

Zeus TextBoxControls Ver. 1.1

Manual

Copyright © 2018-2020 Χρήστος Μουρατίδης

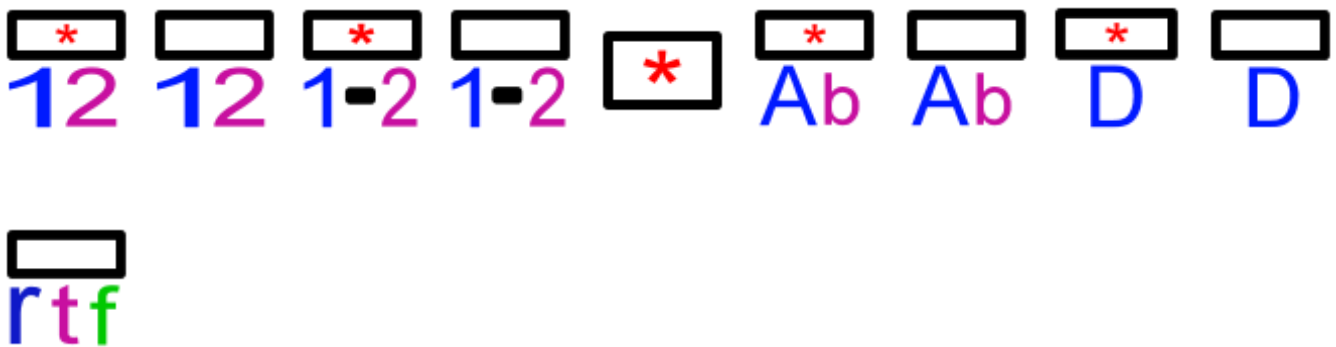
Πίνακας περιεχομένων

ZEUS TEXTBOXCONTROLS VERSION 1.1	1
ΟΙ ΟΡΙΣΜΟΙ ΤΩΝ ΚΛΑΣΕΩΝ	6
ΔΙΑΝΟΜΗ	19
ΕΠΙΚΟΙΝΩΝΙΑ.....	20
ΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ	21
BYTETEXTBOX	22
ΙΔΙΟΤΗΤΕΣ.....	25
<i>ErrorMessageOnIncorrectValueType</i>	26
SHORTTEXTBOX	27
ΙΔΙΟΤΗΤΕΣ.....	30
<i>ErrorMessageOnIncorrectValueType</i>	31
INTEGERTEXTBOX	32
ΙΔΙΟΤΗΤΕΣ.....	35
<i>ErrorMessageOnIncorrectValueType</i>	36
LONGTEXTBOX	37
ΙΔΙΟΤΗΤΕΣ.....	40
<i>ErrorMessageOnIncorrectValueType</i>	41
SINGLETEXTBOX	42
ΙΔΙΟΤΗΤΕΣ.....	45
<i>ErrorMessageOnIncorrectValueType</i>	46
DOUBLETEXTBOX	47
ΙΔΙΟΤΗΤΕΣ.....	50
<i>ErrorMessageOnIncorrectValueType</i>	51
DECIMALTEXTBOX	52
ΙΔΙΟΤΗΤΕΣ.....	55
<i>ErrorMessageOnIncorrectValueType</i>	56
NULLABLE ΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ	57
NULLABLEBYTETEXTBOX	58
ΙΔΙΟΤΗΤΕΣ.....	61
<i>ErrorMessageOnIncorrectValueType</i>	62
NULLABLESHORTTEXTBOX	63
ΙΔΙΟΤΗΤΕΣ.....	66
<i>ErrorMessageOnIncorrectValueType</i>	67
NULLABLEINTEGERTEXTBOX	68
ΙΔΙΟΤΗΤΕΣ.....	71
<i>ErrorMessageOnIncorrectValueType</i>	72
NULLABLELONGTEXTBOX	73
ΙΔΙΟΤΗΤΕΣ.....	76

