



Windows Phone アプリケーション開発

写真加工アプリケーションの作成



日本マイクロソフト株式会社
デベロッパー & プラットフォーム統括本部 監修

内容

写真加工アプリケーションの作成 1

プロジェクトの作成と準備 3

タイトルと背景の作成 4

写真表示画面の作成 5

アプリケーションバーの作成 6

読み込み処理の実装 8

フィルタ処理の実装 10

画像保存処理、バージョン情報表示の追加 11

アプリケーションを完成させる 13

Marketplace に公開する 15

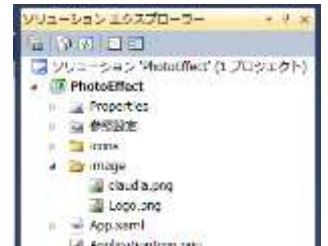
プロジェクトの作成と準備

1. 新しいプロジェクトを作成します。

- Visual C# → Silverlight for Windows Phone → **Windows Phone アプリケーション** を選び、アプリケーション名を入力して（Photo Effect 等）OK ボタンを押す
- OS のバージョンは Windows Phone OS 7.1 を選択する
- プロジェクトが開いたら、画面のレイアウトと、XAML のコードを並べて表示しておきます

2. ソリューションエクスプローラー を開いてプロジェクトを選択し、「プロジェクト」→「新しいフォルダー」でフォルダを作成し、「image」と名前を付けます。

- 以下のファイルを Image フォルダにドラッグ＆ドロップで追加します。
 - ✓ アイコン用の画像(Logo.png)
 - ✓ 重ね合わせ用の画像（Claudia.png）



3. ソリューションエクスプローラーを開いて、App.xaml を展開して App.xaml.cs を開き、以下のコードを追加します。

```
namespace PhotoEffect
{
    public partial class App : Application
    {
        :
        public App()
        {
            // キャッチできない例外のグローバル ハンドラーです。
            UnhandledException += Application_UnhandledException;

            :
            // デバッグ中にグラフィックスのプロファイル情報を表示します。
            if (System.Diagnostics.Debugger.IsAttached)
            {
                // 現在のフレーム レート カウンターを表示します。
                Application.Current.Host.Settings.EnableFrameRateCounter = false;

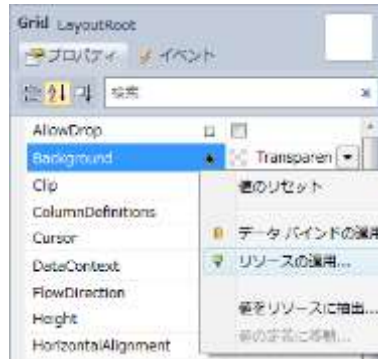
                :
                // アプリケーションが引き続き実行され、バッテリー電源が消耗します。
                PhoneApplicationService.Current.UserIdleDetectionMode = IdleDetectionMode.Disabled;
            }

            //デバイスの表示言語に合わせてアプリケーションの言語を設定します
            RootFrame.Language =
                System.Windows.Markup.XmlLanguage.GetLanguage(
                    System.Globalization.CultureInfo.CurrentUICulture.Name);
        }
    }
}
```

