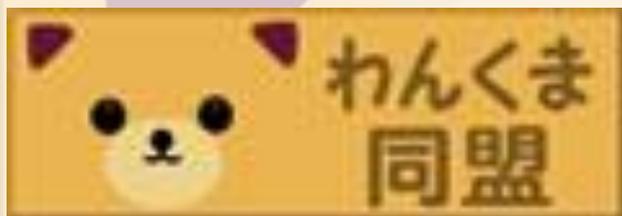


.NET UX Lab

.Net ユーザーエクスペリエンス研究所



XAML の勉強不要！ WPF アプリケーションは作れます

えムナウ (児玉 宏之)

Microsoft MVP for **Development Tools Visual C#**

.Net ユーザーエクスペリエンス研究所

わんくま同盟



わんくま同盟 東京勉強会 #21

WPF アプリケーション

アジェンダ

- 自己紹介
- はじめに
- 簡単なアプリケーションを作ってみる
- Binding 概要
- Linq to SQL とのバインディング
- アニメーション
- 3Dグラフィック
- まとめ

WPF アプリケーション

自己紹介

- 福井県越前市（武生市）出身、東京都杉並区在住。
- ソフトウェア業務歴28年・フリー技術者歴20年以上、でも「永遠の28歳」。
- 4 Bitワンチップからワークステーションまでのソフトを開発、開発言語も多種多様。
- 最近ではSQLサーバー + Windows Forms や WPFの Visual C# ソフト開発が多い。
- Microsoft MVP for Development Tools Visual C# を4年連続で受賞。
- INETAJ 理事、INETAJ スピーカビューロー登録。
- わんくま同盟他でスピーカ経験多数。



WPF アプリケーション

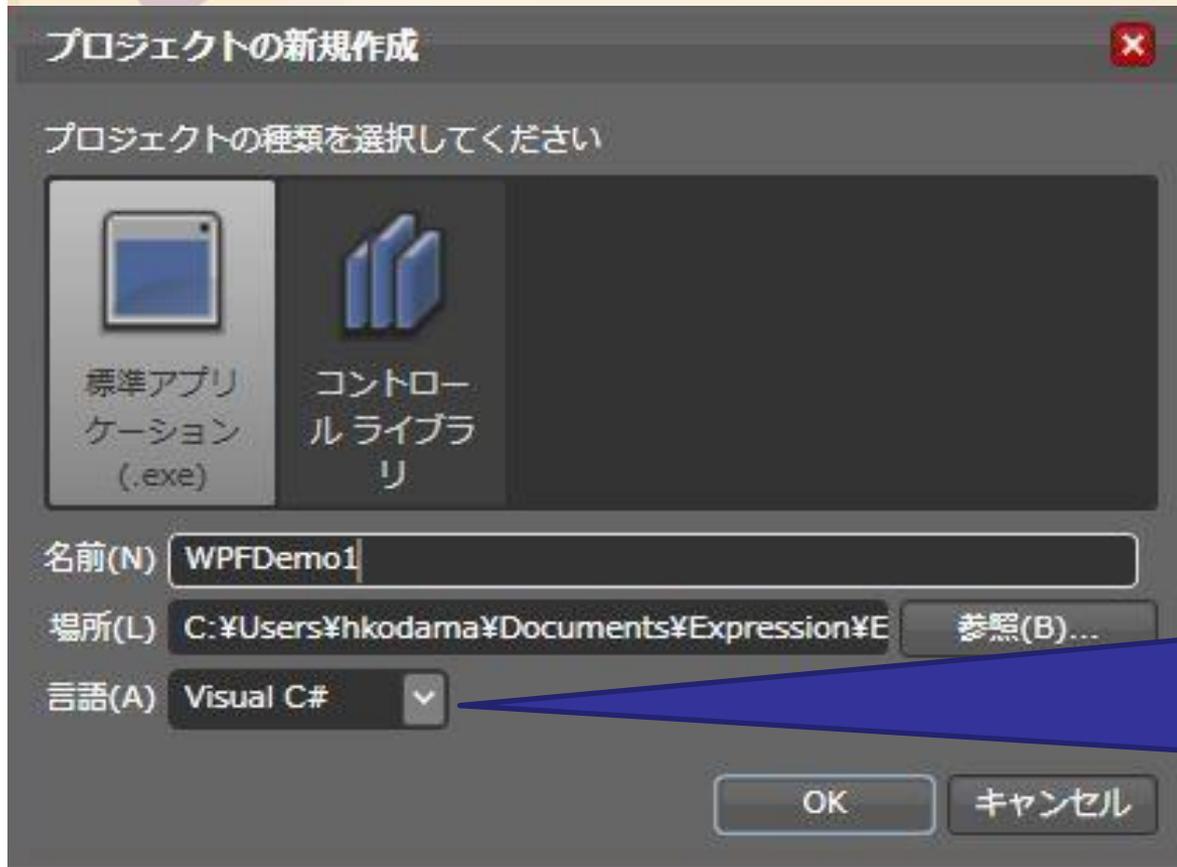
はじめに

- WPF や Silver Light の解説やデモを見て興味を持った方も多と思います。
- しかし、XAML を見て、大変そうだなってしり込みしている人も多いと聞きます。
- そこで、Expression Blend と Visual Studio のコラボレーションで XAML を書かないで WPF アプリケーションの開発が行えることを見てほしいと思います。