<i>ErrorMessageOnIncorrectValueType</i>	77
NULLABLESINGLETEXTBOX	78
ΙΔΙΟΤΗΤΕΣ	81
<i>ErrorMessageOnIncorrectValueType</i>	82
NULLABLEDOUBLETEXTBOX	83
ΙΔΙΟΤΗΤΕΣ	86
<i>ErrorMessageOnIncorrectValueType</i>	87
NULLABLEDECIMALTEXTBOX	88
ΙΔΙΟΤΗΤΕΣ	91
<i>ErrorMessageOnIncorrectValueType</i>	92
ΓΕΝΙΚΟΙ ΑΡΙΘΜΗΤΙΚΟΙ ΤΥΠΟΙ	93
NUMERICTEXTBOX	94
ΙΔΙΟΤΗΤΕΣ	97
<i>ErrorMessageOnIncorrectValueType</i>	98
<i>NumericType</i>	99
NULLABLENUMERICTEXTBOX	100
ΙΔΙΟΤΗΤΕΣ	103
<i>ErrorMessageOnIncorrectValueType</i>	104
<i>NumericType</i>	105
NUMERICRANGETEXTBOX	106
ΙΔΙΟΤΗΤΕΣ	109
<i>ErrorMessageOnIncorrectValueType</i>	110
<i>ErrorMessageOnValueOutOfRange</i>	111
<i>NumericType</i>	112
<i>Range</i>	113
NULLABLENUMERICRANGETEXTBOX	115
ΙΔΙΟΤΗΤΕΣ	118
<i>ErrorMessageOnIncorrectValueType</i>	119
<i>ErrorMessageOnValueOutOfRange</i>	120
<i>NumericType</i>	121
<i>Range</i>	122
TEXTBOXES ΓΙΑ ΑΛΦΑΡΙΘΜΗΤΙΚΑ ΚΑΙ ΑΛΦΑΒΗΤΙΚΑ ΔΕΔΟΜΕΝΑ	124
REQUIREDTEXTBOX	125
ΙΔΙΟΤΗΤΕΣ	128
<i>ErrorMessageOnEmpty</i>	129
ALPHABETICTEXTBOX	130
ΙΔΙΟΤΗΤΕΣ	134
<i>ErrorMessageOnIncorrectValue</i>	135
<i>AlphabeticCharacters</i>	136
<i>ExtraCharacters</i>	138
REQUIREDALPHABETICTEXTBOX	140
ΙΔΙΟΤΗΤΕΣ	144
<i>ErrorMessageOnEmpty</i>	145

<i>ErrorMessageOnIncorrectValue</i>	146
<i>AlphabeticCharacters</i>	147
<i>ExtraCharacters</i>	149
TEXTBOXES ΓΙΑ ΗΜΕΡΟΜΗΝΙΕΣ	151
DATETEXTBOX.....	152
ΙΔΙΟΤΗΤΕΣ.....	155
<i>ErrorMessageOnIncorrectValueType</i>	156
NULLABLEDATETEXTBOX.....	157
ΙΔΙΟΤΗΤΕΣ.....	160
<i>ErrorMessageOnIncorrectValueType</i>	161
ZRICHTEXTBOX.....	162
ΙΔΙΟΤΗΤΕΣ.....	164
<i>DocumentXamlText</i>	165
<i>EmptyDocumentXamlText</i>	167
<i>IsDocumentEmpty</i>	168
ΜΕΘΟΔΟΙ.....	169
<i>ClearDocument()</i>	171
<i>CopyDocument()</i>	172
<i>GetDocumentAsXamlText()</i>	173
<i>GetPlainText()</i>	174
<i>SetDocument(newDocument)</i>	175
<i>SetDocumentFromXamlText(flowDocumentXamlText)</i>	177
ΠΑΡΑΡΤΗΜΑ: Η ΚΛΑΣΗ CUSTOMER	178

Zeus TextBoxControls version 1.1



Κλάσεις: ByteTextBox, NullableByteTextBox, ShortTextBox, NullableShortTextBox, IntegerTextBox, NullableIntegerTextBox, LongTextBox, NullableLongTextBox, SingleTextBox, NullableSingleTextBox, DoubleTextBox, NullableDoubleTextBox, DecimalTextBox, NullableDecimalTextBox, NumericTextBox, NullableNumericTextBox, NumericRangeTextBox, NullableNumericRangeTextBox, RequiredTextBox, AlphabeticTextBox, RequiredAlphabeticTextBox, DateTextBox, NullabelDateTextBox, ZRichTextBox.

Inherits: System.Windows.Controls.TextBox,
System.Windows.Controls.RichTextBox (για το ZRichTextBox)

Namespace: Zeus.WPF.Controls.TextBoxControls

Assembly: ZeusTextBoxControls (in ZeusTextBoxControls.dll)


Dependencies: Zeus.WPF.Classes.Validations (in ZeusValidations.dll)





Περιγραφή








Η βιβλιοθήκη **Zeus TextBoxControls** περιλαμβάνει ένα σύνολο από **εξειδικευμένα TextBoxes**, κυρίως για είσοδο αριθμητικών δεδομένων συγκεκριμένου τύπου (π.χ. Byte, Short, Integer κλπ) αλλά περιέχει και **TextBoxes** για είσοδο μόνο αλφαβητικών χαρακτήρων καθώς και για είσοδο ημερομηνιών. Ο έλεγχος των εισαγόμενων δεδομένων πραγματοποιείται μέσω **εξειδικευμένων ValidationRules** που έχουν ενσωματωθεί κατά περίπτωση. Τα **ValidationRules** βρίσκονται στο **assembly ZeusValidations.dll**.

