Data Binding

Validação de Dados

Validação de dados

Utiliza-se *bindings* bidirecionais para garantir que eventuais alterações sejam propagadas para o objeto associado. No entanto, em muitos casos é necessário efetuar validações antes de os valores serem "copiados" do controle do UI para a propriedade do objeto.

A propagação automática de alterações depende da implementação da interface *INotifyPropertyChanged*. Em outras palavras, o nosso tipo Pessoa usado no exemplo a seguir tem de implementar esta interface para que o objeto de *Binding* consiga propagar uma eventual alteração na propriedade usada na operação de *binding*.

Validação de dados

```
public class Pessoa
    {
        private string _nome;

        public string Nome
        {
            get { return _nome; }
            set { _nome = value; }
        }
        private string _cidade;

        public string Cidade
        {
            get { return _cidade; }
            set { _cidade = value; }
        }
    }
```

Validação de dados

```
...
<TextBox Grid.Column="1"
         Width="150"
         Height="30"
         HorizontalAlignment="Left"
         Margin="5"
         Text="{Binding Nome,Mode=TwoWay}" >
...
```

Apesar do Mode=TwoWay, o objeto não está sendo notificado automaticamente, por que falta implementar a interface INotifyPropertyChaned.

Validação de dados

```
public class Pessoa: INotifyPropertyChanged
   {
            ...
            public event PropertyChangedEventHandler PropertyChanged;

        private void NotificaAlteracaoPropriedade(String propName)
        {
            if (PropertyChanged != null)
            {
                PropertyChanged(this, new
PropertyChangedEventArgs(propName));
            }
        }
   }
```

Validação de dados

Modifique as propriedades do objeto para o seguinte:

```
...
public string Nome
        {
            get { return _nome; }
            set
            {
                if (_nome == value) return;
                _nome = value;
                NotificaAlteracaoPropriedade("Nome");

            }
        }
...
```

Mas ainda não está validando a entrada de dados.

Validação de dados

A plataforma detecta erros sempre que a *attached property Validation.Errors* de um objeto *Binding* contiver erros. Existem vários casos que podem levar um erro a ser adicionado a esta coleção.

- ✓Exceções geradas a partir do conversor utilizado ou do *setter* da propriedade a ser modificada.
- ✓Exceções geradas a partir de um atributo de validação aplicado a propriedade do objeto usada no *binding*.
- ✓Quando o objeto associado ao binding implementa a interface IDataErrorInfo e a propriedade Item de um objeto devolve um valor diferente de null.
- ✓Quando o objeto de binding implementa a interface ***INotifyDataErrorInfo*** e o método GetErrors devolve um valor não nulo.

Validação de dados

Um objeto de *binding* disponibiliza várias propriedades que controlam o tipo de validação efetuado e a forma como é fornecido *feedback* ao usuário.

- Quando a propriedade *ValidatesOnExceptions* possui o valor *true*, todas as exceções geradas durante a atribuição do valor à propriedade são apresentadas graficamente ao usuário.
- Se atribuirmos o valor *true* à propriedade *ValidatesOnDataErrors*, então o objeto *binding* apresenta os erros obtidos a partir da interface *IDataErrorInfo*.
- Finalmente, a propriedade *ValidatesOnNotifyDataError* decide se o objeto de *binding* deve apresentar os erros reportados por objetos que implementam a interface *INotifyDataErrorInfo*.

Validação de dados

```
<TextBox Grid.Column="1"
                Width="150"
                Height="30"
                HorizontalAlignment="Left"
                Margin="5"
                Text="{Binding Nome, Mode=TwoWay,
NotifyOnValidationError=True,
                        ValidatesOnExceptions=True }" >
```

Validação de dados

Utilização de atributos de validação

O *namespace System.ComponentModel.DataAnnotations* contém várias classes (derivadas de *ValidationAttribute*) que podem ser usadas para anotar as propriedades de um objeto usado numa operação de data *binding*. Estas classes permitem definir restrições que têm de ser respeitadas pelos valores atribuídos às propriedades.

Validação de dados

```
private string _nome;
        [Required(AllowEmptyStrings=false,
            ErrorMessage="o nome não pode ser uma string vazia")]
        public string Nome
        {
            get { return _nome; }
            set
            {
                if (_nome == value) return;
                ValidaPropriedade(value, "Nome");
                _nome = value;
                NotificaAlteracaoPropriedade("Nome");

            }
```

Validação de dados

```
private void ValidaPropriedade(Object valor, String nomeProp)
        {
            var context = new ValidationContext(this, null, null)
            {
                MemberName = nomeProp
            };
            Validator.ValidateProperty(valor, context);
        }
```

Validação de dados

Para que tudo funcione, temos apenas de “ativar” as propriedades *NotifyOnValidationError* e *ValidatesOnException* na declaração do *binding* em *XAML*.

```
Text="{Binding Nome,Mode=TwoWay,NotifyOnValidationError=True,
                    ValidatesOnExceptions=True }" >
```