WPF アプリケーション

簡単なアプリケーションを作ってみる

- まず Expression Blend を起動して新しいプロジェクトを作成します。

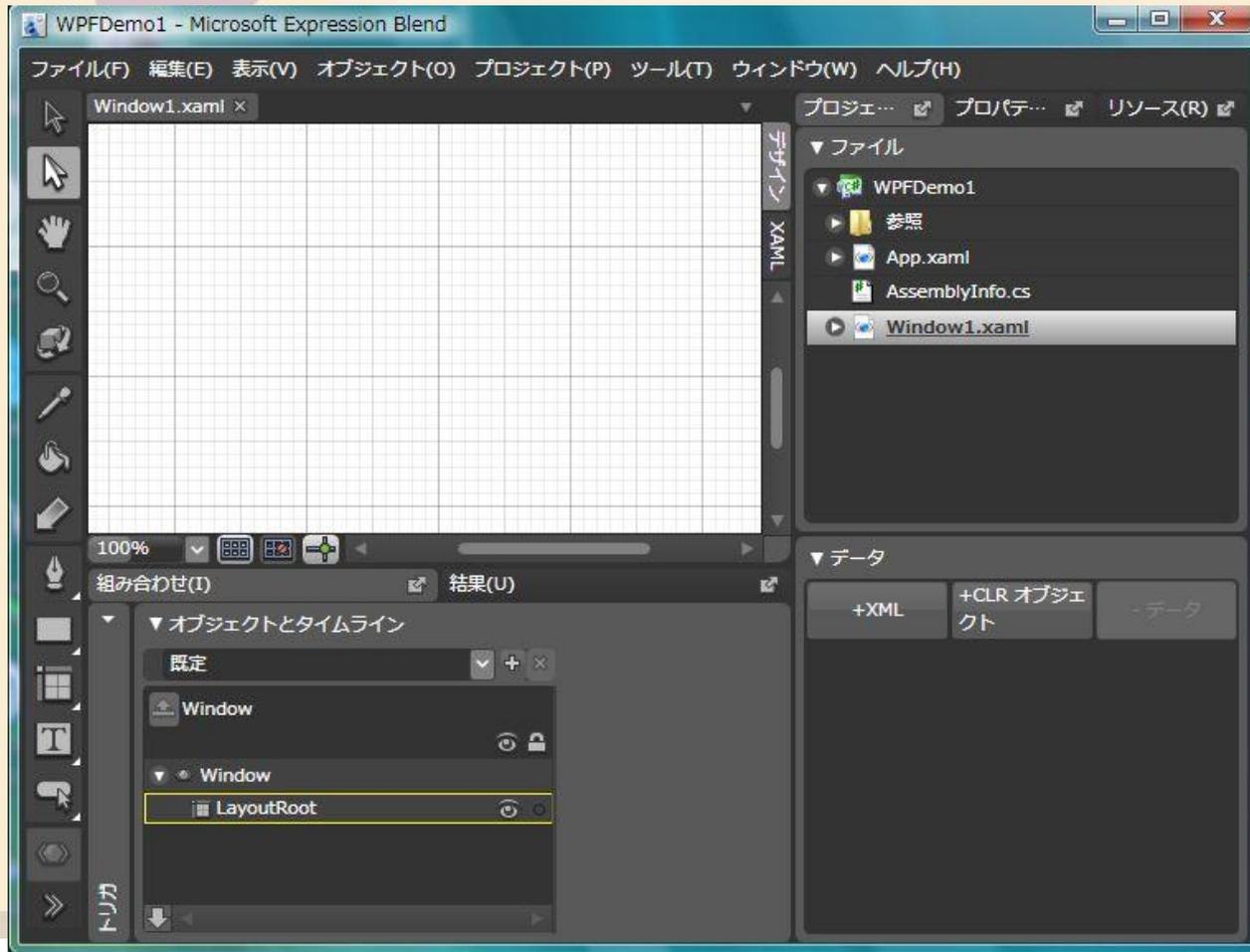


Visual C# と
Visual Basic が
選択可能です。

WPF アプリケーション

簡単なアプリケーションを作ってみる

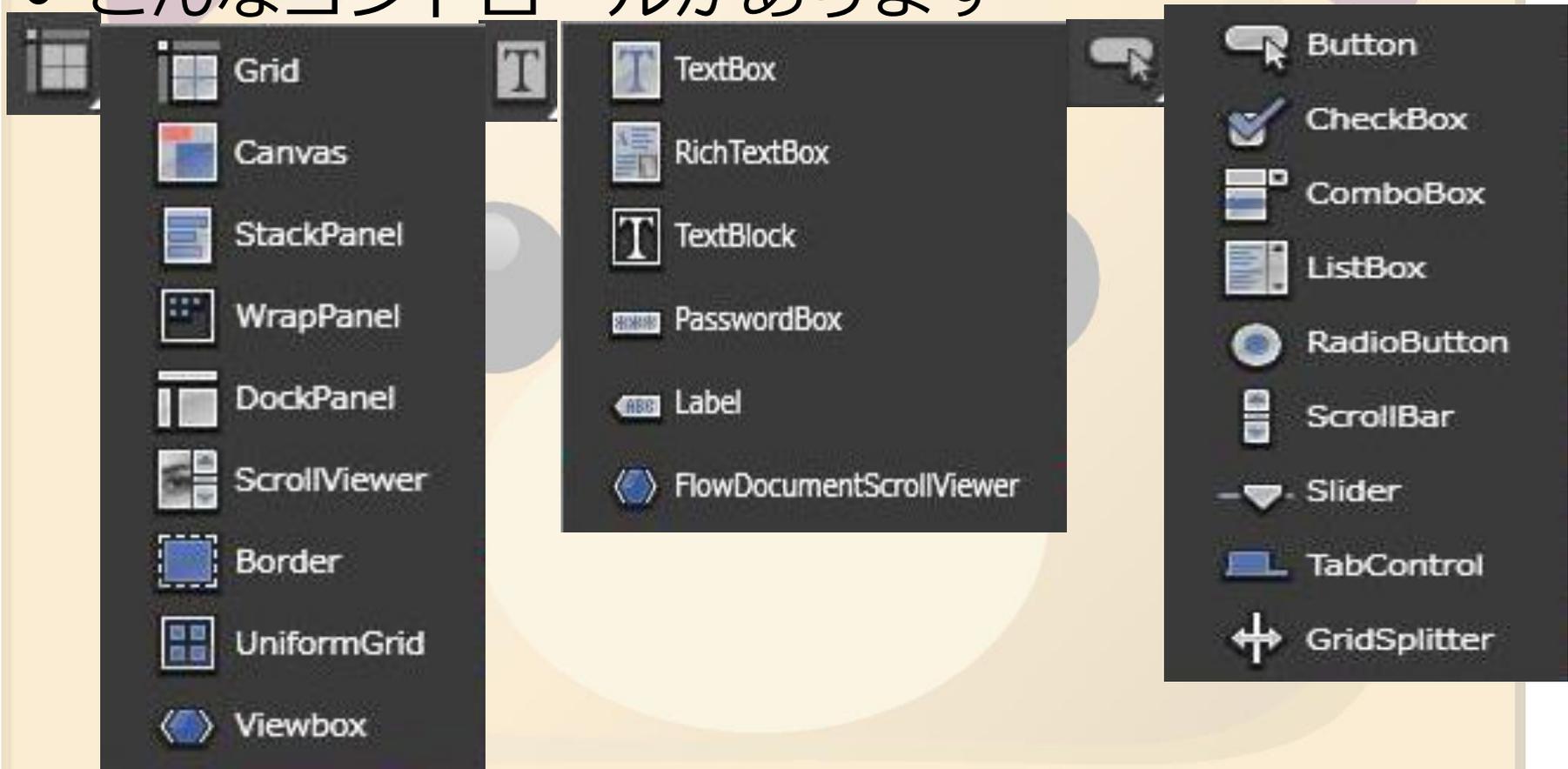
- こんな画面が表示されます。



WPF アプリケーション

簡単なアプリケーションを作ってみる

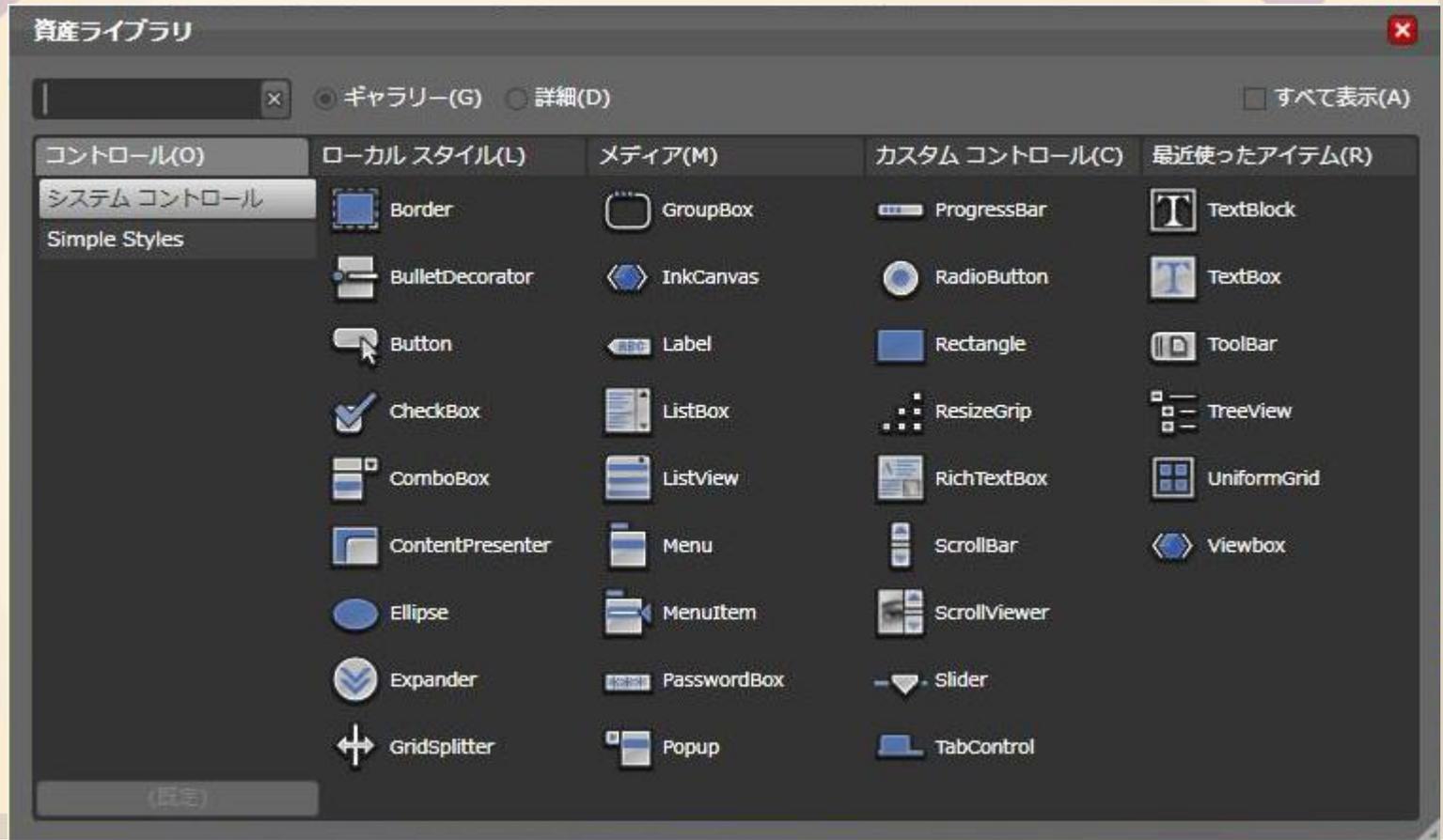
- こんなコントロールがあります



WPF アプリケーション

簡単なアプリケーションを作ってみる

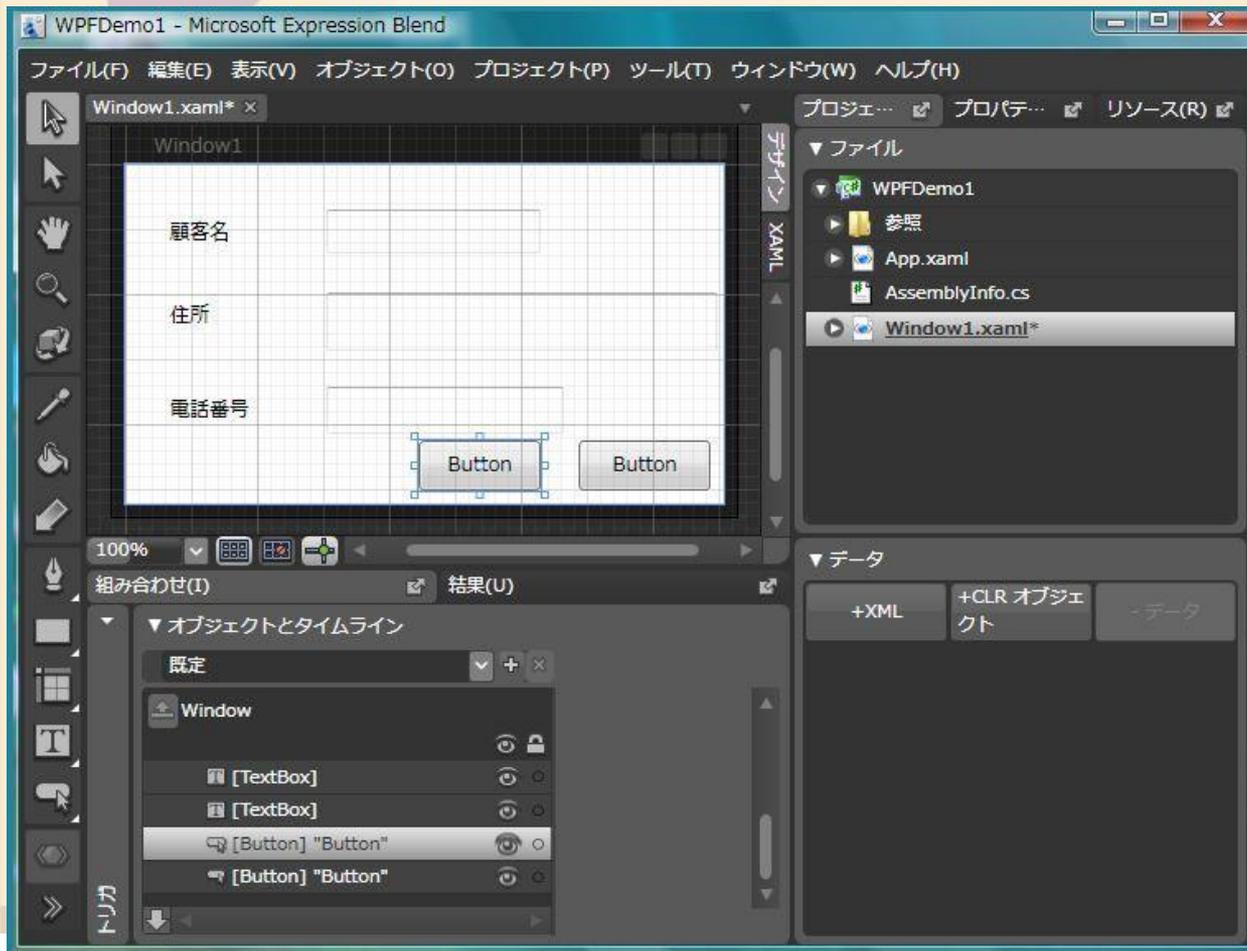
- 資産ライブラリにもコントロールはあります



WPF アプリケーション

簡単なアプリケーションを作ってみる

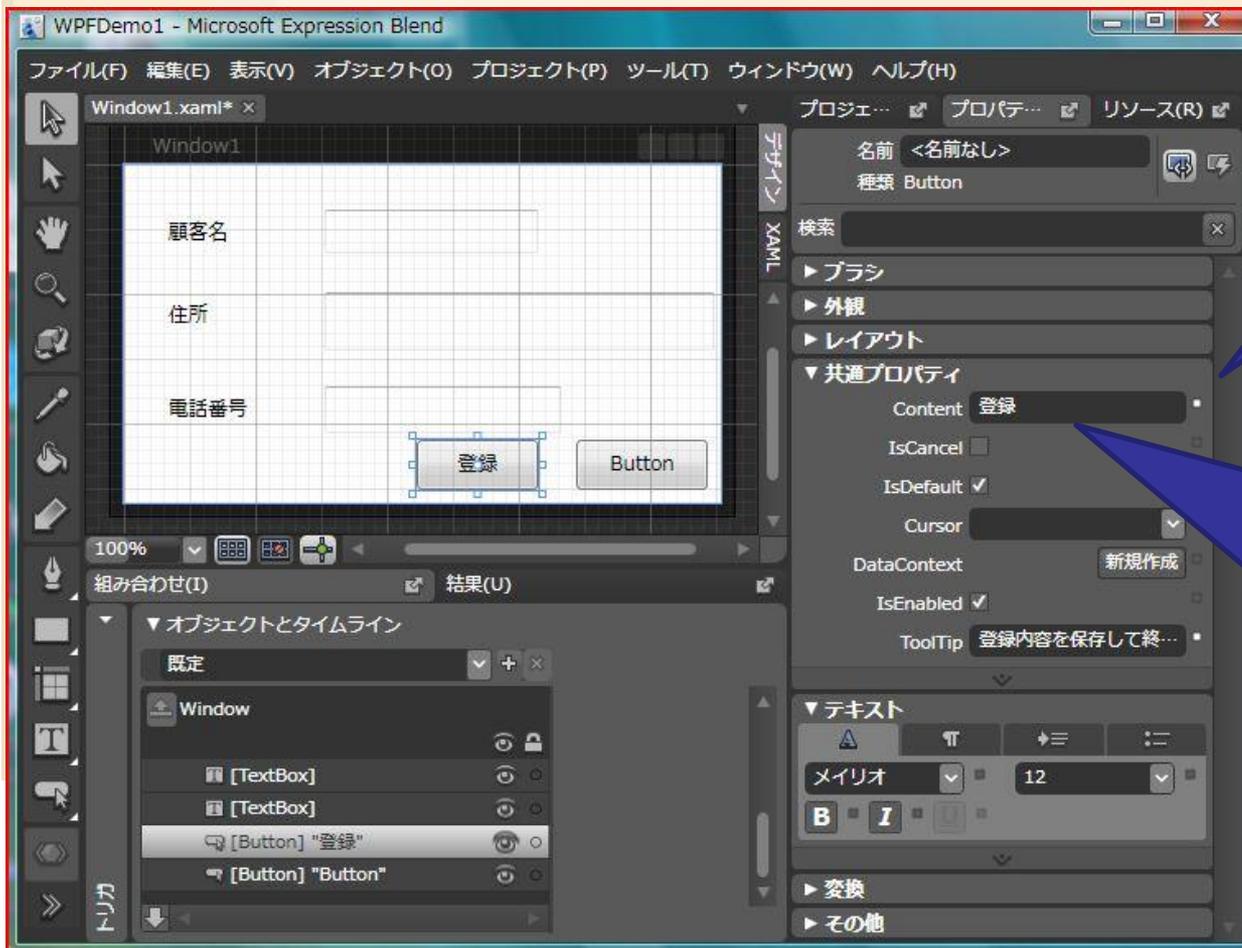
- コントロールを配置します。



WPF アプリケーション

簡単なアプリケーションを作ってみる

- プロパティを修正します。



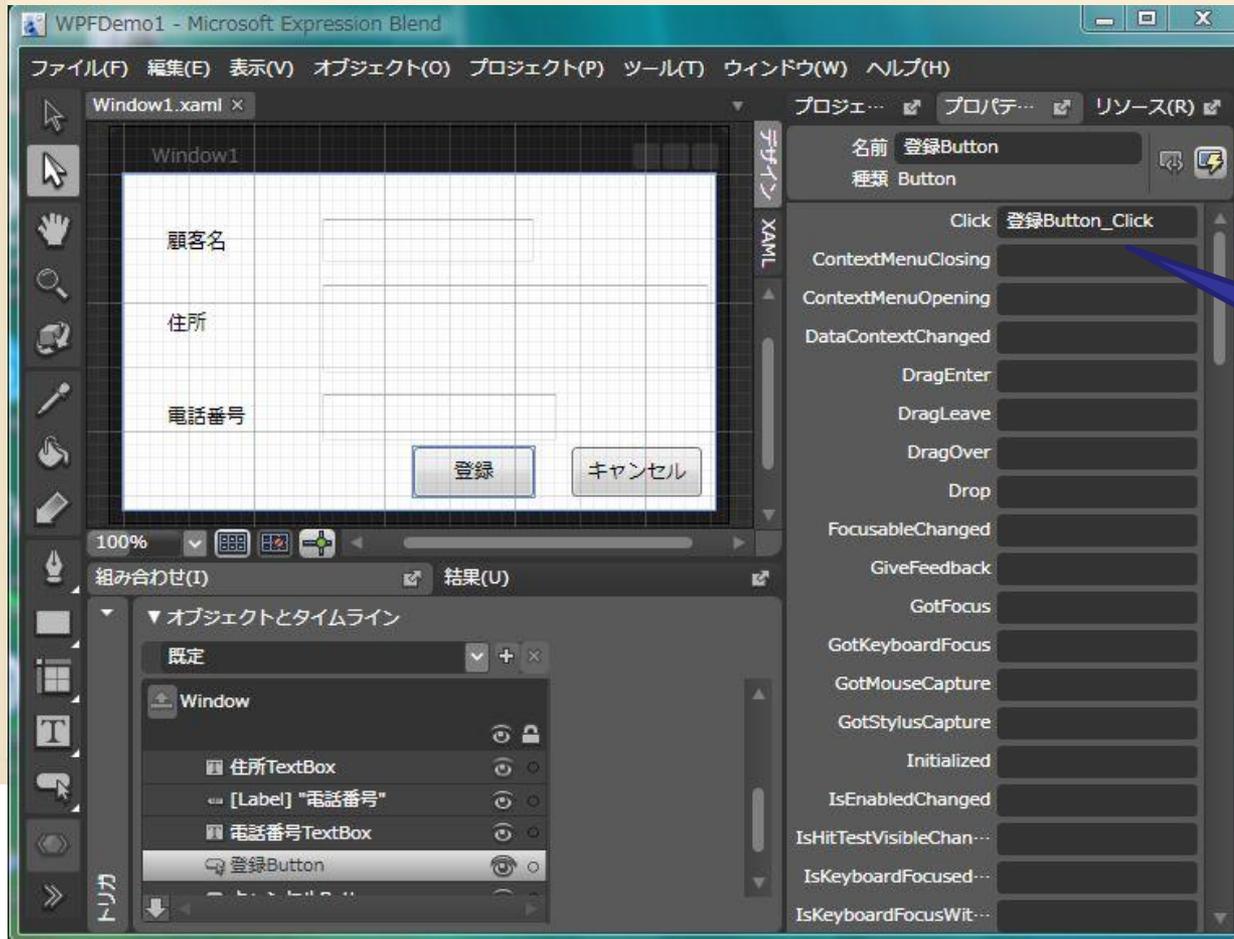
修正した(された)プロパティは白丸が付きます。

修正内容は右クリックメニューでリセット可能です(もちろんUNDOも可能)。

WPF アプリケーション

簡単なアプリケーションを作ってみる

- イベントを作成します。

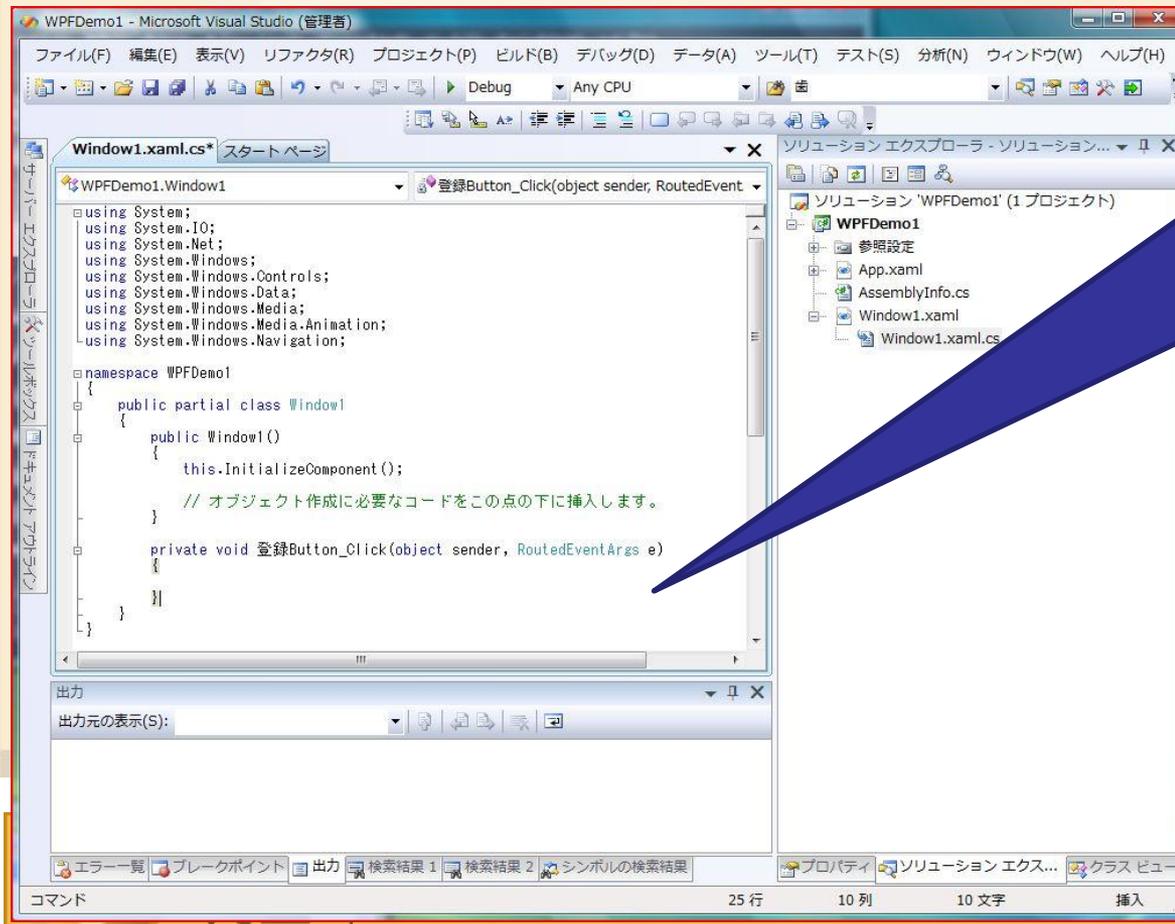


ここでプロパティ
