Validation with WPF & Silverlight

グレープシティ株式会社
八巻 雄哉

本日の内容

本セッションは、赤間さんのセッションで
時間的な都合から紹介しきれなかった
WPFとSilverlightの検証機能を解説し、
グレープシティが現在開発中の
InputMan for WPFで提供する検証機能を
ご紹介するセッションです。

本セッションはそれぞれの最新バージョンである
WPF 3.5 SP1とSilverlight 3を前提にした内容となっています。

アジェンダ

- 検証の基本
- エラー表示の方法
- ValidationRuleを継承したカスタムクラス
- 検証ルールの実行タイミング
- インスタンス単位の検証
- InputMan for WPF CTP

Validation with WPF & Silverlight

検証の基本

検証の基本

- Binding.ValidationRulesプロパティ
  - ValidationRuleクラス
- Validation.Errors添付プロパティ
  - ValidationErrorクラス

検証はValidationRulesプロパティに設定されたValidationRuleに基づいて行われる。

検証エラーの場合には、Errors添付プロパティにValidationErrorが追加される。

添付プロパティ

実際には他のクラス（Canvas）で定義されているプロパティに、各クラス（Button）がそれぞれ別の値を指定できる。

```
<Canvas>
   <Button Canvas.Top="50" Canvas.Left="50"
         Content="ボタン"/>
</Canvas>
```

ValidationRule

- ルールは大きく分けると3つ

	WPF	Silverlight
ExceptionValidationRule（例外ベースの入力検証）	○	○
DataErrorValidationRule（IDataErrorInfo入力検証）	○	-
ValidationRuleを継承したカスタムクラス	○	-

ルールの設定

簡略記法

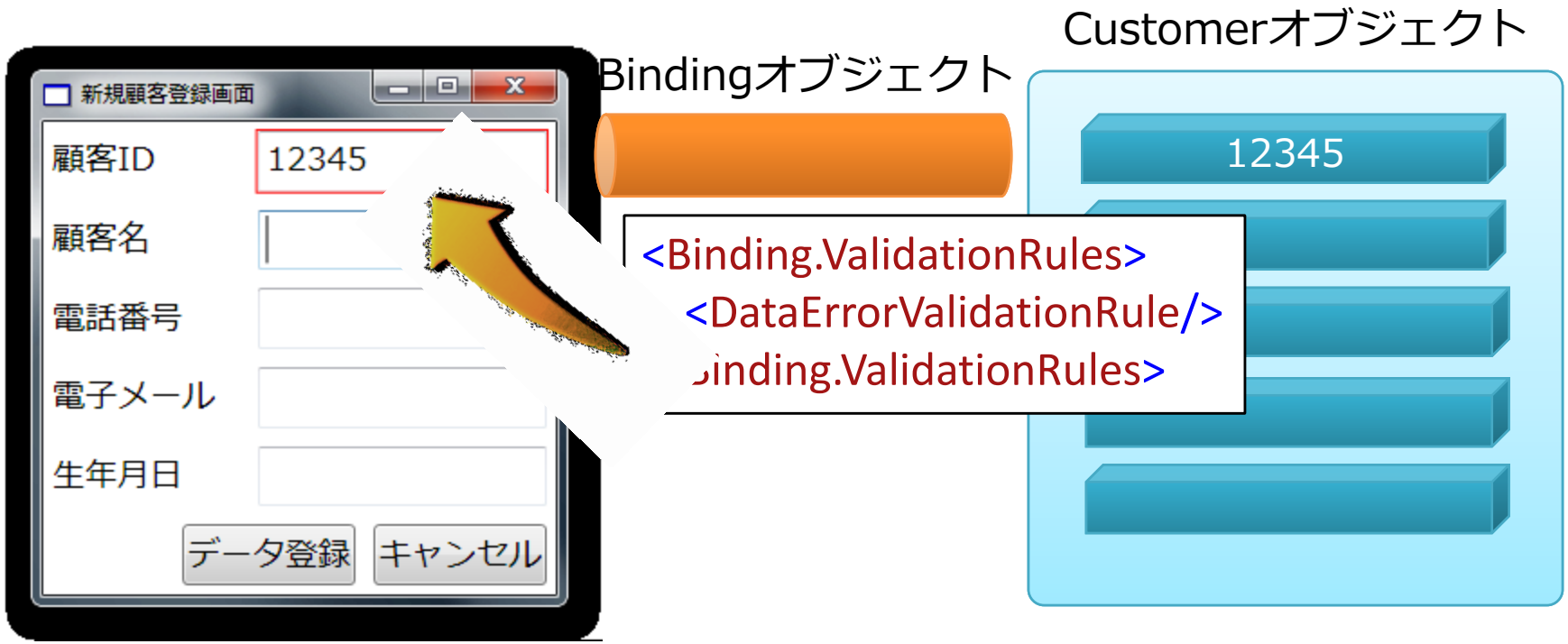
```
Text="{Binding Name, ValidatesOnDataErrors=True}"
```

