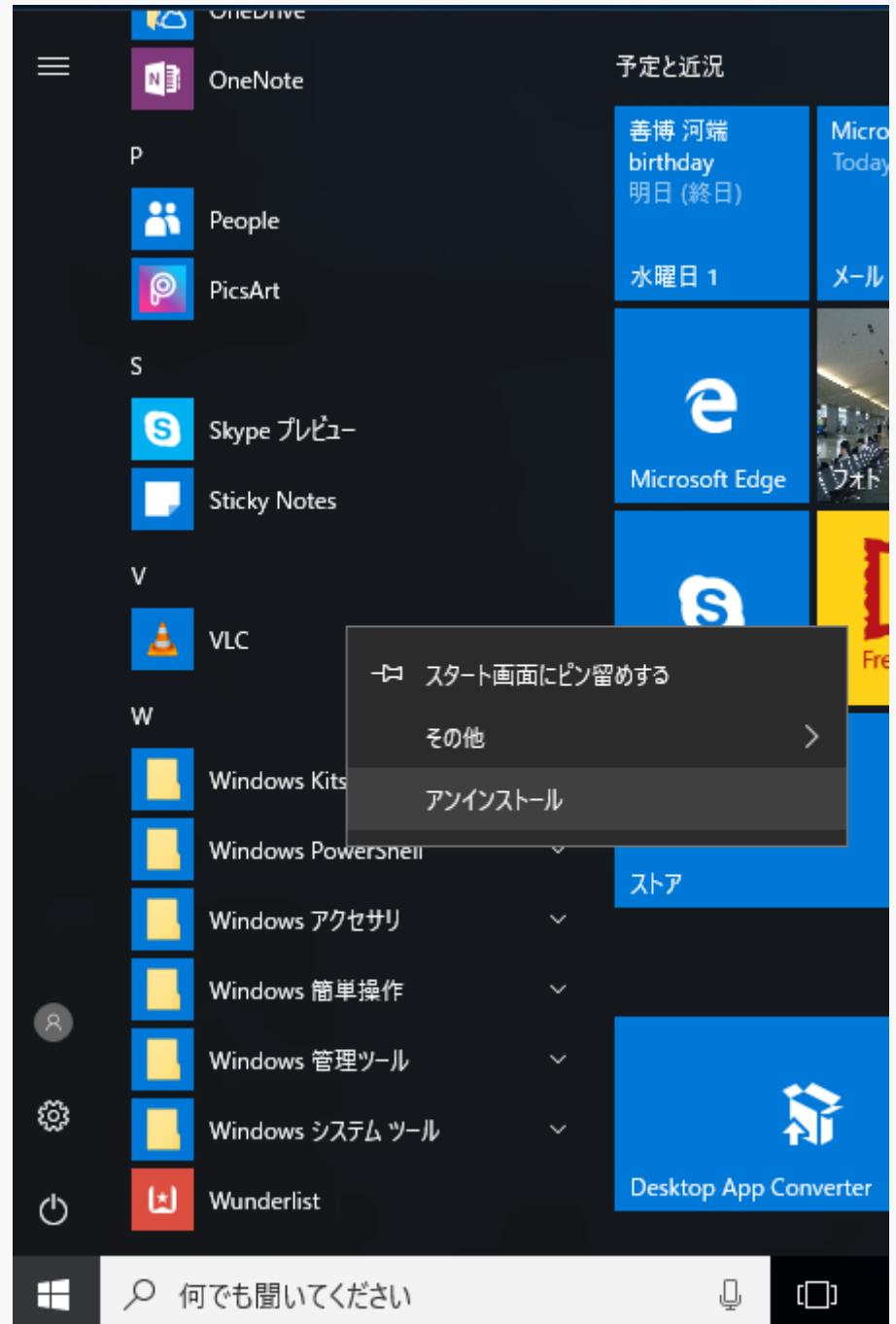
作成した PRI ファイルを、PackageFiles フォルダへコピーします。

演習 5

開発環境へのアプリのインストール

アプリのアンインストール

スタートメニューから、VLC を選択して、コンテキストメニューからアンインストールを選択します



開発用にアプリをインストールします

```
C:¥DAC¥VLC> powershell  
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All  
rights reserved.
```

```
PS C:¥DAC¥VLC> Add-AppxPackage -Register  
.¥PackageFiles¥AppxManifest.xml
```

変換済みのアプリのテストを行います

実行ファイルのみを変更した場合は、 PackageFiles フォルダ内の実行ファイルを入れ替えて、テストします

演習 5 のまとめ

アプリのアンインストール方法を学習しました
デバッグ用のアプリのインストール方法を学習しました

アンインストール後に、実行ファイルを入れ替えます。
デバッグ用にアプリをインストールして、テストします。
レジストリ (HKLM ハイブ) など変更していない場合は、
DAC を使って変換し直す必要はありません。

演習 6

SDK ツールを使った Appx パッケージの作成

Appx パッケージを作成します

makeappx ユーティリティで Appx パッケージを作成します。

Succeeded と表示されれば Appx パッケージが作成されます。

```
C:\¥DAC¥VLC> SET PATH="C:\Program Files  
(x86)\Windows Kits\10\bin\x86";%PATH%  
C:\¥DAC¥VLC> makeappx pack /d .\PackageFiles /l /p  
vlc1.appx  
.....  
.scale-200.pri".  
Package creation succeeded.
```