とイベントを切り
替えます。

空白状態からダ
ブルクリックで
自動的に作成さ
れます。

WPF アプリケーション 簡単なアプリケーションを作ってみる

- Visual Studio が起動します。



自動的に
Visual Studio
が起動されて作
成されたイベン
トが表示されま
す。

WPF アプリケーション

簡単なアプリケーションを作ってみる

- こんな風に Expression Blend でコントロールを配置していったってプロパティを変更して画面を作成します。
- イベント処理は Visual Studio でプログラムを作成していきます。
- 簡単な画面はこのようにして作成できます。
- Expression Blend と Visual Studio の切り替え時には必ず保存してください、切り替えたときに変更があったことを自動的に認識して再読み込みのダイアログが出て再読み込みします。

DEMO

簡単なアプリケーションを作ってみる

えムナウ (児玉 宏之)

Microsoft MVP for **Development Tools Visual C#**

.Net ユーザーエクスペリエンス研究所

わんくま同盟



わんくま同盟 東京勉強会 #21

WPF アプリケーション

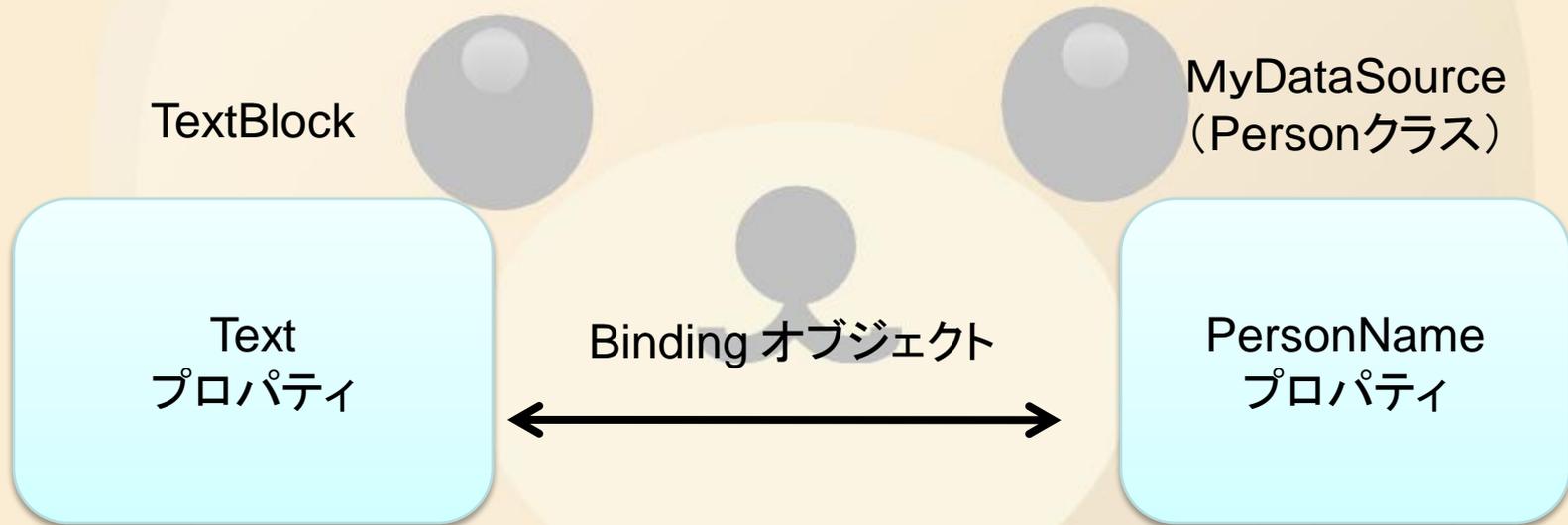
Bindingの概要

- Windows Presentation Foundation (WPF) データ バインディングは、アプリケーションがデータを提供し、UIと結びつけることで、柔軟な UI 表現、ビジネス ロジックと UI の明確な分離を実現します。
- データ フローの方向やソースの更新の要因を選択でき、表示のためのデータ変換や格納時のデータの検証を行えます。