=

```
<TextBox>
  <TextBox.Text>
    <Binding Path="Name">
      <Binding.ValidationRules>
        <DataErrorValidationRule/>
      </Binding.ValidationRules>
    </Binding>
  </TextBox.Text>
</TextBox>
```

ValidationErrorクラス

- ErrorContentプロパティ
  - エラーメッセージを取得します。
- Exceptionプロパティ
  - 検証エラーの原因となった例外を取得します。



TextBoxのValidation.Errors添付プロパティにValidationErrorが追加される。

Validation.Errors添付プロパティ

名前	値
Validation.GetErrors(TextBox_ID)	Count = 1
(0)	{System.Windows.Controls.ValidationError}
BindingInError	{System.Windows.Data.BindingExpression}
ErrorContent	"ID は半角英数大文字4文字です。" {String}
Exception	Nothing
RuleInError	{System.Windows.Controls.DataErrorValidationRule}

Validation with WPF & Silverlight

エラー表示の方法

エラー表示の方法

	WPF	Silverlight
ErrorTemplate 添付プロパティ	○	-
ValidationStates (VisualStateGroup)	- WPF 4で追加？	○
HasError 添付プロパティ	○	○
イベント	○ Validation.Error 添付イベント	○ FrameworkElement.BindingValidationErrorイベント

ErrorTemplate添付プロパティ

- Validation.ErrorTemplate（ControlTemplate型）

デフォルトのErrorTemplate

```
<Validation.ErrorTemplate>
  <ControlTemplate>
    <Border BorderBrush="#FFFF0000" BorderThickness="1,1,1,1">
      <AdornedElementPlaceholder />
    </Border>
  </ControlTemplate>
</Validation.ErrorTemplate>
```

- AdornedElementPlaceholderはErrorTemplateが適用される要素（今回の場合TextBox）を表す。

VisualStateManager（VSM）

- 動的な外観状態と、その外観状態に遷移する条件や遷移にかかる時間を管理する一連の機能を提供するのがVSM

ex
- ButtonコントロールのPressedというVisualStateは、ボタンが押された時の状態
- Pressedという名称で、背景色が赤に遷移するVisualStateを定義した場合、そのボタンは押された時だけ赤色に変化する。

@IT　WPF／Silverlight UIフレームワーク入門
第4回　“見た目”を決めるコントロール・テンプレート
http://www.atmarkit.co.jp/fdotnet/vblab/uiframework_04/uiframework_04_03.html

ValidationStates (VisualStateGroup)

VisualState	説明
Valid	コントロールが有効です。
InvalidFocused	フォーカスがある状態で、コントロールが無効です。
InvalidUnfocused	フォーカスがない状態で、コントロールが無効です。

HasError添付プロパティ

- Validation.HasError
  - ValidationErrorが存在する場合はTrue

```
<Style TargetType="TextBox">
  <Style.Triggers>
    <Trigger Property="Validation.HasError" Value="True">
      <Setter Property="ToolTip"
          Value="{Binding RelativeSource={RelativeSource Self},
                  Path=(Validation.Errors)[0].ErrorContent}"/>
      <Setter Property="Background" Value="Red"/>
    </Trigger>
  </Style.Triggers>
</Style>
```