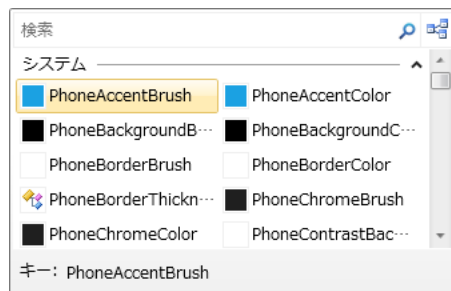
タイトルと背景の作成

1. XAMLコードの <Grid x:Name="LayoutRoot"... を選択して、**Background** プロパティを設定

✚ Background プロパティの右の◆を押して、「リソースの適応…」を選択



✚ 左上の「PhoneAccentBrush」を選択します。



2. 画面上にある「ページ名」を選択して、「Del」キーで削除します



3. 画面上の「マイアプリケーション」を選択して、以下のプロパティを設定 します。

✚ **Text** プロパティ：アプリケーション名を入力する

✚ **FontSize** プロパティ：20 → 48

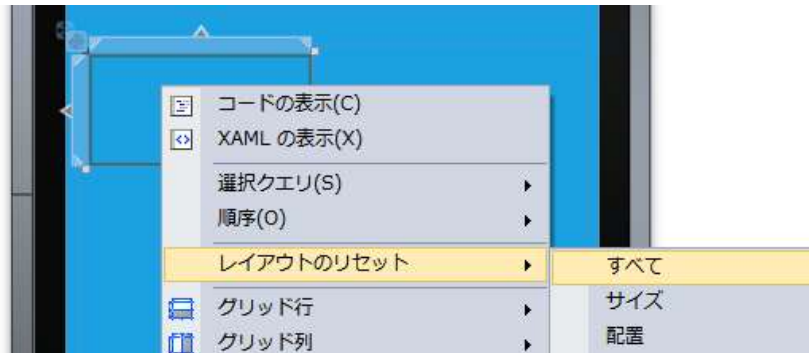
✚ **FontWeight** プロパティ：Bold

✚ **TextAlignment** プロパティ：Center

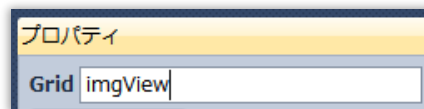


写真表示画面の作成 Visual Studio

1. 画面中央を選択
2. ツールボックスの **Grid** コントロールをダブルクリックし、画面に配置。
配置された Grid コントロールを右クリックし、
メニューから「レイアウトのリセット」→「すべて」を選択する

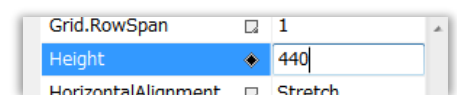


プロパティパネルの Grid の右の欄の名前を「imgView」に変更する



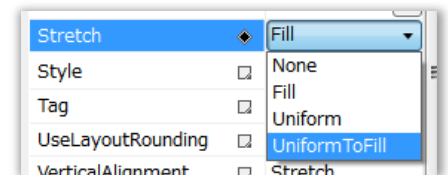
3. 同様に **Grid** コントロールをダブルクリックで配置し、「レイアウトのリセット」→「すべて」を選択する
そのままプロパティを設定

- 名前 (プロパティ欄の一番上) : grid2 → imgGrid
- Background プロパティ : White
- Height プロパティ : Auto → 440
- Width プロパティ : Auto → 440

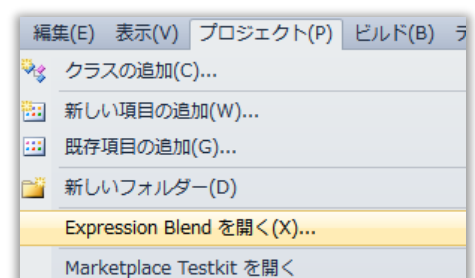


4. **Image** コントロールをダブルクリックで配置し、「レイアウトのリセット」→「すべて」を選択。
そのままプロパティを設定

- 名前 (プロパティ欄の一番上) : image1 → imgPhoto
- Source プロパティ : [...] ボタンを押して、Logo.png を選択
- Stretch プロパティ : Fill → UniformToFill

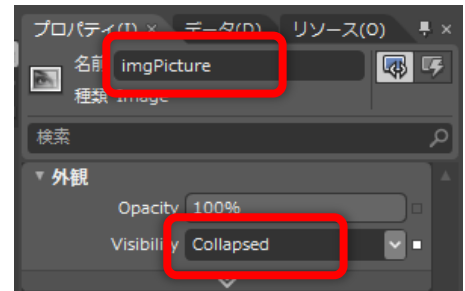
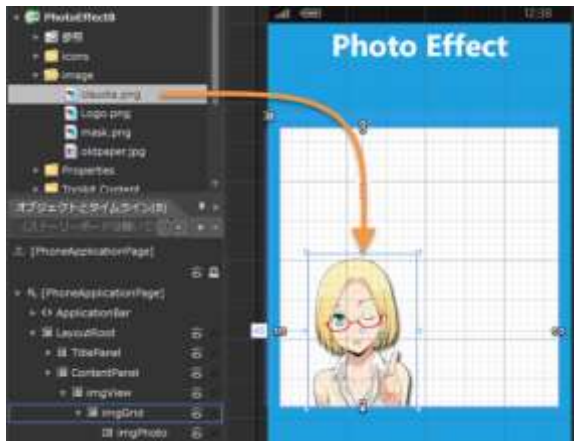