WPF アプリケーション

Bindingの概要

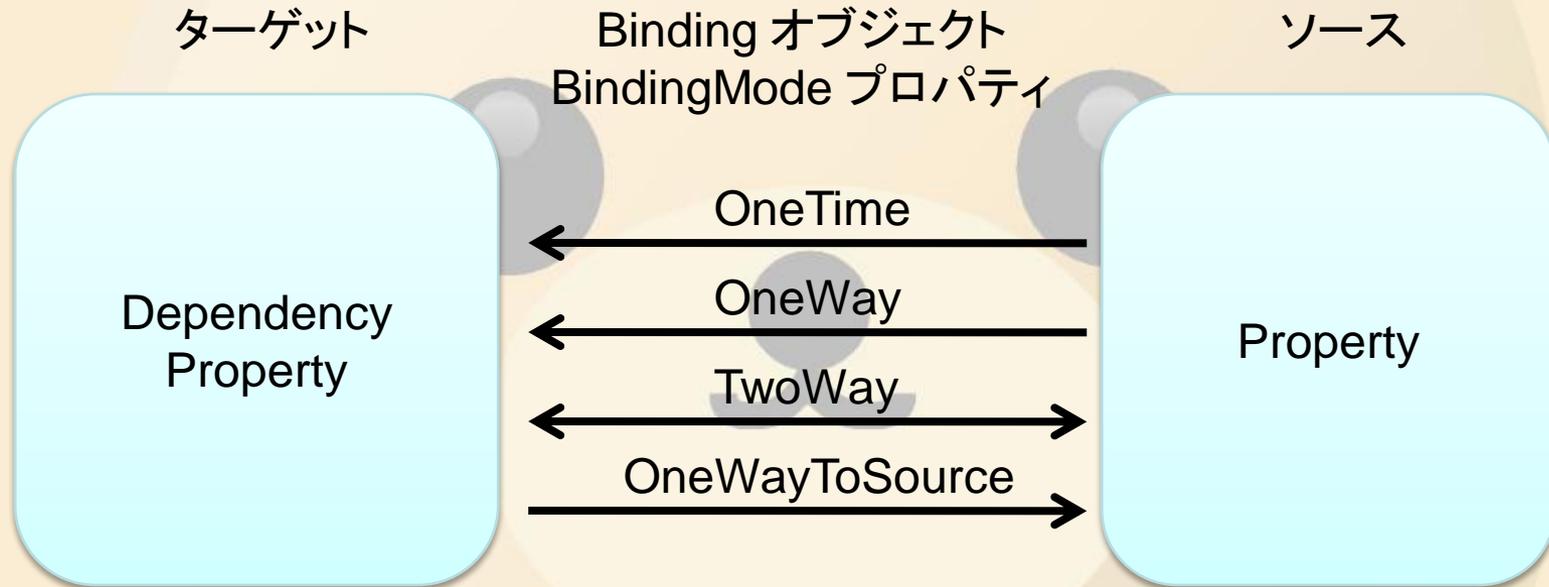
- Binding オブジェクトでコントロールのプロパティとデータソースのプロパティを結びつけます。



WPF アプリケーション

Bindingの概要

- データの方向を決めることができます。

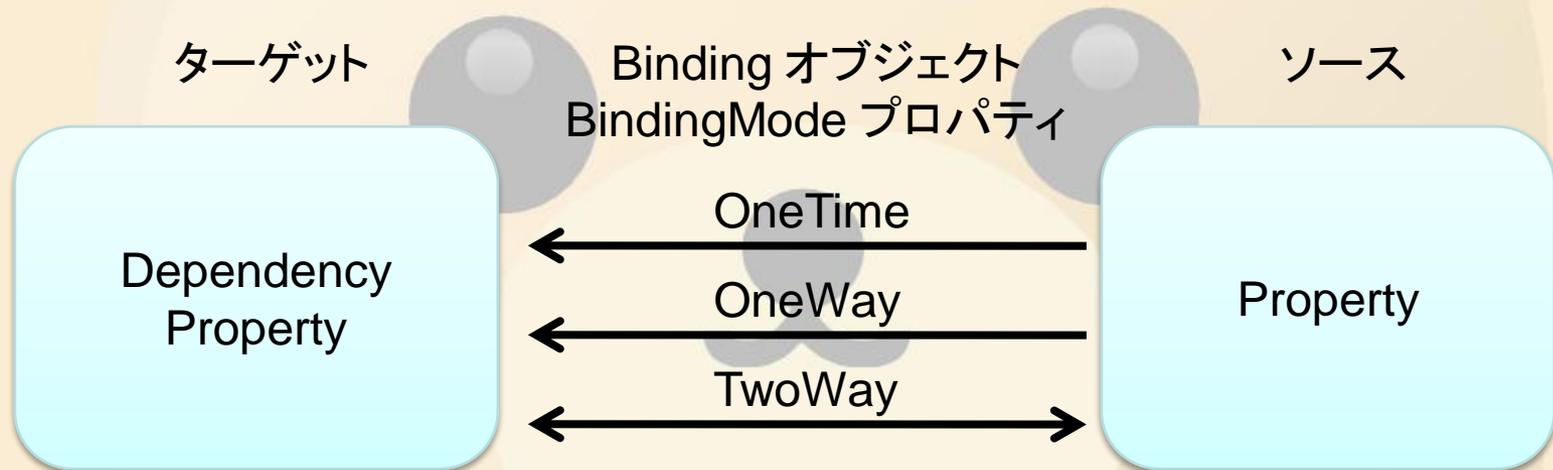


BindingModeでDefault はテキスト ボックスやチェック ボックスなど編集可能な場合はTwoWay、それ以外のほとんどのプロパティはOneWay。

WPF アプリケーション

Bindingの概要

- ターゲットのコントロールのプロパティの更新タイミングを設定できます。

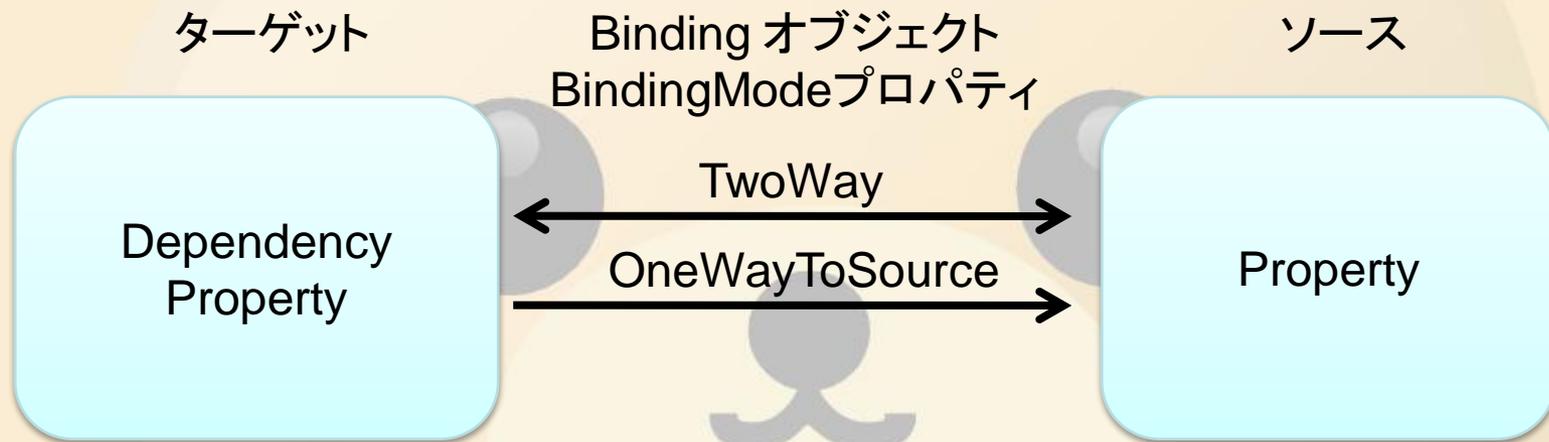


OneTimeはアプリケーションの起動時またはデータ コンテキストの変更時
OneWay・TwoWayはプロパティ変更時

WPF アプリケーション

Bindingの概要

- ソースのプロパティの更新タイミングを設定できます。



Binding.UpdateSourceTrigger プロパティ
PropertyChanged の場合ターゲットプロパティ変更時、
LostFocus の場合ターゲットがフォーカスを失った時、
Explicit の場合アプリケーションが UpdateSource を呼び出した時。

WPF アプリケーション

Bindingの概要

- DataContext や ItemsSource で扱うデータをまとめて設定できます。

顧客ListView. ItemsSource
(親要素)

代入

顧客クラス
インスタンス

TextBlock.Text
プロパティ
TextBlock.Text
プロパティ
TextBlock.Text
プロパティ
TextBlock.Text
プロパティ
TextBlock.Text
プロパティ
TextBlock.Text
プロパティ

Binding オブジェクト

Binding オブジェクト

Binding オブジェクト

Binding オブジェクト

Binding オブジェクト

Binding オブジェクト

ID
プロパティ
ふりがな
プロパティ
氏名
プロパティ
誕生日
プロパティ
都道府県ID
プロパティ
郵便番号
プロパティ

WPF アプリケーション Bindingの概要

- Expression Blend のデータソース
 - XML データ ソース
 - XML 形式のデータを提供できるローカルまたはリモートの XML ファイル
 - 共通言語ランタイム (CLR) オブジェクト データ ソース
 - ObservableCollection クラスが提供
 - IEnumerable インターフェイスと
INotifyCollectionChanged インターフェイスと
INotifyPropertyChanged インターフェイスを実装したクラス
 - Linq to SQL データ

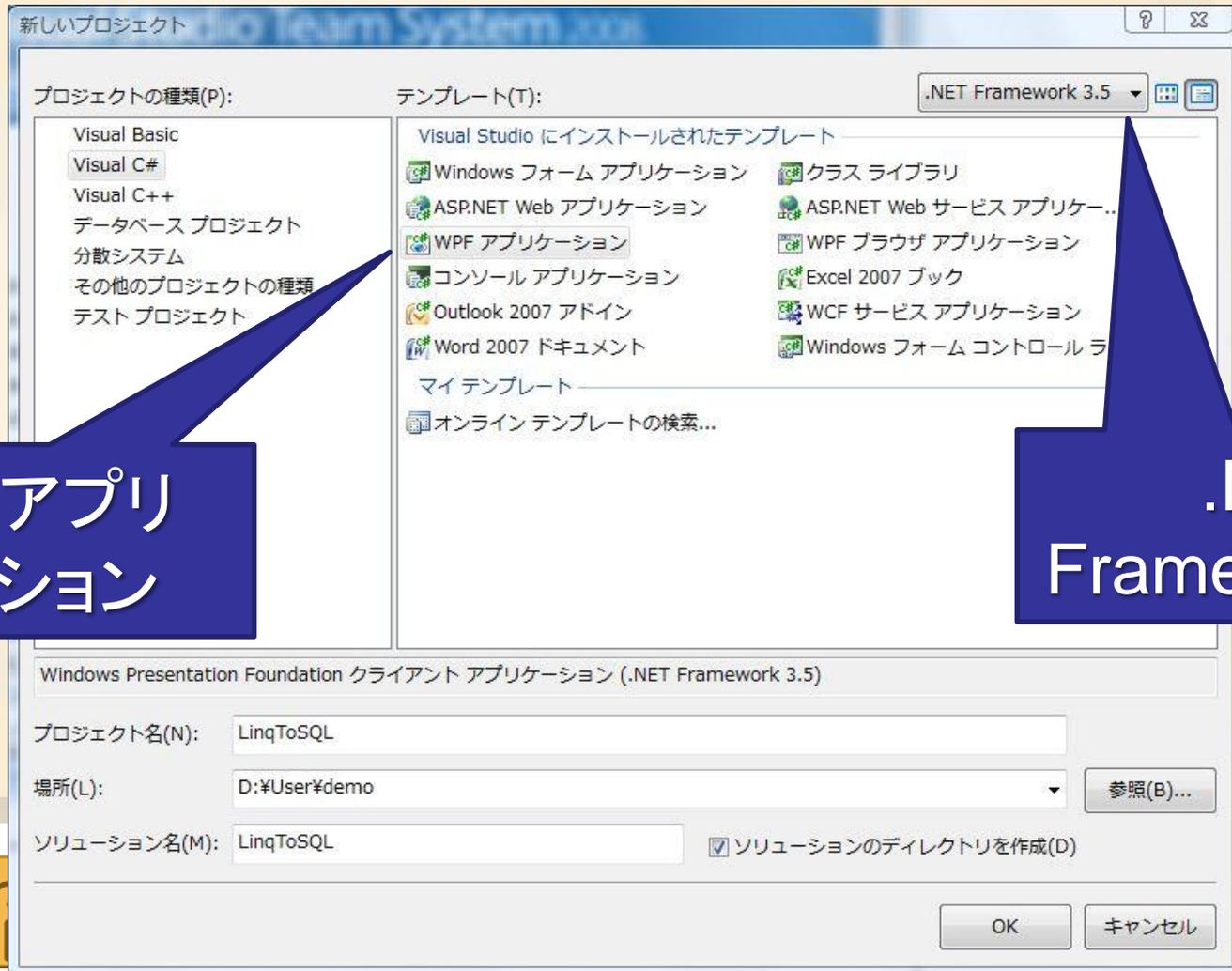
WPF アプリケーション

Bindingの概要

- WPF のデータソース
 - Expression Blend のデータソースの2つ
 - ADO.NET データ
 - ADO.NET DataView は、IBindingList を実装し、バインディング エンジンがリスンする変更通知を提供
 - DependencyObject
 - WPF のコントロールは全て DependencyObject
 - 任意の DependencyObject の依存関係プロパティにバインド、コントロールのプロパティ間のバインディングに主に使用
 - Blend ではプロパティウィンドウからデータバインドで設定可能