Appx パッケージに署名します

演習 2 と 演習 3 を参照します。

この演習では、vlc1.appx という appx パッケージを作成しました。

演習 6 のまとめ

Appx パッケージを手動で作成する方法を学習しました。

AppxManifest.xml など、アプリを構成するファイルを変更した場合に Appx パッケージを作成し直す必要があります。

PRI リソースを使用する場合は、makeappx ユーティリティに /l (エル) オプションを使用します。

PRI リソースを使用しない場合は、 /l オプションは不要です。

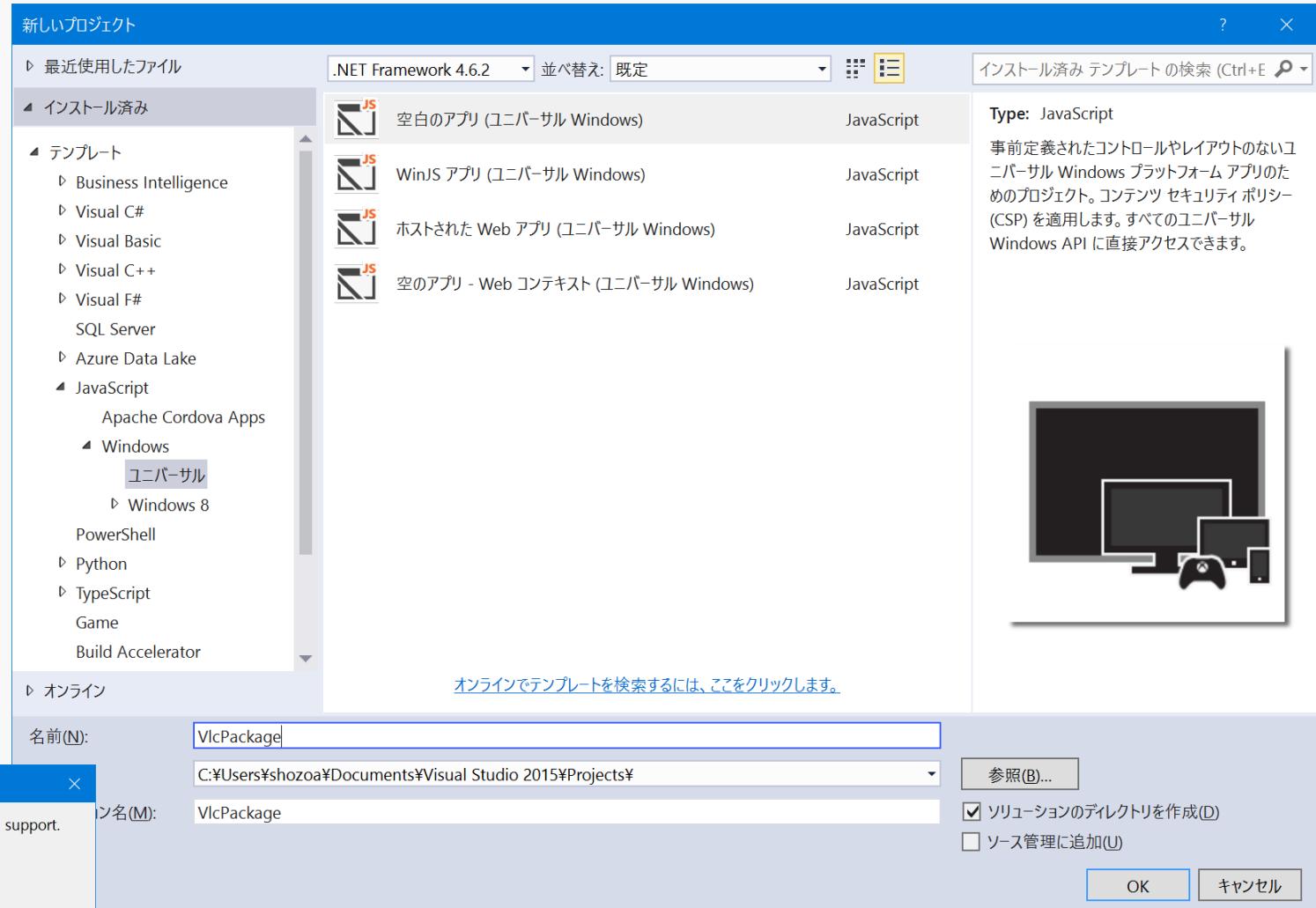
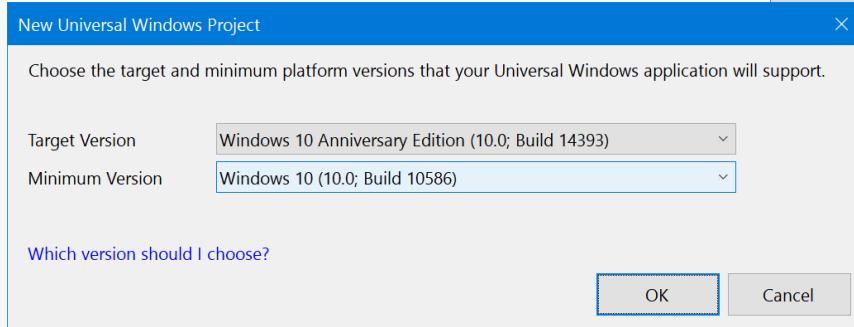
演習 7（オプション）