5. **F6** キーを押していったんビルトします。
6. 「プロジェクト」メニューから「Expression Blend を開く」を選択



写真表示画面の作成 Expression Blend

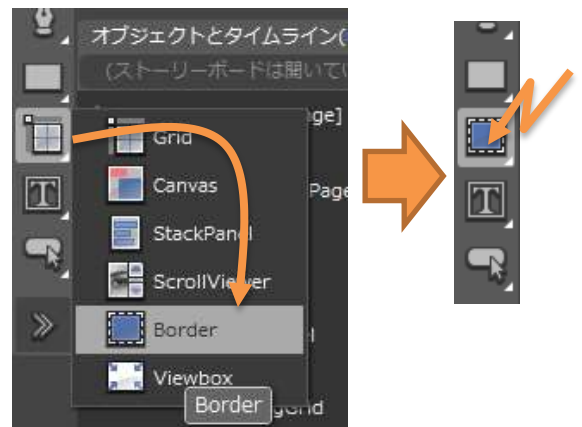
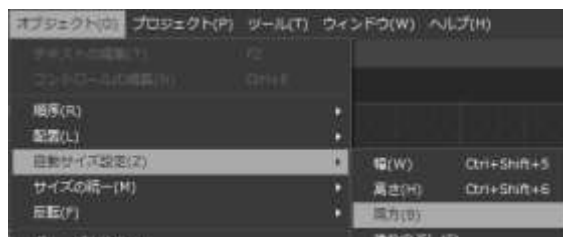
1. 「プロジェクト」の Image フォルダから「Claudia.png」を画面上にドラッグ&ドロップし、配置します。



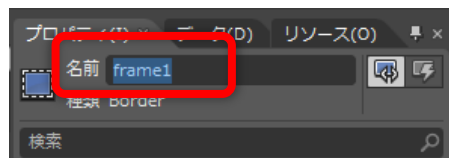
画面の右上の「名前」で「imgPicture」と名前を付けます。
また、「外観」パネルの、Visibility を Collapsed に変更します。

2. 画面左のツールボックスの中から、Border を選択して、アイコンが変わったらダブルクリックで配置します。

画面に配置されたら、「オブジェクト」メニュー から
「自動サイズ設定」→「両方」を選択します。

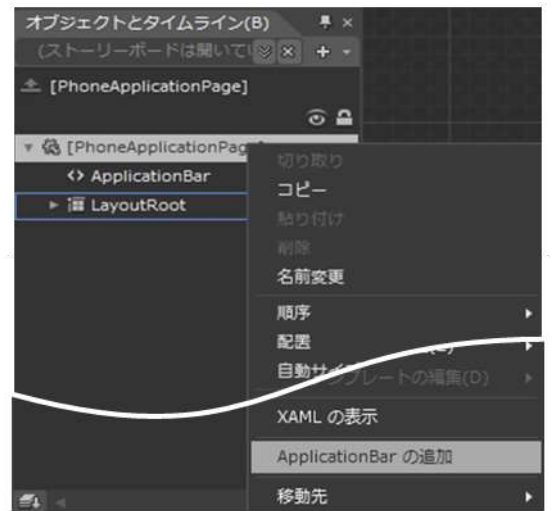


最後に、右上の「名前」欄で、frame と名前を付けます。



アプリケーションバーの作成

1. 画面左の「オブジェクトとタイムライン」パネルから「PhoneApplicationPage」を右クリックし、メニューの下にある「**ApplicationBarの追加**」を選択する

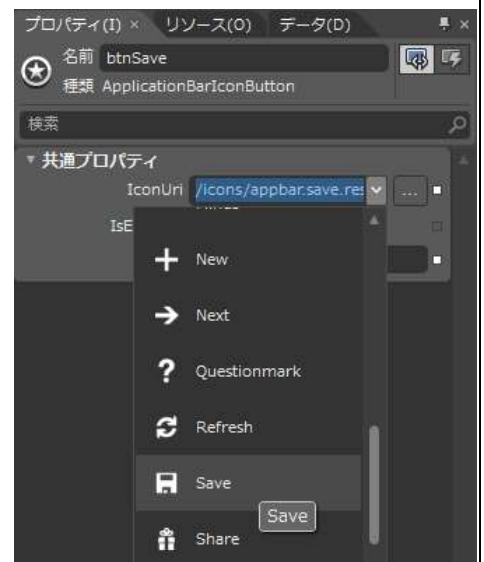


2. 作成された「ApplicationBar」を右クリックし、「**ApplicationBarIconButtonの追加**」を選択する。
ボタンが追加されたら、画面右の「プロパティ」パネルでアイコンとTextを設定する