WPF アプリケーション Linq to SQL とのバインディング

- Visual Studio で WPF アプリケーションを作成。



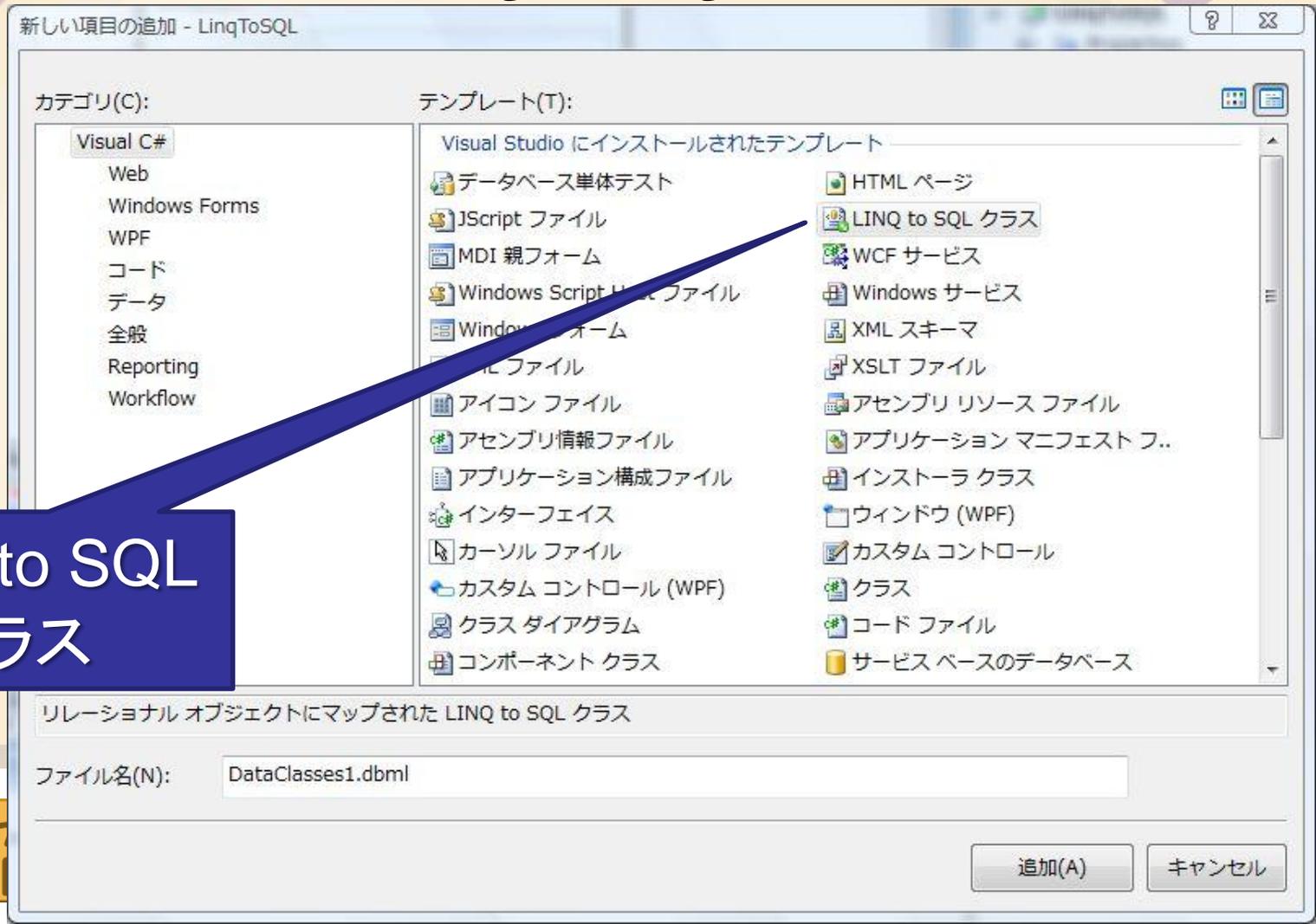
WPFアプリ
ケーション

.NET
Framework 3.5



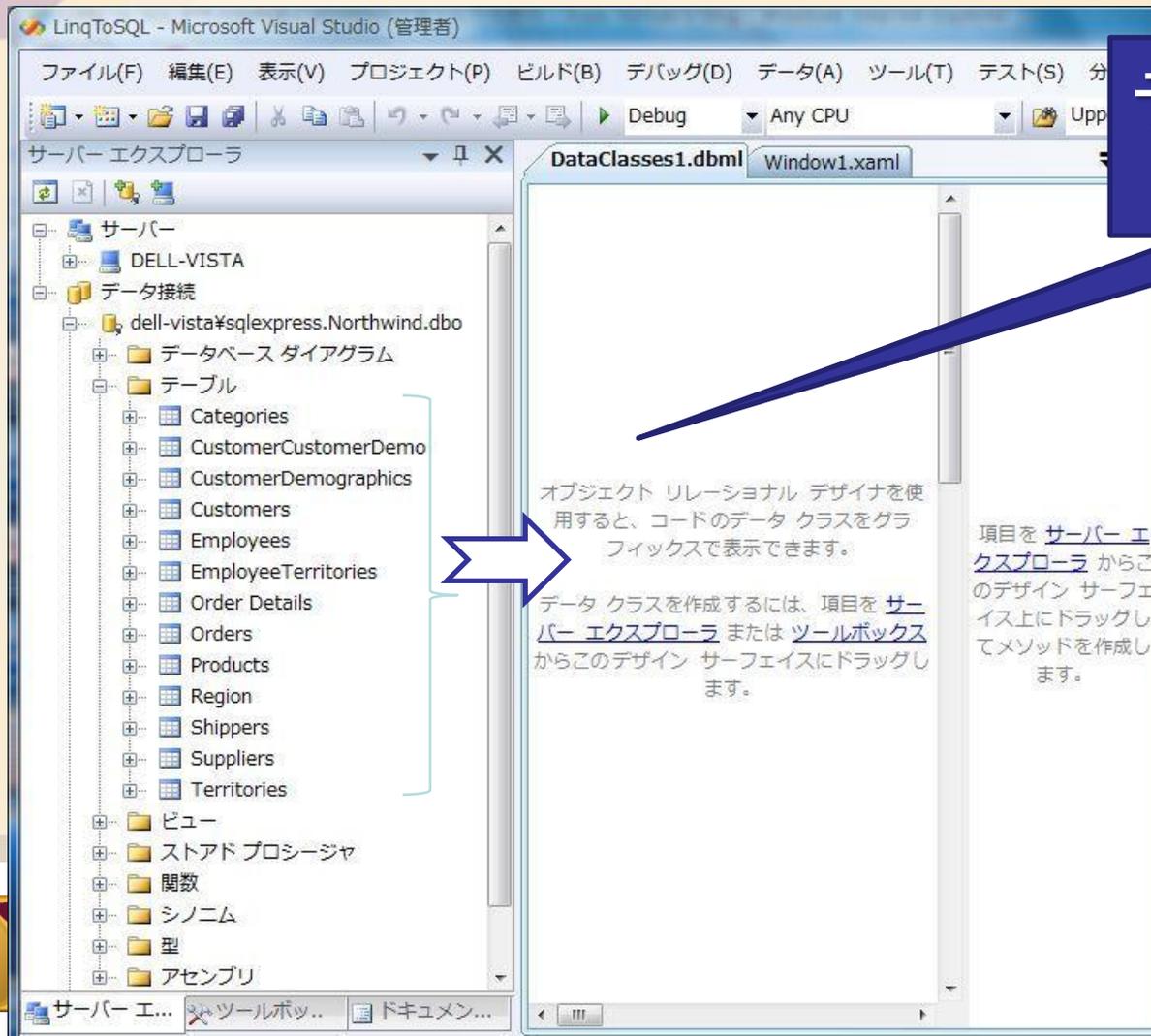
WPF アプリケーション Linq to SQL とのバインディング

- プロジェクトに LINQ to SQL クラス追加。



WPF アプリケーション Linq to SQL とのバインディング

- データ接続からテーブルをドラッグ。

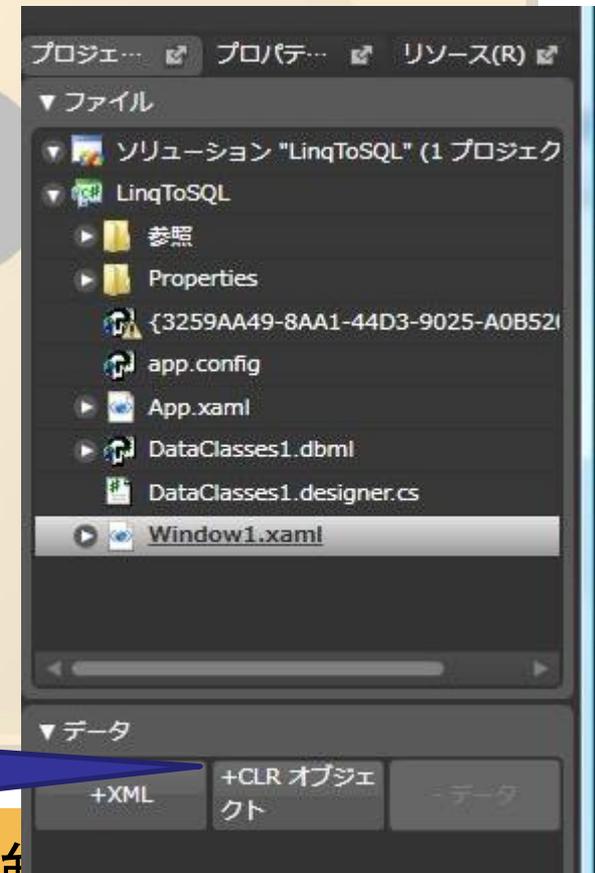


テーブルをここにドラッグ



WPF アプリケーション Linq to SQL とのバインディング

- Expression Blend で同じプロジェクトを読み込みます。
- ListBox を追加して listbox1 と名前を付けておきます。
- Blend に作成した LINQ to SQL クラスを追加するには、データ + CLR オブジェクト をクリックします。