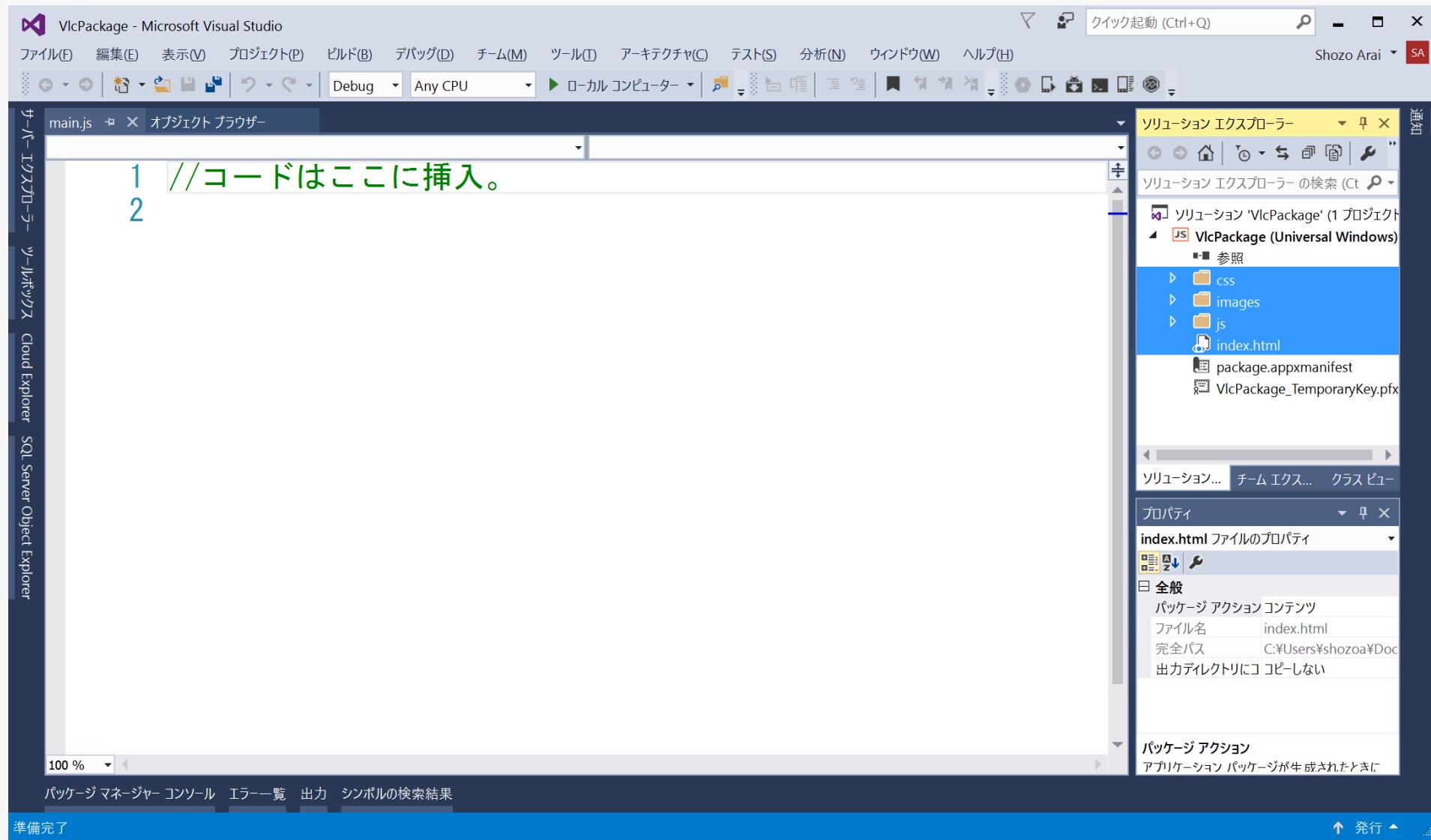
Visual Studio を使った Appx パッケージの作成

新しいプロジェクトを作成します

[ファイル]-[新規作成]-
[プロジェクト]を選択し
ます

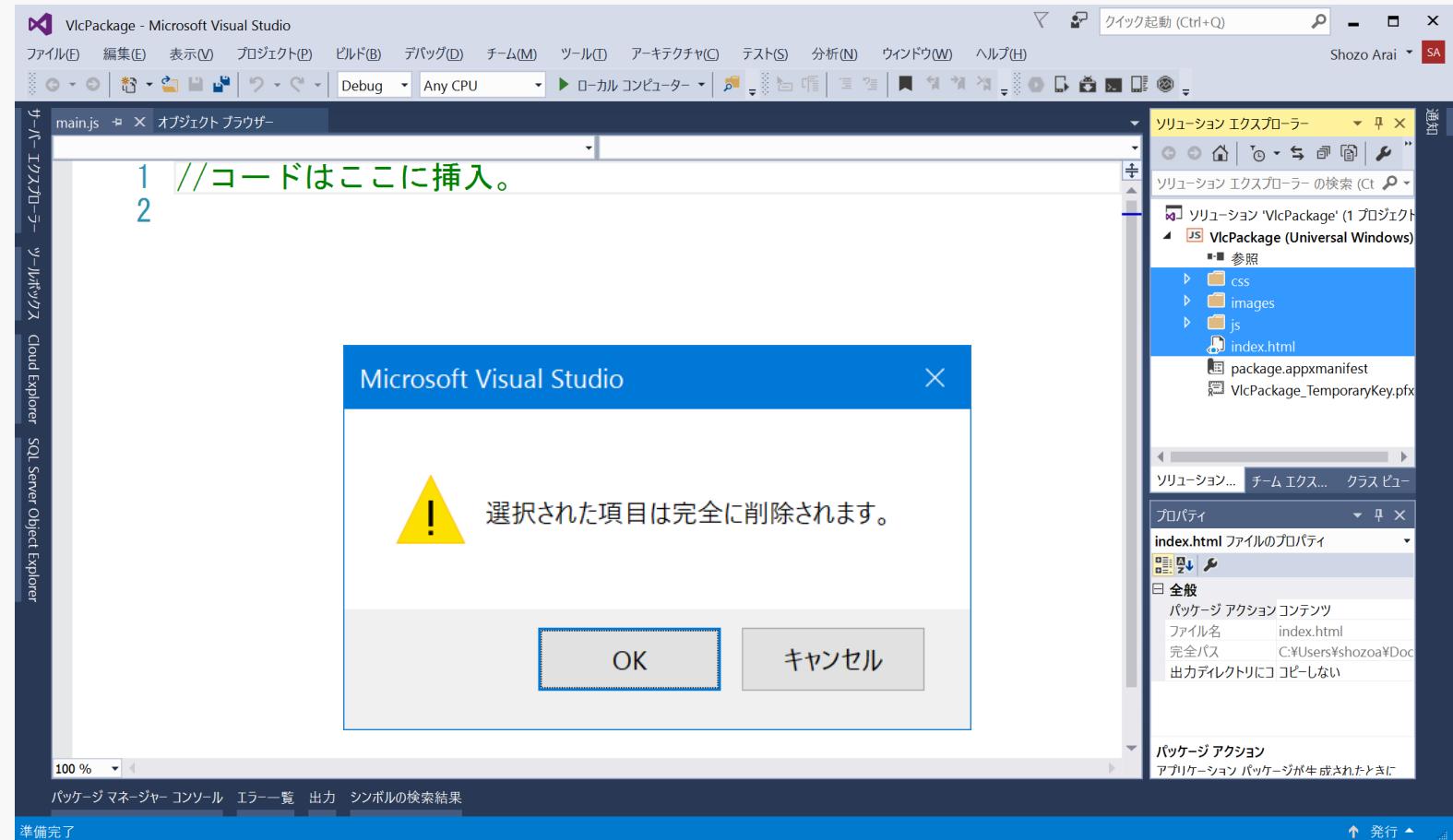
[JavaScript]-
[Windows]-[ユニバーサル]-[空のアプリ
(Windows ユニバーサル)]でプロジェクトを作
成します





プロジェクトから既存の項目を削除します

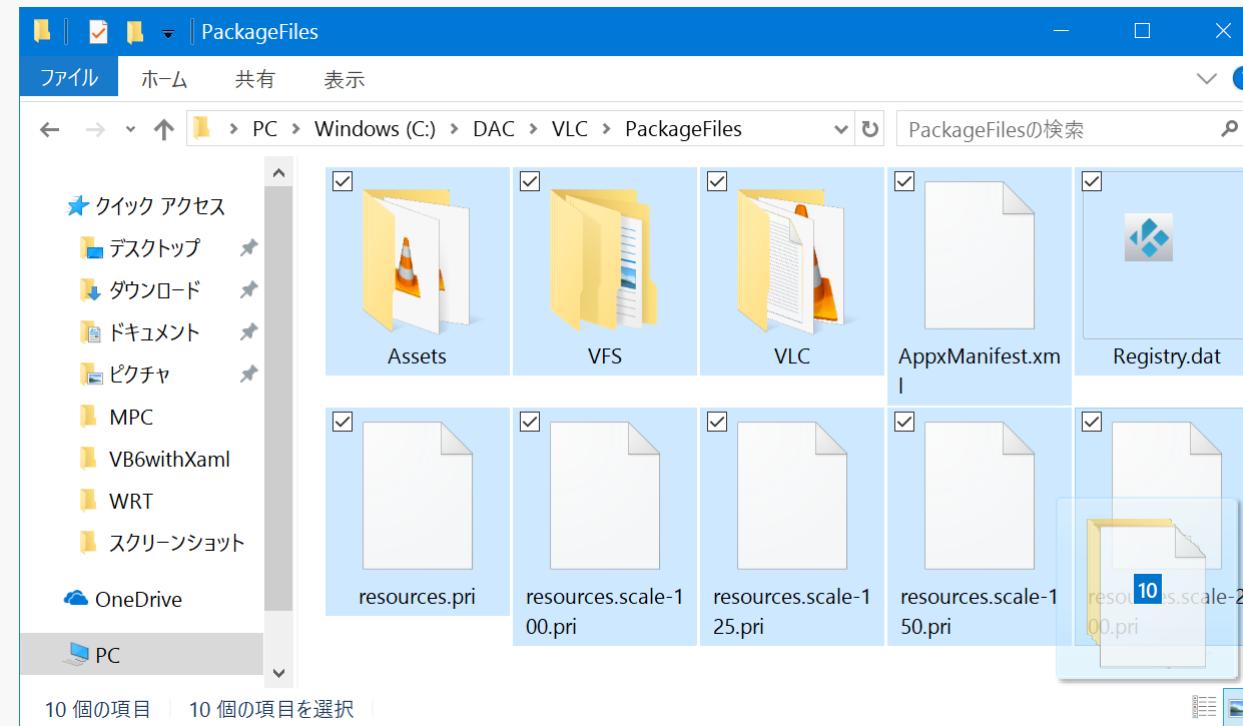
css フォルダ、
js フォルダ、
images フォルダ、
index.html を削
除します。
[コンテキストメ
ニュー]-[削除]
を選択します