- 名前：btnLoad
- IconUrlプロパティ：ドロップダウンを開いて、Cameraを選択する
- Textプロパティ：読み込み

3. 2で作成された「ApplicationBar」を右クリックし、「**ApplicationBarIconButtonの追加**」を選択したら、画面右の「プロパティ」パネルでアイコンとTextを設定する

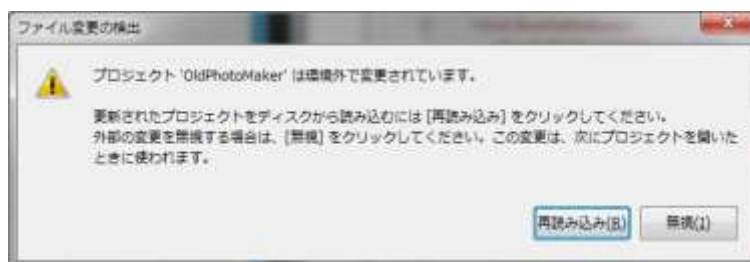
- 名前：btnSave
- IconUrlプロパティ：ドロップダウンを開いて、Saveを選択する
- Textプロパティ：保存



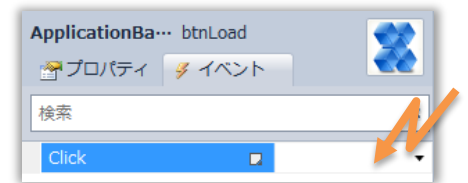
4. 2で作成された「ApplicationBar」を右クリックし、「**ApplicationBarMenuItemの追加**」を選択したら、画面右の「プロパティ」パネルでアイコンとTextを設定する

- 名前：mnuVersion
- Textプロパティ：バージョン情報

5. 「プロジェクト」メニューの「プロジェクトのビルド」を選んでいったんコンパイルする（Ctrl+Shift+B）
6. Visual Studio に切り替える。「ファイル変更の検出」ダイアログが表示されたら「再読み込み」を押す



読み込み処理の実装



1. XAMLコードから、`<shell:ApplicationBarIconButton x:Name="btnLoad">` を選択し、「プロパティ」パネルの「イベント」タブを開き、Click の右の空欄をダブルクリックして、関数を作成します

```
private void btnLoad_Click(object sender, EventArgs e)
{
}
}
```

2. 写真を選択するための機能 `PhotoChooserTask` を定義します。
尚、`task.Completed` の後は `+=` と入力し Tabキーを2回押すと、
`task_Completed` 関数が自動的に作成されます。

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Tasks;

namespace PhotoEffect
{
    public partial class MainPage : PhoneApplicationPage
    {
        PhotoChooserTask task = new PhotoChooserTask();

        // コンストラクター
        public MainPage()
        {
            InitializeComponent();
            task.Completed += new EventHandler<PhotoResult>(task_Completed);
            task.ShowCamera = true;
        }

        void task_Completed(object sender, PhotoResult e)
        {
            throw new NotImplementedException();
        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
        }
    }
}
```


3. 呼び出し処理を実装します。

```
private void btnLoad_Click(object sender, EventArgs e)
{
    task.PixelWidth = 440;
    task.PixelHeight = 440;
    task.Show();
}
```

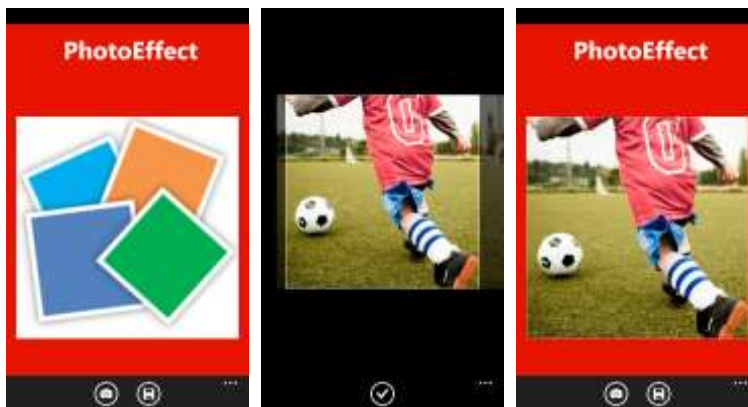
4. `throw new NotImplementedException();` を削除し、画像の読み込み処理を追加します。