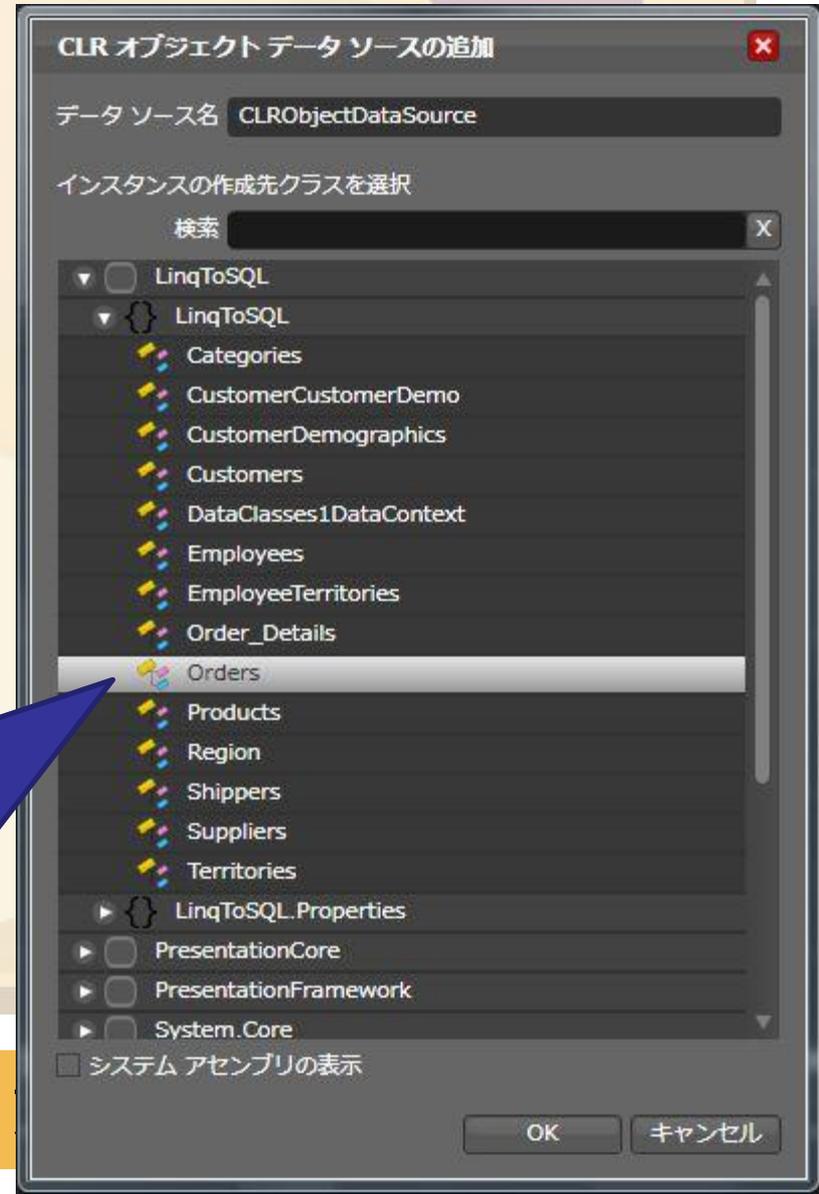


データ +CLR オブジェクト

WPF アプリケーション Linq to SQL とのバインディング

- ListBox に表示したいテーブルを選択して、CLR オブジェクト データソースを作ります。

LinqToSQL アセンブリの
LinqToSQL 名前空間の
Orders クラス

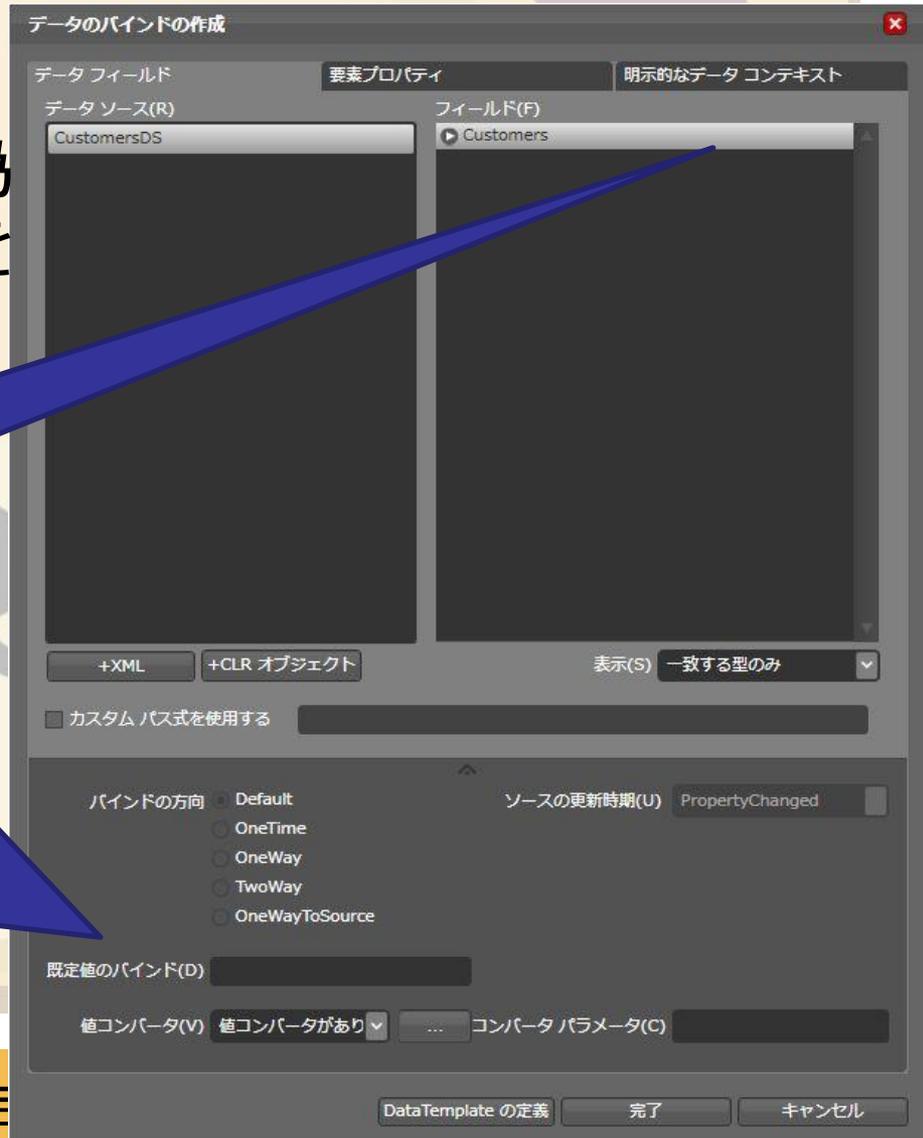


WPF アプリケーション Linq to SQL とのバインディング

- ここでは複数の項目をまとめてバインドする為データソースのクラスを選択し DataTemplate の定義を押します。

CustomerDS データソース
Customers クラス

バインドの方向
ソースの更新タイミング
規定値・値コンバータ
を指定できます



WPF アプリケーション Linq to SQL とのバインディング

- 表示したい項目を選ぶと自動的にデータテンプレートを作成します。

データテンプレートの作成

Listbox で表示するデータ フィールドを選択してください。さらに、各フィールドを表すコントロールの種類と、そのコントロールのプロパティを選択してください。プロパティは各フィールド内のデータに接続されます。

既定の Listbox スタイリング

現在または定義済みのデータ テンプレート: Customersテンプレート

新しいデータ テンプレートと表示フィールド

名前: Customersテンプレート1

フィールド	コントロールの種類
<input type="checkbox"/> Address : (String)	Stack
<input type="checkbox"/> City : (String)	Textf
<input checked="" type="checkbox"/> CompanyName : (String)	Textf
<input checked="" type="checkbox"/> ContactName : (String)	Textf
<input type="checkbox"/> ContactTitle : (String)	Textf
<input type="checkbox"/> Country : (String)	Textf
<input type="checkbox"/> CustomerCustomerDemo(配列) : (EntitySet<CustomerCustomerDemo>)	Item

サンプル データの生成

OK キャンセル

CompanyName
ContactName
を表示させます。



WPF アプリケーション

```
public partial class Window1 : Window
{
    private DataClasses1DataContext
        dataContext = new DataClasses1DataContext();

    public Window1()
    {
        InitializeComponent();
        this.listbox1.ItemsSource = dataContext.Customers;
    }
}
```

ローカルに
DataContext を
用意します

ItemSource に
Customers を
設定します

WPF アプリケーション

Binding

- クラスのプロパティと UI のプロパティを Binding オブジェクトで結びつければ、データバインディングできます。
- クラスのインスタンスを UI の DataContext や ItemSource に設定して、クラスのプロパティ名と UI のプロパティ名 を結びつければ、データバインディングできます。
- ListBox や ListView などは、DataTemplate でデータの表示を設定できます。

DEMO

Binding

えムナウ (児玉 宏之)

Microsoft MVP for **Development Tools Visual C#**

.Net ユーザーエクスペリエンス研究所

わんくま同盟



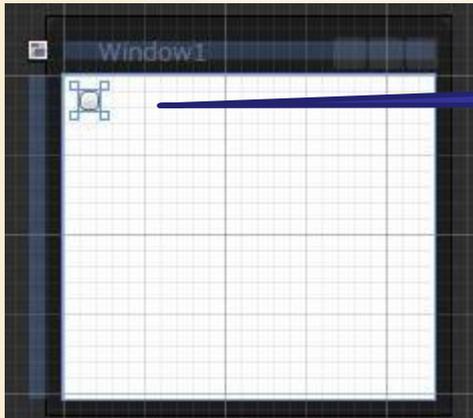
わんくま同盟 東京勉強会 #21

WPF アプリケーション アニメーション

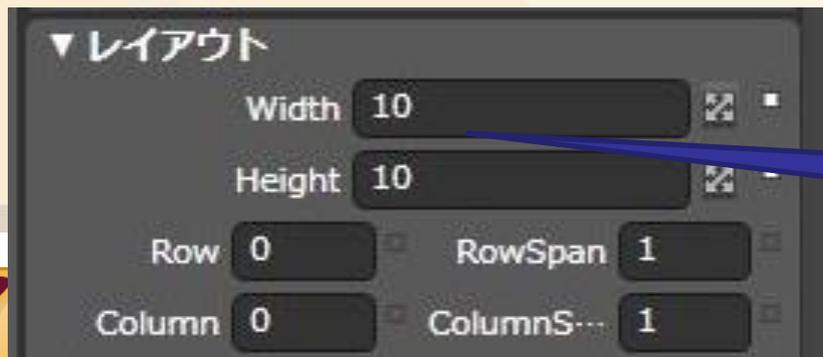
- UI のプロパティを時間に合わせて変更していけば、UI は簡単なアニメーションをしているように見えます。
- Expression Blend では、タイムライン上にキーフレームを作成しプロパティを変更することで、キーフレーム間の時間に指定されたプロパティ値を補間して、なめらかなアニメーションにしてくれます。
- アニメーションの開始や終了はトリガのイベントで指定します。