Παρακάτω, παρατίθενται τα **TextBoxControls** ανά τύπο δεδομένων.



Για **Αριθμητικά δεδομένα:**

	ByteTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Byte . Η τιμή απαιτείται (required).
---	-----------------------------	--



	NullableByteTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Byte . Η τιμή δεν απαιτείται (can be null).
	ShortTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Short . Η τιμή απαιτείται (required).
	NullableShortTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Short . Η τιμή δεν απαιτείται (can be null).
	IntegerTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Integer . Η τιμή απαιτείται (required).
	NullableIntegerTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Integer . Η τιμή δεν απαιτείται (can be null).
	LongTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Long . Η τιμή απαιτείται (required).
	NullableLongTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Long . Η τιμή δεν απαιτείται (can be null).
	SingleTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Single . Η τιμή απαιτείται (required).
	NullableSingleTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Single . Η τιμή δεν απαιτείται (can be null).
	DoubleTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Double . Η τιμή απαιτείται (required).

	NullableDoubleTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Double . Η τιμή δεν απαιτείται (can be null).
	DecimalTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Decimal . Η τιμή απαιτείται (required).
	NullableDecimalTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου Decimal . Η τιμή δεν απαιτείται (can be null).
	NumericTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου . Ο τύπος καθορίζεται από την ιδιότητα NumericType . Αποτελεί ένα γενικό TextBox για την είσοδο αριθμητικών δεδομένων και μπορεί να χρησιμοποιηθεί ισοδύναμα αντί των παραπάνω εξειδικευμένων, ανά τύπο, TextBoxes. Η τιμή απαιτείται (required).
	NullableNumericTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου . Ο τύπος καθορίζεται από την ιδιότητα NumericType . Αποτελεί ένα γενικό TextBox για την είσοδο αριθμητικών δεδομένων και μπορεί να χρησιμοποιηθεί ισοδύναμα αντί των παραπάνω εξειδικευμένων, ανά τύπο, TextBoxes. Η τιμή δεν απαιτείται (can be null).
	NumericRangeTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου και για συγκεκριμένο εύρος αποδεκτών τιμών . Ο τύπος καθορίζεται από την ιδιότητα NumericType και το εύρος των αποδεκτών τιμών καθορίζεται από την ιδιότητα Range . Η τιμή απαιτείται (required).
	NullableNumericRangeTextBox	Επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου και για συγκεκριμένο εύρος αποδεκτών τιμών . Ο τύπος καθορίζεται από την ιδιότητα NumericType και το εύρος των αποδεκτών τιμών καθορίζεται από την ιδιότητα Range . Η τιμή δεν απαιτείται (can be null).


Για **Αλφαριθμητικά δεδομένα:**

	RequiredTextBox	Απαιτεί την είσοδο δεδομένου στο TextBox.
	AlphabeticTextBox	Επιτρέπει την είσοδο μόνο αλφαβητικών δεδομένων . Οι αλφαβητικοί χαρακτήρες καθορίζονται στην ιδιότητα AlphabeticCharacters . Εξ' ορισμού αποδεκτοί είναι οι χαρακτήρες του Αγγλικού αλφαβήτου αλλά μπορούμε, μέσω αυτής της ιδιότητας, να προσθέσουμε και άλλους. Επιπλέον, στην ιδιότητα ExtraCharacters μπορούμε να καθορίσουμε κι άλλους αποδεκτούς χαρακτήρες που σημασιολογικά δεν είναι αλφαβητικοί (όπως το κενό, το θαυμαστικό, η τελεία, το κόμμα κλπ). Η τιμή δεν απαιτείται (can be empty).
	RequiredAlphabeticTextBox	Επιτρέπει την είσοδο μόνο αλφαβητικών δεδομένων . Οι αλφαβητικοί χαρακτήρες καθορίζονται στην ιδιότητα AlphabeticCharacters . Εξ' ορισμού αποδεκτοί είναι οι χαρακτήρες του Αγγλικού αλφαβήτου αλλά μπορούμε, μέσω αυτής της ιδιότητας, να προσθέσουμε και άλλους. Επιπλέον, στην ιδιότητα ExtraCharacters μπορούμε να καθορίσουμε κι άλλους αποδεκτούς χαρακτήρες που σημασιολογικά δεν είναι αλφαβητικοί (όπως το κενό, το θαυμαστικό, η τελεία, το κόμμα κλπ). Η τιμή απαιτείται (required).