```
using System.Windows.Media.Imaging;

:
// 【処理1-2】画像読み込み処理
void task_Completed(object sender, PhotoResult e)
{
    throw new NotImplementedException();
    if (e.TaskResult != TaskResult.OK)
        return;

    // 選択した画像をimageコントロールに表示する
    BitmapImage bmp = new BitmapImage();
    bmp.SetSource(e.ChosenPhoto);
    imgPhoto.Source = bmp;
```

5. ここでいったん実行してみます。F5キーを押してアプリケーションを起動してエミュレーターで確認します。
アプリケーションバーのカメラボタンを押して、画像を選択すると画像が読み込まれます。



フィルタ処理の実装

1. フィルタ処理を追加します。

今回は白枠の追加、画像を傾ける処理、そして配置した画像を表示する処理を追加しています。

```
void task_Completed(object sender, PhotoResult e)
{
    if (e.TaskResult != TaskResult.OK)
        return;

    //選択した画像をimageコントロールに表示する
    BitmapImage bmp = new BitmapImage();
    bmp.SetSource(e.ChosenPhoto);
    imgPhoto.Source = bmp;

    // 白枠を追加する
    frame.BorderBrush = new SolidColorBrush(Colors.White);
    frame.BorderThickness = new Thickness(15);

    // 傾ける
    RotateTransform rotate = new RotateTransform();
    rotate.Angle = 5;
    imgView.RenderTransformOrigin = new Point(0.5, 0.5);
    imgView.RenderTransform = rotate;

    // 画像の重ね合わせ
    imgPicture.Visibility = System.Windows.Visibility.Visible;
}
```

2. F5キーを押してエミュレーターで確認します。読み込まれた画像が加工されます。



画像保存処理、バージョン情報表示の追加

1. MainPage.xamlを開きます。

XAMLコードから、<shell:ApplicationBarIconButton x:Name="btnSave" を選択し、「プロパティ」パネルの「イベント」タブを開きます。Click の右の空欄をダブルクリックして、関数を作成します

```
private void btnSave_Click(object sender, EventArgs e)
{
}
}
```

2. 「プロジェクト」メニューから「参照の追加」選択します。

Microsoft.Xna.Framework を選択して、OKを押します。



3. 画像を保存するためのコードを追加します。

```
using System.IO;
using Microsoft.Xna.Framework.Media;
:
private void btnSave_Click(object sender, EventArgs e)
{
    //Grid から Jpeg 保存用ストリームを作成
    WriteableBitmap wp = new WriteableBitmap(imgGrid, null);
    MemoryStream stream = new MemoryStream();
    wp.SaveJpeg(stream, wp.PixelWidth, wp.PixelHeight, 0, 100);

    //PictureHub に保存する
    using (MediaLibrary lib = new MediaLibrary())
    {
        lib.SavePicture("PhotoEffect-" +
            DateTime.Now.ToString("yyyyMMddhhmmss"), stream.ToArray());
        MessageBox.Show("保存しました");
    }
}
```

4. F5キーを押してエミュレーターで確認します。写真を選択後保存ボタンで画像が保存されます。読み込みボタンを押せば、保存された画像を確認することができます。



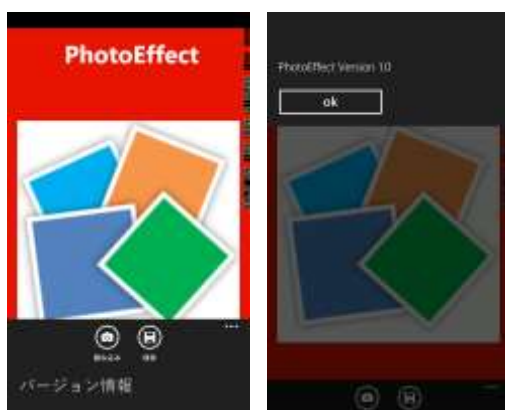
5. MainPage.xamlを開きます。
XAMLコードから、<shell:ApplicationBarItem x:Name="mnuVersion" を選択し、
「プロパティ」パネルの「イベント」タブを開きます。Click の右の空欄をダブルクリックして、関数を作成します

```
private void mnuVersion_Click(object sender, EventArgs e)
{
}
}
```

バージョン情報を表示するためのコードを追加します。