WPF アプリケーション アニメーション

- マウスをかざすと大きくなるボタンを作ってみましょう。
 - 新しいプロジェクトを作成しボタンを配置します。



ボタンを配置します。



10x10の大きさにします。

WPF アプリケーション アニメーション

- マウスがボタンの描画領域に入ったときに発生するイベントは MouseEnter です、トリガにボタンの MouseEnter イベントを作成します。



+ イベントをクリックして
イベントを追加します。

ボタンを選択します。

MouseEnter イベント
を選択します。

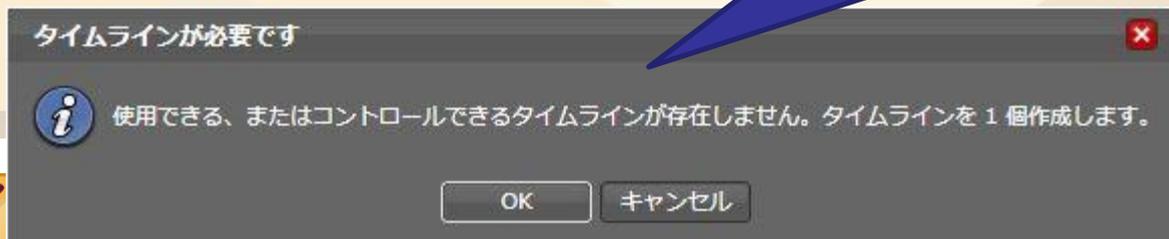
WPF アプリケーション アニメーション

- MouseEnter イベント発生時に実行するタイムラインを割り当てます。



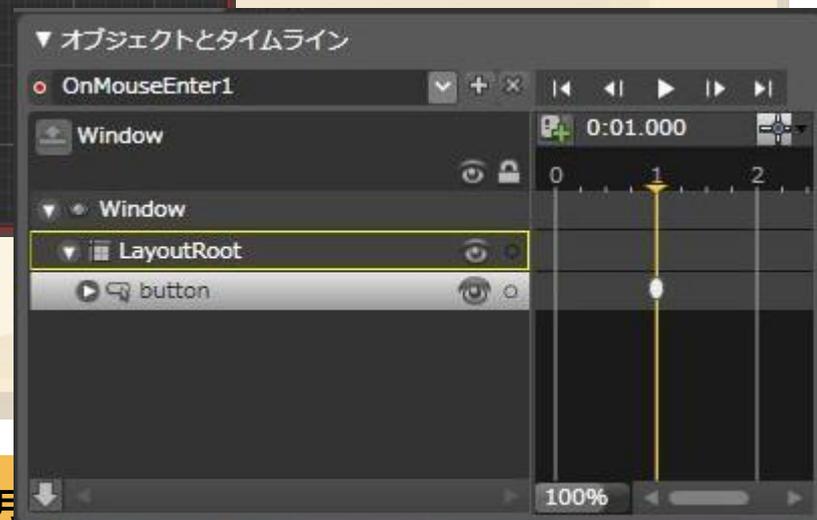
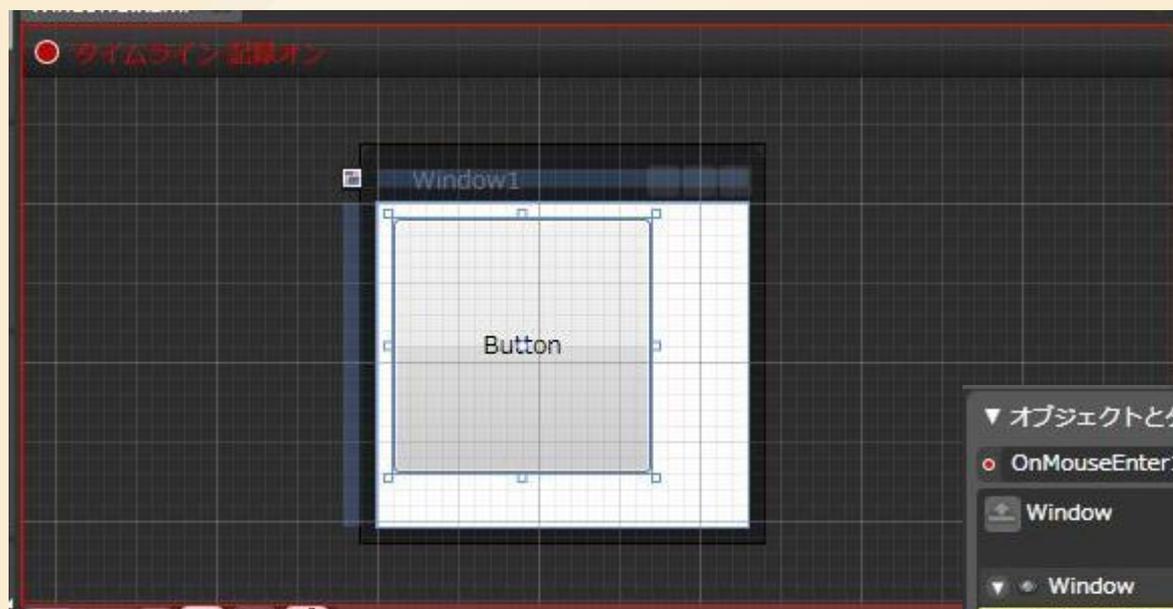
+をクリックしてタイムラインを割り当てます。