Appx レイアウトをプロジェクトへ追加

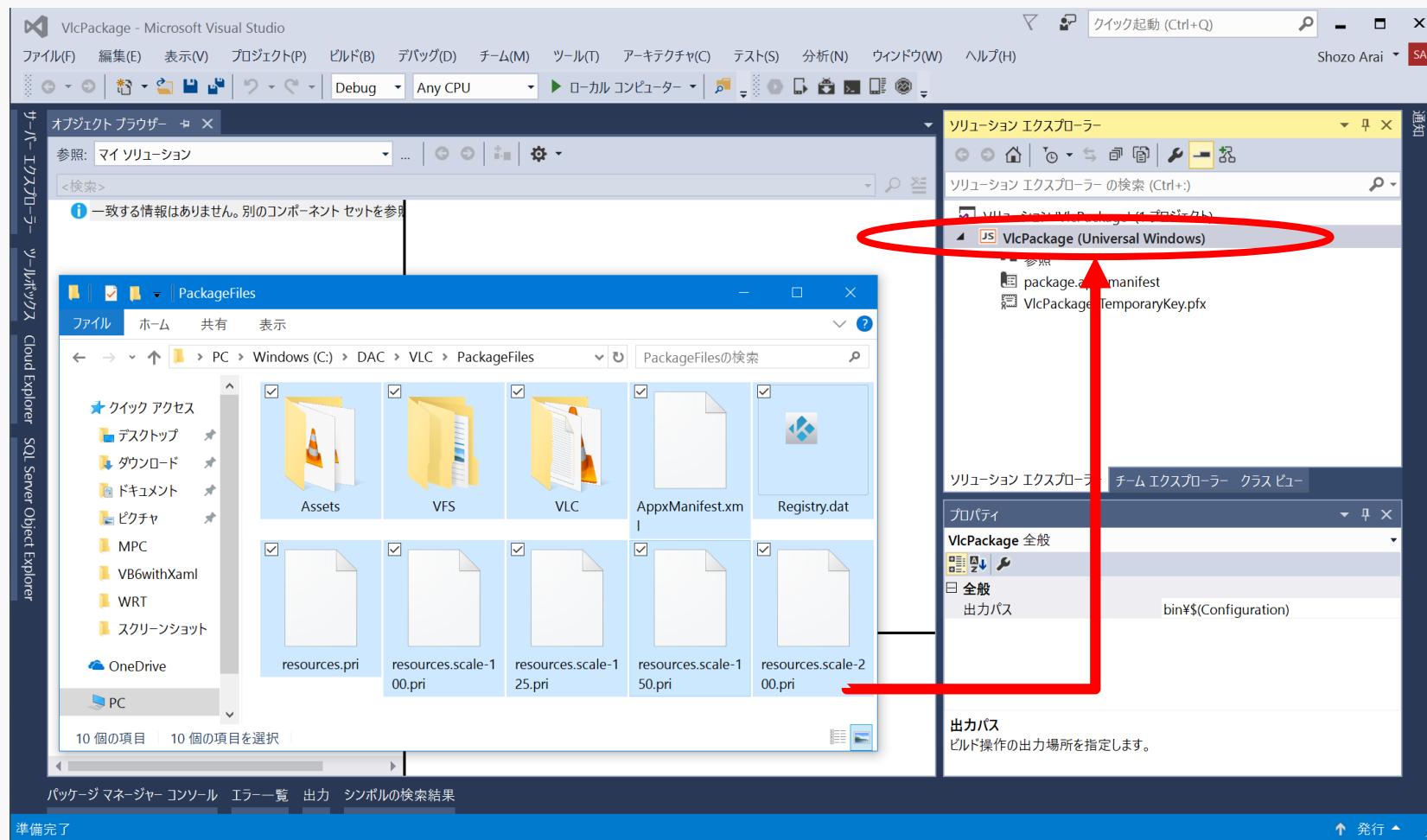
演習 1 で作成した、C:\DAC\VLC\PackageFiles フォルダの内容をプロジェクトへ追加します