イベント

- WPF
  - Validation.Error添付イベント
- Silverlight
  - FrameworkElement.BindingValidationErrorイベント
- ValidationErrorが追加／削除された際に発生するイベント
- Binding.NotifyOnValidationErrorプロパティをTrueに設定する必要あり。

ValidationErrorEventAction

- ValidationErrorEventArgs.Action プロパティ
  - ValidationErrorEventAction.Added
    新しいValidationErrorオブジェクトが検出された。
  - ValidationErrorEventAction.Removed
    既存のValidationErrorが削除された。

```
If e.Action = ValidationErrorEventAction.Added Then
    ‘検証エラーが発生した時の処理
ElseIf e.Action = ValidationErrorEventAction.Removed Then
    ‘検証エラーがなくなった時の処理
End If
```

Validation with WPF & Silverlight

ValidationRuleを継承したカスタムクラス

ValidationRuleを継承したカスタムクラス

- ValidationRuleクラスを継承し、ValidateメソッドをオーバーライドPublic Class IDValidation
    Inherits ValidationRule

    Public Overrides Function Validate(ByVal value As Object, ByVal cultureInfo As System.Globalization.CultureInfo) As System.Windows.Controls.ValidationResult

        Dim result As ValidationResult

        If value Is Nothing Then
            result = New ValidationResult(False, "IDは必須入力項目です。")
        ElseIf (Regex.IsMatch(value, "^[0-9A-Z]{4}$") = False) Then

Validation with WPF & Silverlight

検証ルールの実行タイミング

ValidationRule.ValidationStepプロパティ（WPFのみ）

ValidationStep 列挙値	説明	ExceptionValidationRule	DataErrorValidationRule	ValidationRuleを継承したカスタムクラス
RawProposedValue	変換が行われる前	デフォルト※1	不可	デフォルト
ConvertedProposedValue	変換が行われる前	可	不可	可
UpdatedValue	ソースの更新後	可※1	可	可※2
CommittedValue	値がソースにコミットされた後	可※1	デフォルト	可※2

※1　これらの設定値は意味をなさないと思われる。
※2　ValidationメソッドのvalueはBindingExpressionオブジェクト

Validation with WPF & Silverlight

インスタンス単位の検証

BindingGroupプロパティ（WPFのみ）

- 以下の条件のどちらかを満たしている場合、Bindingがグループ化される。
  - BindingのソースとBindingGroupを設定した要素のDataContextが同じ
  - BindingのBindingGroupNameプロパティがBindingGroupのNameと同じ

```
<Grid.BindingGroup>
  <BindingGroup>
    <BindingGroup.ValidationRules>
      <local:ObjectValidation/>
    </BindingGroup.ValidationRules>
  </BindingGroup>
</Grid.BindingGroup>
```

個々の検証をまとめて実行

- Bindingがグループ化されることにより、
  各Bindingは個別に更新されなくなる。
- 任意のタイミングで
  一気にすべての項目を更新させることが可能

- 個々の検証をまとめて実行できる。

個々ValidationRuleは、ValidationStepが
RawProposedValueであるもののみがBindingGroupの
ValidationRuleよりも前に実行される。

編集トランザクション

メソッド名	説明
BeginEdit	編集トランザクションを開始します。
CommitEdit	すべてのValidationRuleを実行し、 成功した場合はバインディングソースを更新します。
CancelEdit	集トランザクションを終了し、保留中の変更を破棄します。

編集トランザクション

```
Private Sub SubmitButton_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs)
    If LayoutRoot.BindingGroup.CommitEdit() Then
        MessageBox.Show("登録されました。")
        LayoutRoot.BindingGroup.BeginEdit()
    End If
End Sub

Private Sub CancelButton_Click(ByVal sender As System.Object, ByVal e As
System.Windows.RoutedEventArgs)
    LayoutRoot.BindingGroup.CancelEdit()
    LayoutRoot.BindingGroup.BeginEdit()
End Sub
```