```
private void mnuVersion_Click(object sender, EventArgs e)
{
    MessageBox.Show("PhotoEffect Version 1.0");
}
}
```

6. F5キーを押してエミュレーターで確認します。アプリケーションバーのメニューからバージョン情報を表示することができます。



アプリケーションを完成させる

■アプリケーションアイコンの作成

Windows Phone Icon Maker で Icon を作ります。 <http://bit.ly/wpiconmaker>

Windows Phone Icon Maker で画像を読み込んで、アイコンにする領域をマウスでドラッグし、最後に Save Image ボタンを押すと 3 つのアイコンが作成されます。

- ✓ 300px : Marketplace にアップする際に必要なアイコン
- ✓ ApplicationIcon.png : アプリケーションプロジェクトで使われるアプリリスト用アイコン
- ✓ Background.png : アプリケーションプロジェクトで使われるタイル用アイコン
- ✓

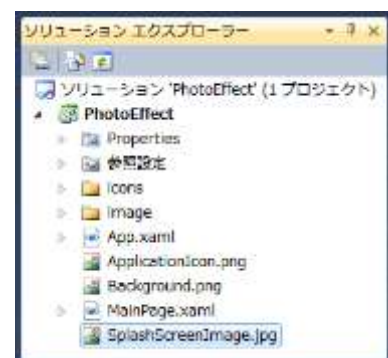


作成されたアイコンの中から、ApplicationIcon.png と、Background.png をプロジェクトファイルにドラッグ&ドロップして上書きします。

■スプラッシュスクリーンの準備

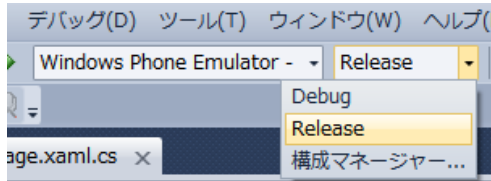
起動時に表示されるスプラッシュスクリーンは 480x800のjpeg画像です。このサイズとフォーマットのスプラッシュスクリーン用画像が用意できたら、SplashScreenImage.jpg と名前を付けて、プロジェクトファイルにドラッグ&ドロップして上書きします。

もし、必要がない場合は、SplashScreenImage.jpg をプロジェクトから削除します。



■リリースバージョンのビルド

ツールバーのビルドのバージョンをDebugからReleaseに変更して、F6を押してコンパイルします。



尚、Visual Studio Express を使用してこのメニューが表示されない場合は、「ツール」メニューから「設定」→「上級者用の設定」を選択してメニューを表示します。



■スクリーンショットの用意

エミュレーターからアプリケーションを起動します。（画面右上に数字が出ていないことを確認）



そして、エミュレーターの上のメニューの一番下の » ボタンをおしてエミュレーターの追加機能を表示します。

スクリーンショットタブを開いて、キャプチャボタンで撮影します。撮影したら「上書き保存…」ボタンで画像として保存します。（アイコン画像と同じフォルダに保存するとよいでしょう）スクリーンショット画像は1枚から8枚用意しておきます。



Marketplace に公開する

1. Windows Phone デベロッパーセンターを開く





-  <https://dev.windowsphone.com/ja-jp/home> を開く
-  画面左上の「新しいアプリを申請」を選ぶ

2. アプリの情報入力



アプリの情報

アプリのエイリアスを指定し、価格を設定して、その他の関連情報を入力してください




-  アプリのエイリアスの入力（これは自己管理用で、外には出ない名前です）
-  カテゴリの選択
-  価格の設定
-  配布する地域の選択
(推奨：コンテンツルールの厳しさがこれ以下のすべてのマーケットに配布する)

3. XAP のアップロードと説明



XAPのアップロードと説明

ここでは、アプリ内のXAPごとに説明を入力し、ストアでアプリを紹介するスクリーンショットをアップロードします。

-  アプリケーションの Xap ファイルのアップロード
通常は %USERPROFILE%\Documents\Visual Studio 2010\Projects\アプリ名\アプリ名\Bin\Release にあります。
-  アプリケーションの説明を記入
説明文の最後に「デザイン上の理由から一部は英語表記となっております」と入れておく
-  300x300 のアイコンとスクリーンショットをアップロード
「参照」を押して、すべてのアイコンとスクリーンショットをまとめてアップロード可能です

4. 必要があればオプションの設定



アプリ内広告を追加

広告から収入を得るための情報はすべてここに掲載されています。



マーケット選択とカスタム価格

アプリでは、国/地域によって異なる価格や購入可否を定義するオプションがあります。

5. 左下の申請ボタンで申請