Windows エクスプローラで C:\DAC\VLC\PackageFiles フォルダの内容を Visual Studio へドラッグ & ドロップします



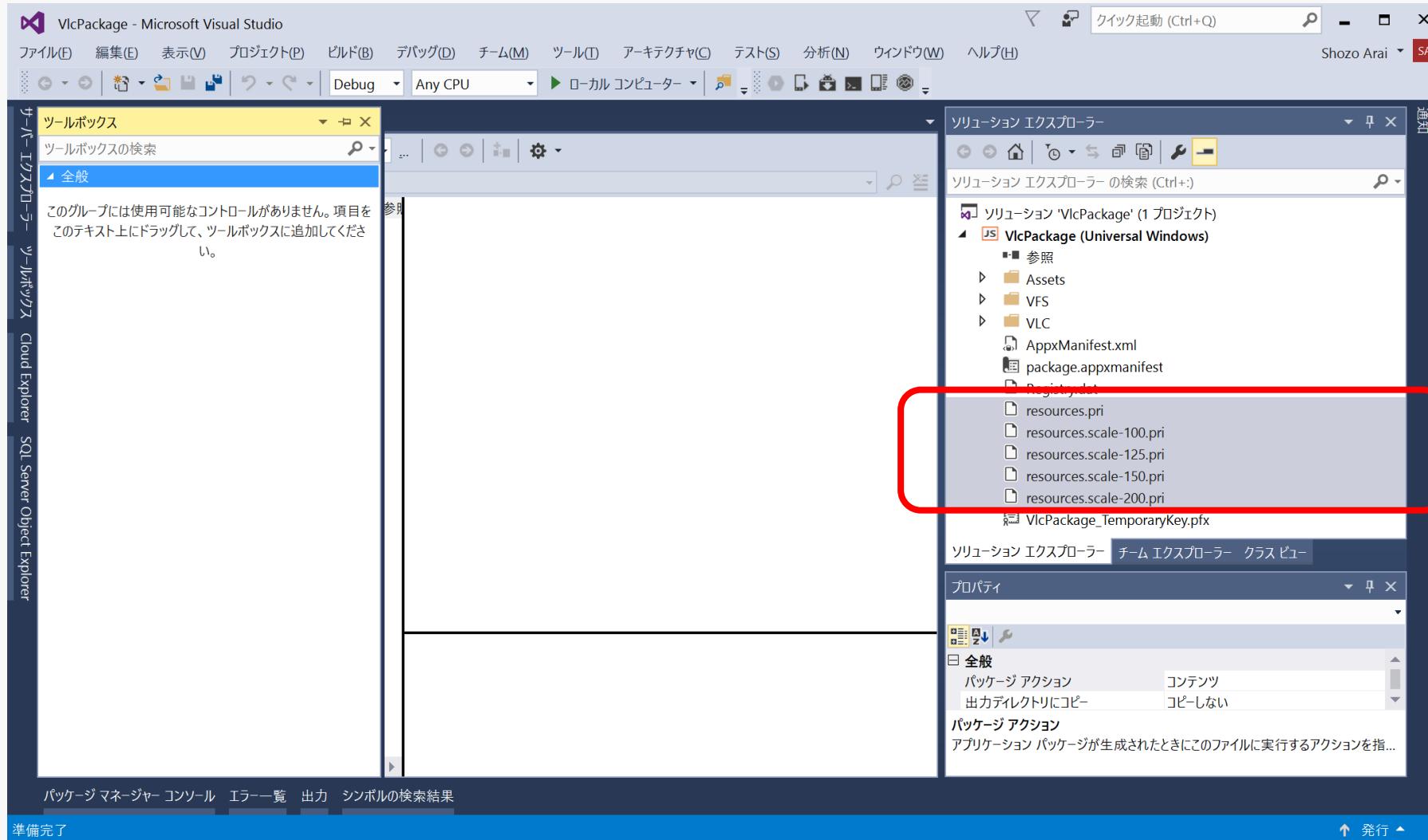
Appx レイアウトをプロジェクトへ追加

ソリューションエクスプローラーのプロジェクトへ
ドロップします



不要なものを削除します

pri ファイルが含まれている場合は、削除します



Package.appxmanifest を編集します

AppxManifest.xml を package.appxmanifest へ
マージします

Package 要素へ名前空間を追加します

Identiny 要素を修正します

Properties 要素から Capabilities 要素を置き換えます

Package 要素へ名前空間を追加

2 行目から 6 行目の Package 要素を変更します
名前空間を 4 つ追加します

変更前

```
<Package
```

```
    xmlns="http://schemas.microsoft.com/appx/manifest/foundation/windows10"
    xmlns:mp="http://schemas.microsoft.com/appx/2014/phone/manifest"
    xmlns:uap="http://schemas.microsoft.com/appx/manifest/uap/windows10"
    IgnorableNamespaces="uap mp">
```

Package 要素へ名前空間を追加

2 行目から 6 行目の Package 要素を変更します

変更後

```
<Package  
    xmlns="http://schemas.microsoft.com/appx/manifest/foundation/windows10"  
    xmlns:mp="http://schemas.microsoft.com/appx/2014/phone/manifest"  
    xmlns:uap="http://schemas.microsoft.com/appx/manifest/uap/windows10"  
    xmlns:uap2="http://schemas.microsoft.com/appx/manifest/uap/windows10/2"  
    xmlns:uap3="http://schemas.microsoft.com/appx/manifest/uap/windows10/3"  
  
xmlns:rescap="http://schemas.microsoft.com/appx/manifest/foundation/windows10/restrict  
edcapabilities"  
    xmlns:desktop="http://schemas.microsoft.com/appx/manifest/desktop/windows10"  
    IgnorableNamespaces="uap mp uap2 uap3 rescap desktop">
```

Identity 要素を変更します

8 行目から 11 行目の Identity 要素を変更します
AppxManifest.xml の内容を反映させます

変更前

```
<Identity  
    Name="57776f58-6df0-43d0-b2f0-94d688609340"  
    Version="1.0.0.0"  
    Publisher="CN=shozoa" />
```

Identity 要素を変更します

8 行目から 11 行目の Identity 要素を変更します

AppxManifest.xml の内容を反映させます

Name 、 ProcessorArchitecture、 Version

Publisher は変更しないで構いません

変更後

```
<Identity  
    Name="VLC"  
ProcessorArchitecture="x86"  
    Version="2.2.4.0"  
    Publisher="CN=shozoa" />
```

Properties 要素以降を置き換えます

15 行目から 50 行目の Properties 要素から Capabilities 要素を置き換えます

変更前

```
<Properties>
  <DisplayName>VlcPackage</DisplayName>
  <PublisherDisplayName>shozoa</PublisherDisplayName>
  <Logo>images\storelogo.png</Logo>
</Properties>
...
<Capability Name="internetClient" />
</Capabilities>
```

Properties 要素以降を置き換えます

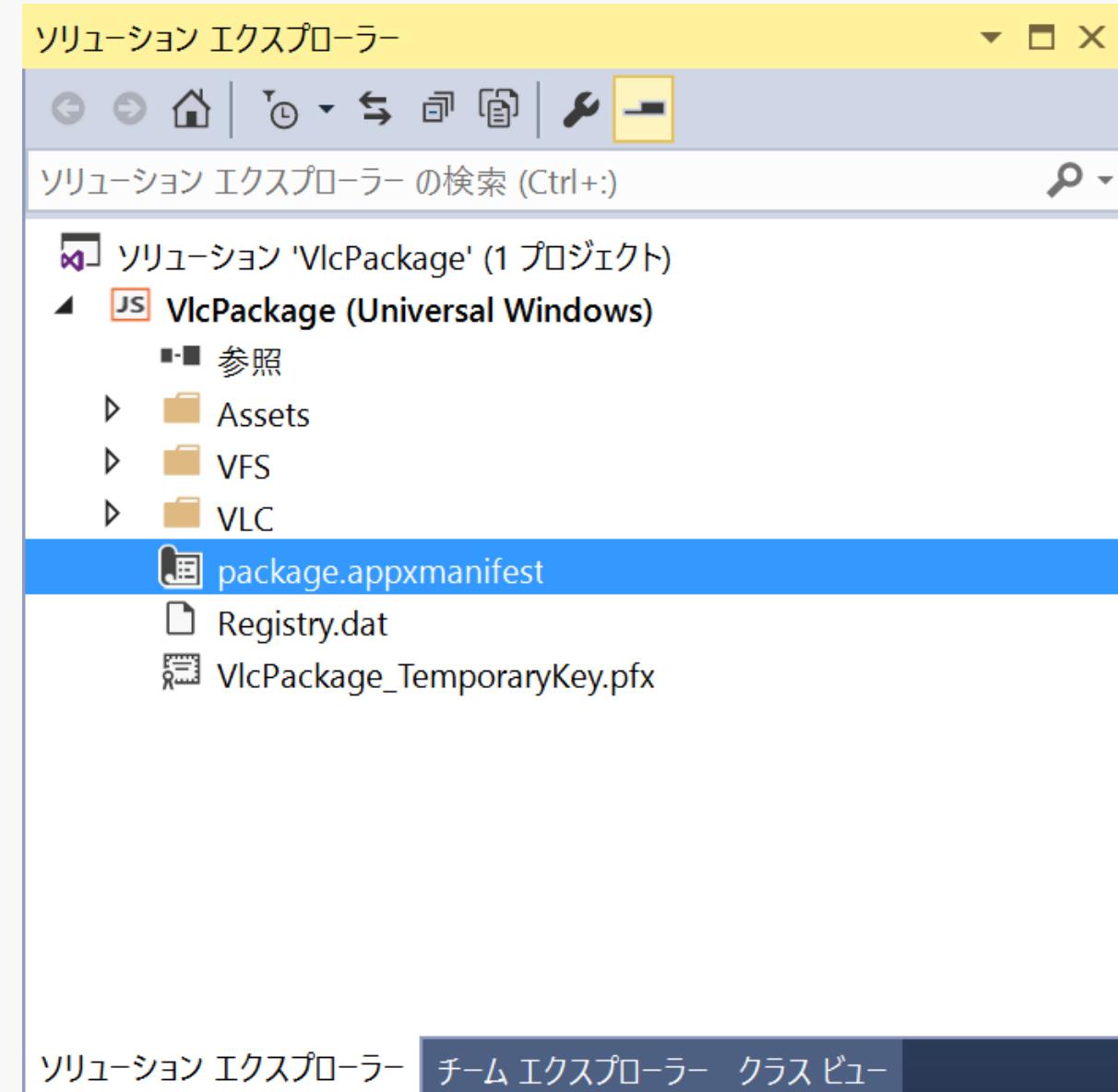
15 行目から 50 行目の Properties 要素から Capabilities 要素を置き換えます