Validation with WPF & Silverlight

InputMan for WPF CTP

標準でカバーできないケース

- WPF、Silverlightの検証は、Bindingを設定しているプロパティに対してのみ使用可能
- Windowsフォーム
  - Validatingイベント、Validatedイベント、CausesValidationプロパティ

InputMan for WPF CTP

- 現在グレープシティで開発中の「日本仕様の入力支援コンポーネント」InputManのWPF版
- 7種のコントロール
  - GcTextBox（テキスト）
  - GcMask（マスク）
  - GcDate（日付）
  - GcNumber（数値）
  - GcDropDownCalculator（カレン
  - GcDropDownCalendar（電卓）
  - GcValidationIndicator（検証インジケータ）

InputMan for WPF CTP

- 7種のコンバーター
  - BooleanToVisibilityConverter
  - DateFormatConverter
  - DateTimeDifferenceValueConverter
  - DecimalDifferenceValueConverter
  - MaskFormatConverter
  - NumberFormatConverter
  - ObjectToDateTimeConverter
- 2種のコンポーネント
  - GcImeManager（IMEマネージャ）
  - GcValidationManager（検証マネージャ）

GcValidationManagerはBinding使用の有無に関わらず、任意のプロパティ値を検証可能

InputManが提供する検証機能

機能	説明
検証	入力コントロールが、データソースに連結されている場合、されていない場合、いずれの場合も使用できる検証機能。検証の対象となるコントロールのプロパティ、検証タイミング、検証ルールを設定して使用。
検証ルール	WPF標準のValidationRuleクラスを継承する検証ルールを計10種類提供。入力されたデータの型、書式、値の範囲など、業務アプリケーションで使用頻度の高い検証ルールを用意。
検証アクション	不正なデータが入力されたときに実行するアクションを２種類提供。
インジケータ	不正なデータが入力されたときにアイコンを表示するコントロール。アイコンのツールチップにエラーの内容を設定。

Hello InputMan Validation

- GcValidationmanagerクラス
  - ValidationItem添付プロパティ

```
<TextBox x:Name="TextBox1"
    im:GcValidationManager.ValidationItem="{StaticResource ValidationItem_ID}"/>
<im:GcValidationIndicator ElementName="TextBox1"/>
```