タイムラインが存在しないので新規で自動作成されます。



WPF アプリケーション アニメーション

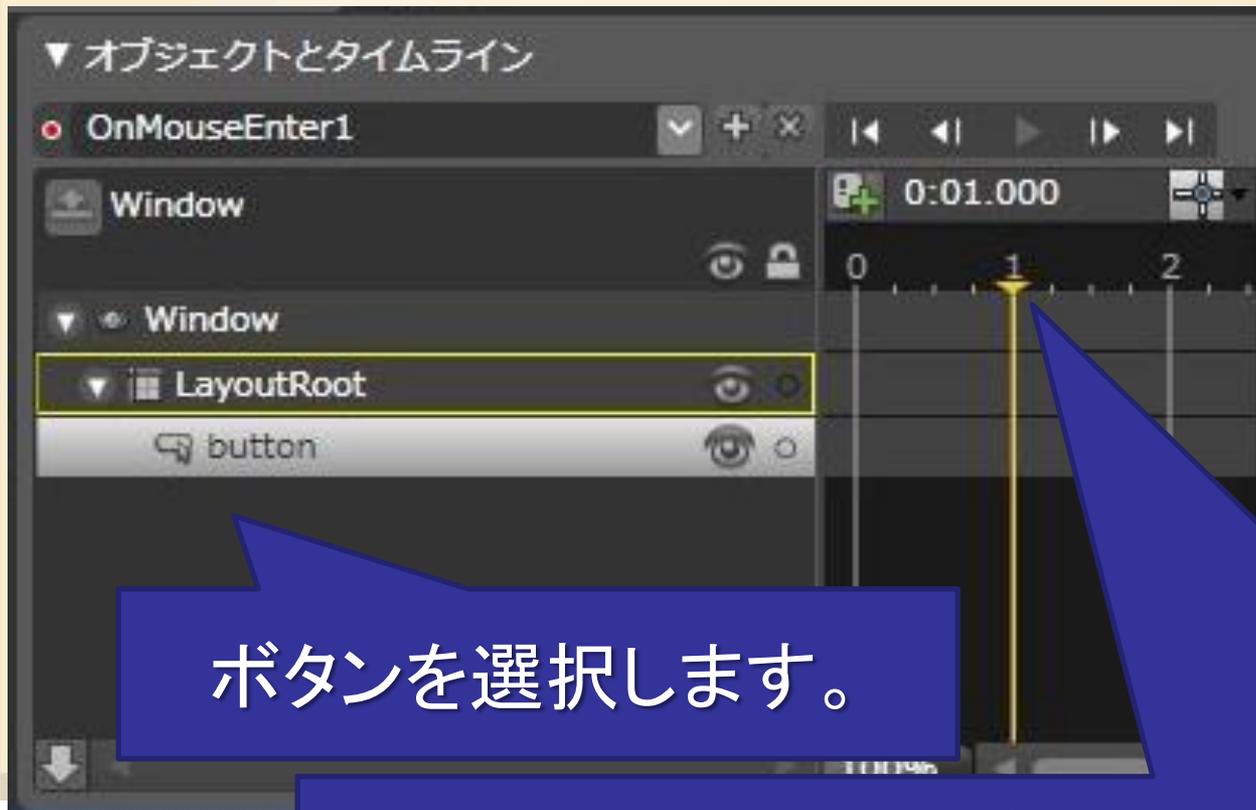
- タイムラインが表示され記録オンの状態になります。



注)このハードコピーは
作業中の画面です。

WPF アプリケーション アニメーション

- MouseEnter イベント発生で1秒後にボタンサイズを指定します。



ボタンを選択します。

タイムライン再生ヘッドを1秒の位置にします。



WPF アプリケーション アニメーション

- MouseEnter イベント発生で1秒後にボタンサイズを既定の10から128にします。

Width 128
Height 128

Row 0
Column 0

HorizontalAlign...
VerticalAlignment

Margin ← 8
↑ 8

▼ オブジェクトとタイムライン

OnMouseEnter1

Window

Window

LayoutRoot

button

0:01.000

0 1 2

100%

Width Height を 128 に
します。

キーフレーム(白丸)が作成されます。

同盟

WPF アプリケーション アニメーション

- MouseLeave イベント発生でタイムラインを停止しボタンの大きさを規定値(10x10)にします。



+ イベントをクリックして
イベントを追加します。

ボタンを選択します。

MouseLeave イベント
を選択します。

タイムラインを選択します。

タイムライン停止を選択します。

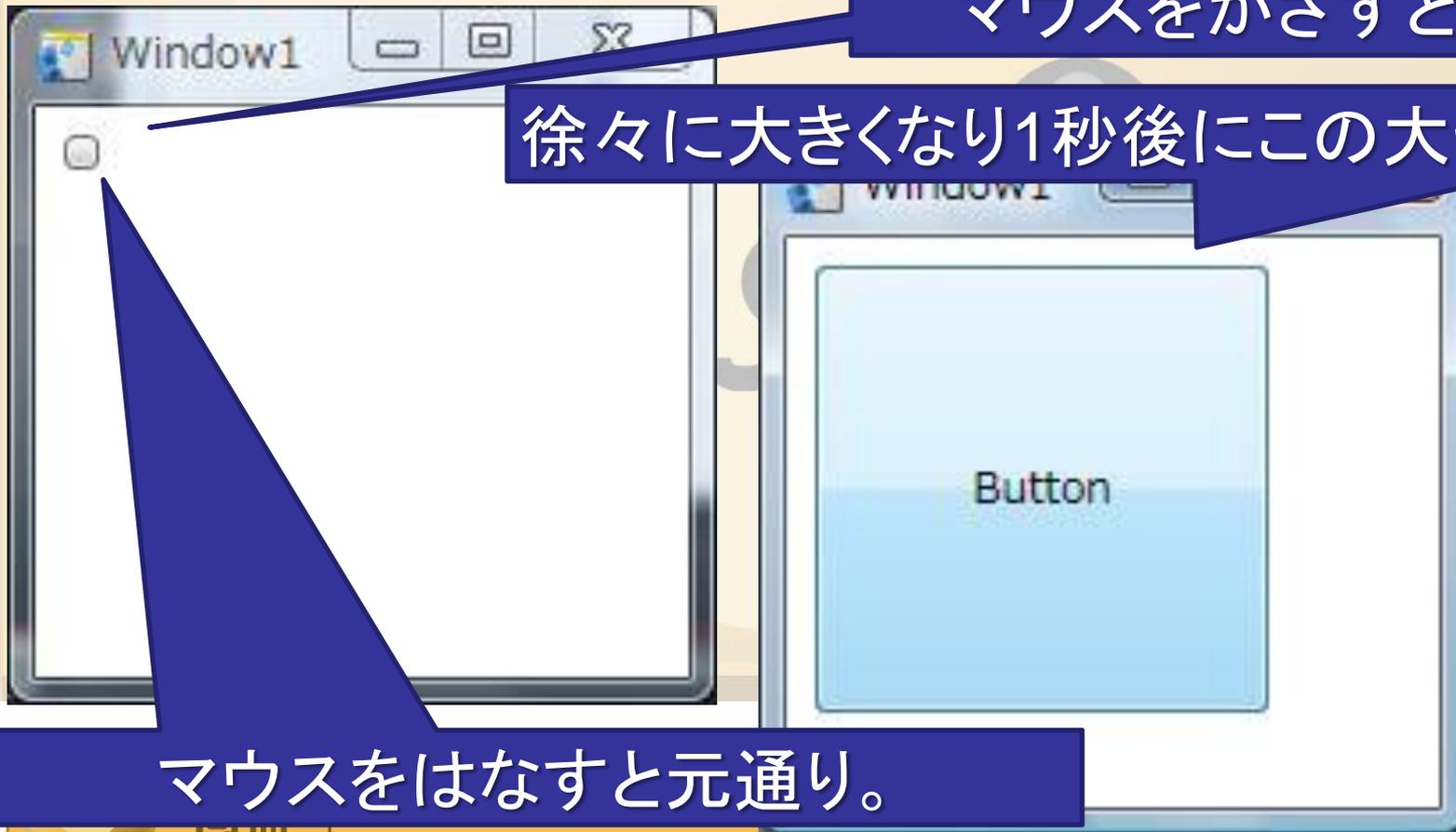
WPF アプリケーション アニメーション

- マウスをかざしたりはなしたりするとボタンの大きさが変わります。

マウスをかざすと。

徐々に大きくなり1秒後にこの大きさに。

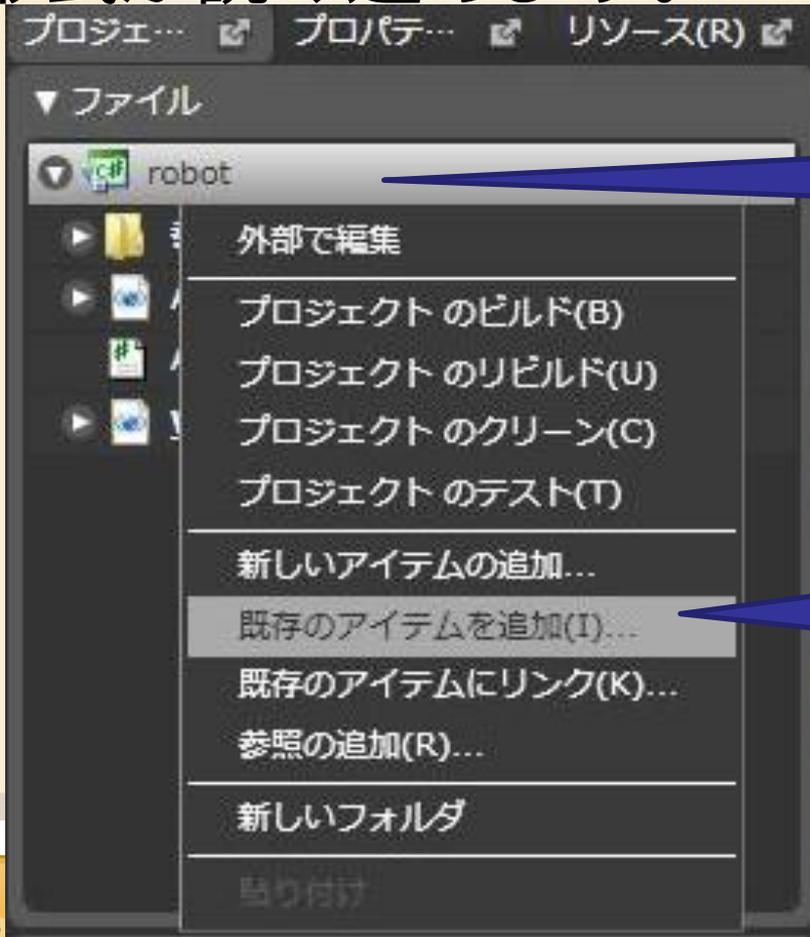
マウスをはなすと元通り。



WPF アプリケーション

3Dグラフィック

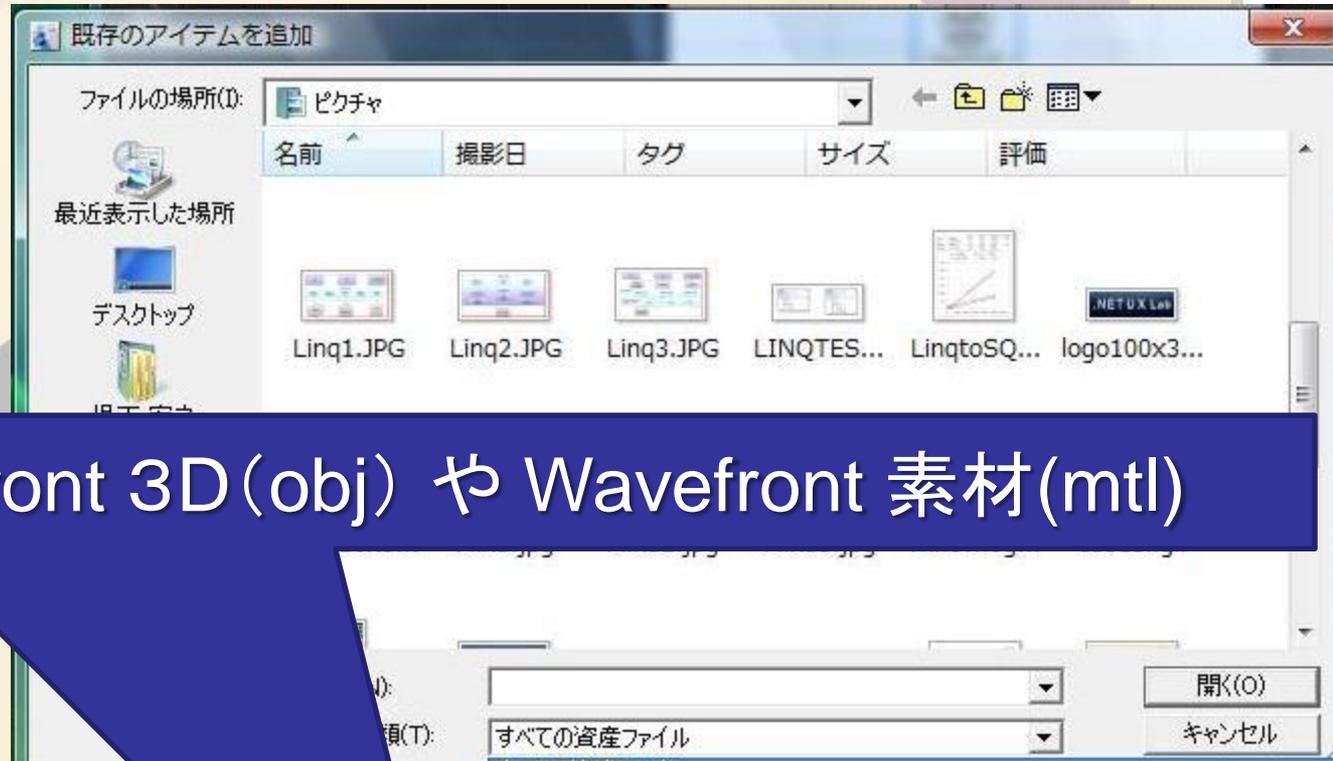
- 3DグラフィックはXAML形式や Wave front 3D形式が読み込めます。



プロジェクトを選択します。

既存のアイテムを追加を選択します。

WPF アプリケーション 3Dグラフィック



Wavefront 3D(obj) や Wavefront 素材(mtl)

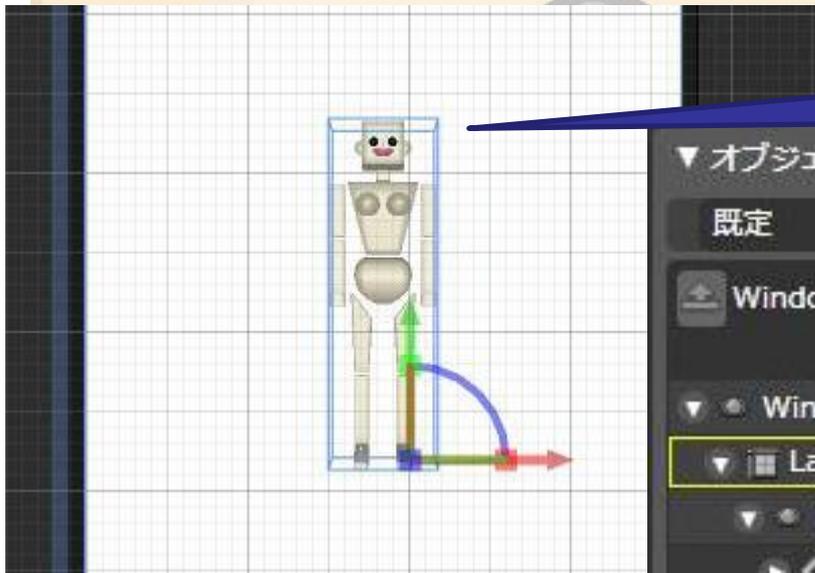


すべての資産ファイル

- すべての資産ファイル
- XML ドキュメント (*.xml)
- Visual C# (*.cs)
- Visual Basic (*.vb)
- ライセンス ファイル (*.licx)
- XAML (*.xaml)
- XAML リソース ディクショナリ (*.xaml)
- BMP (*.bmp)
- アイコン (*.ico)
- PNG (*.png)
- JPEG (*.jpg, *.jpeg)
- GIF (*.gif)
- TIFF (*.tif, *.tiff)
- オーディオとビデオ (*.dvr-ms, *.mid, *.rmi, *.midi, *.mpeg, *.mpg, *.m1v, *.mp2, *.mpa, *)
- Wavefront 3D (*.obj)
- Wavefront 素材の説明 (*.mtl)
- アプリケーション定義ファイル (*.xaml)
- すべてのファイル

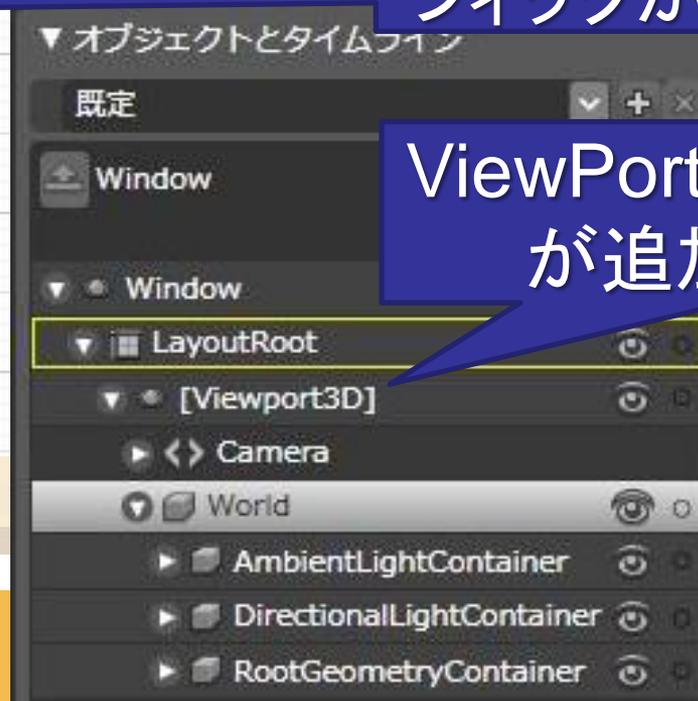
WPF アプリケーション 3Dグラフィック

- プロジェクトの obj ファイルをダブルクリックすると、ViewPort 3D・カメラ・環境光・指向性ライトが自動的に作成されます。



読み込まれた3Dグラフィックが表示されます。

ViewPort3D オブジェクトが追加されます。



DEMO

アニメーションと3Dグラフィック

えムナウ (児玉 宏之)

Microsoft MVP for **Development Tools Visual C#**

.Net ユーザーエクスペリエンス研究所

わんくま同盟



わんくま同盟 東京勉強会 #21

WPF アプリケーション

まとめ

- Expression Blend と Visual Studio のコラボレーションで XAML を書かないで WPF アプリケーションの開発が行えます。
- バインディングを使えばデータと UI を分離でき、表示のためのデータ変換や格納時のデータの検証を行えます。
- アニメーションを作成すれば Expression Blend で多種多様な視覚効果を追加できます。
- 3Dオブジェクトも XAML 形式や Wave front 3D 形式から読み込めアニメーションもできます。

Resources

.Net ユーザーエクスペリエンス研究所

<http://mnow.jp/>

わんくま同盟

<http://www.wankuma.com/>

INETA JAPAN

<http://www.ineta.jp/>

Microsoft Community Ring ホーム

<http://www.microsoft.com/japan/powerpro/community/>

Microsoft Expression

<http://www.microsoft.com/japan/products/expression/default.mspx>

Microsoft Visual Studio

<http://www.microsoft.com/japan/msdn/vstudio/>

Microsoft MSDN

<http://msdn2.microsoft.com/ja-jp/library/bb546194.aspx>

<http://msdn2.microsoft.com/ja-jp/library/ms754130.aspx>

<http://msdn2.microsoft.com/ja-jp/library/ms753307.aspx>

<http://msdn2.microsoft.com/ja-jp/library/ms771633.aspx>

<http://msdn2.microsoft.com/ja-jp/library/ms747122.aspx>