変更後

```
<Properties>
  <DisplayName>VLC</DisplayName>
  <PublisherDisplayName>videoLAN</PublisherDisplayName>
  <Logo>Assets¥AppStoreLogo.png</Logo>
</Properties>
...
</Application>
</Applications>
```

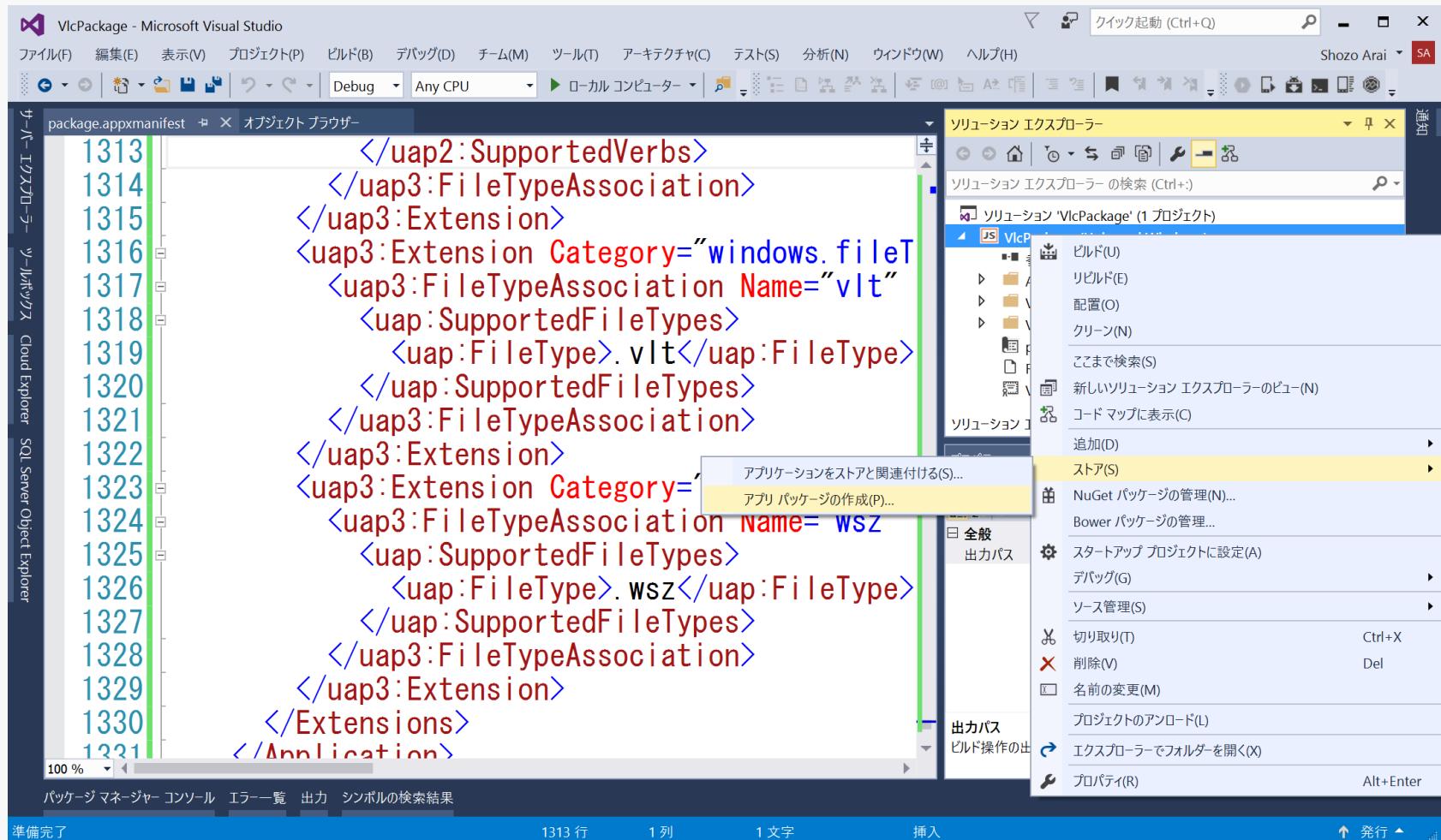
プロジェクトから不要なものを削除します

Package.appxmanifest の
編集が完了したら、
AppxManifest.xml をプロ
ジェクトから削除します



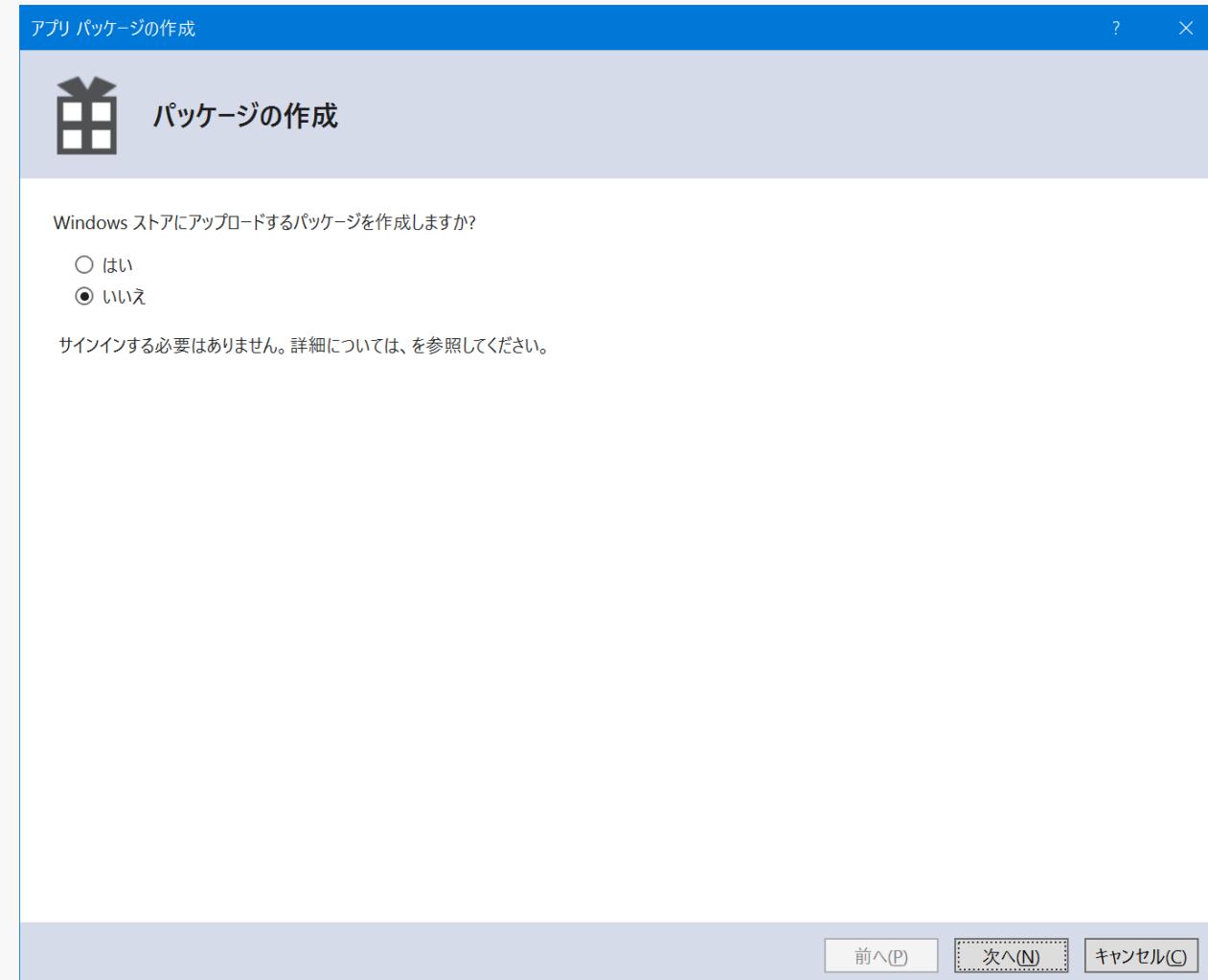