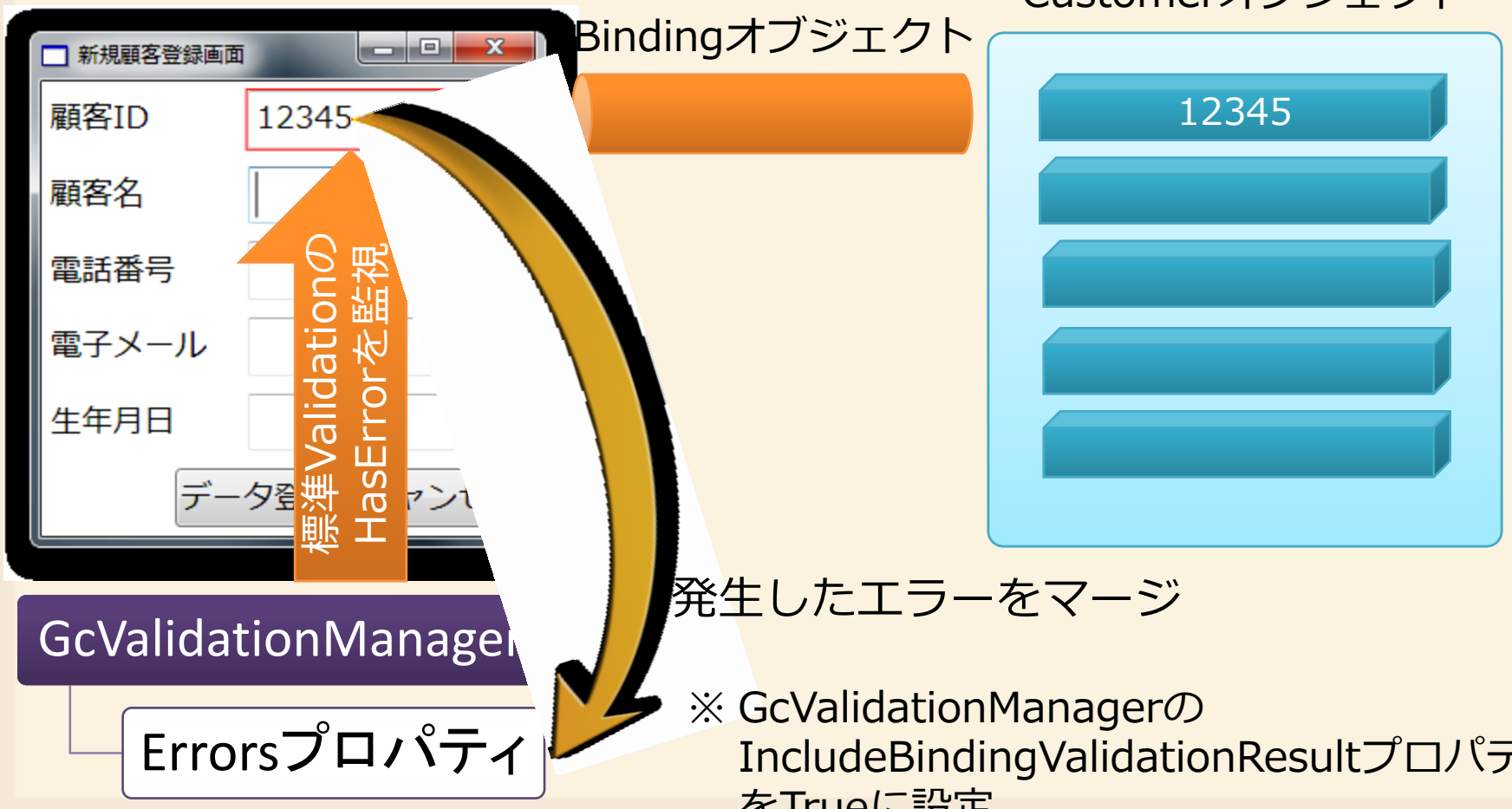
```
<Grid.Resources>
   <im:ValidationItem x:Key="ValidationItem_ID" x:Shared="False"
                      ValidatedProperty="Text">
      <im:ValidationItem.ValidationRules>
         <im:RegularExpressionRule Expression="^[0-9A-Z]{4}$"
                                   ErrorContent="ID は半角英数大文字4 文字です。" />
      </im:ValidationItem.ValidationRules>
   </im:ValidationItem>
</Grid.Resources>
```

InputManが提供する検証ルール

標準Validationでも使用可能なルールが10種類

検証ルール	説明
ValueRule	入力された値を指定した値と比較する検証
ValueRangeRule	入力された値が指定した範囲内にあるか検証
InvalidDateInputRule	入力された日付が日付として有効か検証 ※InputManの日付コントロール専用
RequiredFieldRule	入力された値が空でないことを検証
RegularExpressionRule	正規表現を使用して入力された値を検証
InvalidPairCharRule	カッコなどのペアとなる文字が両方存在するかどうかを検証
InvalidTypeRule	コントロールに入力された値が指定したデータ型に一致するかどうかを検証
SurrogateCharRule	サロゲートペア文字が入力されていないかどうかを検証
ExcludeListRule	文字列リストに設定した禁止文字列が入力されていないかどうかを検証
IncludeListRule	文字列リストに設定した文字列のいずれかが入力されているかどうか検証

標準Validationとの連携



InputMan for WPF CTPプログラム

- 評価していただける方に、
  InputMan for WPFのCTP版を提供します。
- CTP版には製品のほかに、
  以下のものが含まれています。
  - 評価ガイド（PDFファイル）
  - サンプルプロジェクト
  - リリースノート
  - リファレンス（英語）
- 評価期間は11月末日までを予定

CTPプログラム参加方法

InputManCTP@grapecity.com
宛にメールを送信ください。

追って詳細をご連絡させていただきます。

たくさんのご参加お待ちしております。


GrapeCity®