Appx パッケージの作成

[ストア]-[アプリ パッケージの作成] を選択します



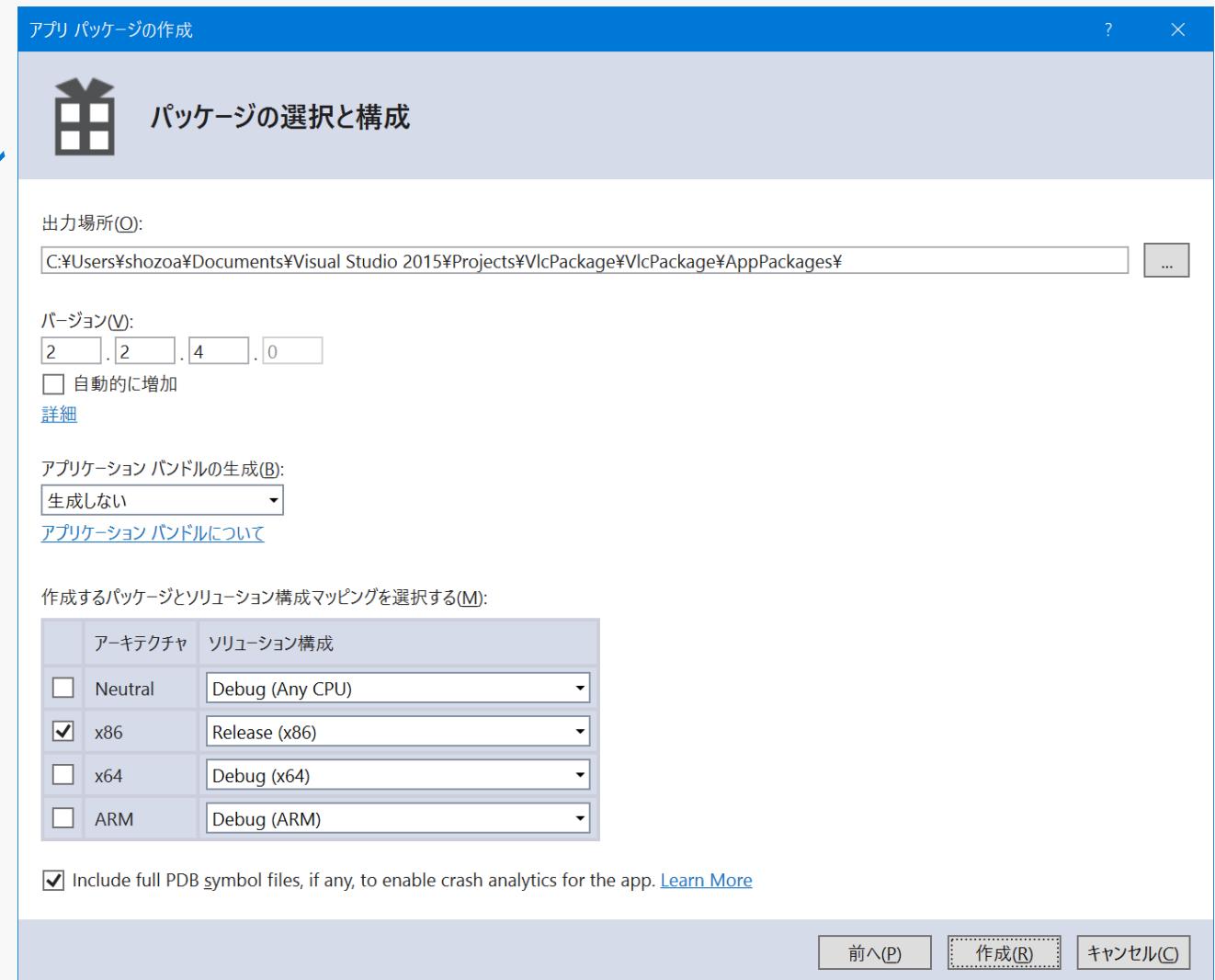
AppX パッケージを作成します

作成するパッケージが、テスト用途か提出用かを選択します



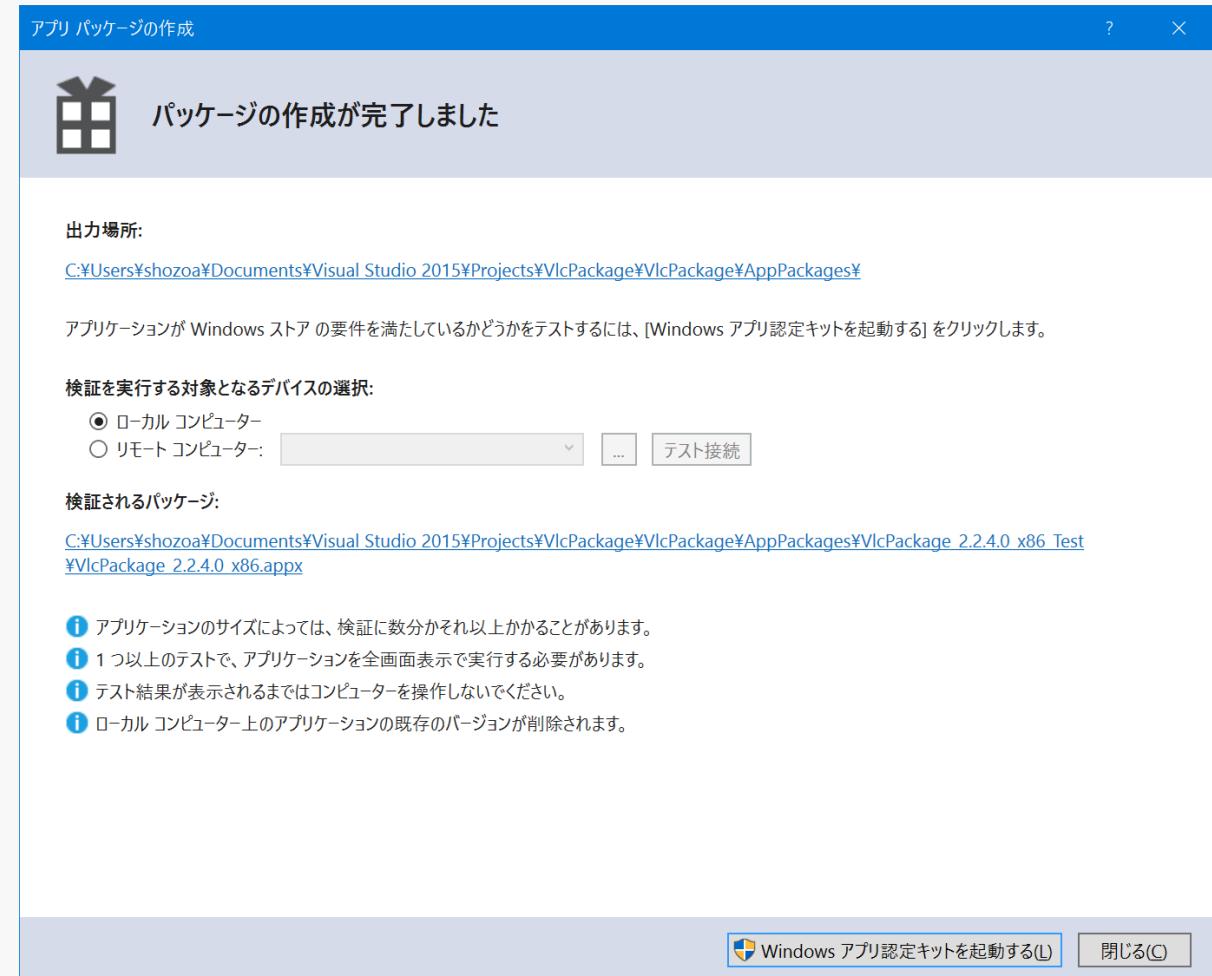
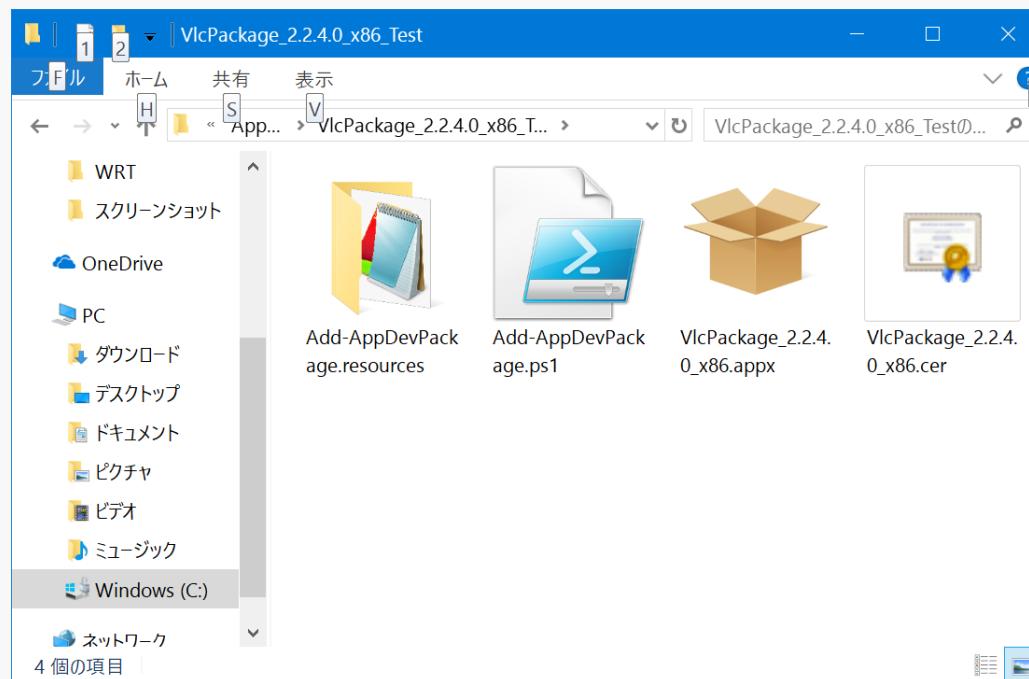
AppX パッケージを作成します

バージョン
アプリケーションバンドル
パッケージの構成
を選択します



AppX パッケージの完成

フォルダ内に証明書やインストール用のスクリプトができます



VLC ではエラーが発生します

エラー APPX3207 アプリケーションマニフェストで参照している 310x310 の正方形のロゴイメージ 'Assets¥AppLargeTile.scale-400.png' はイメージファイルの最大サイズを越えています。204800 バイトより大きくすることはできません

AppLargeTile.scale-400.png を削除してから、Appx パッケージを作成します

演習 7 のまとめ

Appx パッケージを Visual Studio で作成する方法
を学習しました

MakeAppx.exe ユーティリティを使う方法と違って、Visual Studio が PRI リソースの作成から Appx パッケージの作成まで行ってくれます。

また、Visual Studio がマニフェストに対するチェックを行います。

このプロジェクトでは、アプリのデバッグはできません。