Για **Ημερομηνιακά δεδομένα:**

	DateTextBox	Επιτρέπει την είσοδο μόνο ημερομηνιακών δεδομένων, τύπου Date . Η τιμή απαιτείται (required).
	NullableDateTextBox	Επιτρέπει την είσοδο μόνο ημερομηνιακών δεδομένων, τύπου Date . Η τιμή δεν απαιτείται (can be null).

Για **Εμπλουτισμένο κείμενο (rtf)**:

	ZRichTextBox	Επιτρέπει την επεξεργασία εμπλουτισμένου κειμένου (rtf) . Πρόκειται για ένα κανονικό RichTextBox με τη διαφορά ότι εμμέσως μπορούμε να κάνουμε bind την ιδιότητα Document .
---	------------------------------	---

Σημείωση: Στα παραδείγματα κώδικα γίνεται χρήση της **κλάσης Customer**. Θα τη βρείτε στο [Παράρτημα: Η κλάση Customer](#).

Οι ορισμοί των κλάσεων

Οι κλάσεις έχουν οριστεί ως εξής:

- Για την κλάση **ByteTextBox**:

Σύνταξη:

VB:

```
Public Class ByteTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:ByteTextBox ... />
```

- Για την κλάση **NullableByteTextBox**:

Σύνταξη:

VB:

```
Public Class NullableByteTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NullableByteTextBox ... />
```

- Για την κλάση **ShortTextBox**:

Σύνταξη:

VB:

```
Public Class ShortTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:ShortTextBox ... />
```

- Για την κλάση **NullableShortTextBox**:

Σύνταξη:

VB:

```
Public Class NullableShortTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableShortTextBox ... />
```

- Για την κλάση **IntegerTextBox**:

Σύνταξη:

VB:

```
Public Class IntegerTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:IntegerTextBox ... />
```

- Για την κλάση **NullableIntegerTextBox**:

Σύνταξη:

VB:

```
Public Class NullableIntegerTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableIntegerTextBox ... />
```

- Για την κλάση **LongTextBox**:

Σύνταξη:

VB:

```
Public Class LongTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:LongTextBox ... />
```

- Για την κλάση **NullableLongTextBox**:

Σύνταξη:

VB:

```
Public Class NullableLongTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableLongTextBox ... />
```

- Για την κλάση **SingleTextBox**:

Σύνταξη:

VB:

```
Public Class SingleTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:SingleTextBox ... />
```

- Για την κλάση **NullableSingleTextBox**:

Σύνταξη:

VB:

```
Public Class NullableSingleTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableSingleTextBox ... />
```

- Για την κλάση **DoubleTextBox**:

Σύνταξη:

VB:

```
Public Class DoubleTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:DoubleTextBox ... />
```

- Για την κλάση **NullableDoubleTextBox**:

Σύνταξη:

VB:

```
Public Class NullableDoubleTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableDoubleTextBox ... />
```

- Για την κλάση **DecimalTextBox**:

Σύνταξη:

VB:

```
Public Class DecimalTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:DecimalTextBox ... />
```

- Για την κλάση **NullableDecimalTextBox**:

Σύνταξη:

VB:

```
Public Class NullableDecimalTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableDecimalTextBox ... />
```

- Για την κλάση **NumericTextBox**:

Σύνταξη:

VB:

```
Public Class NumericTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NumericTextBox NumericType="IntegerType"... />
```

- Για την κλάση **NullableNumericTextBox**:

Σύνταξη:

VB:

```
Public Class NullableNumericTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableNumericTextBox NumericType="IntegerType" ... />
```

- Για την κλάση **NumericRangeTextBox**:

Σύνταξη:

VB:

```
Public Class NumericRangeTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NumericRangeTextBox NumericType="IntegerType" Range="10:20;50:100" ... />
```

- Για την κλάση **NullableNumericRangeTextBox**:

Σύνταξη:

VB:

```
Public Class NullableNumericRangeTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableNumericRangeTextBox NumericType="DoubleType" Range="10:20;30,5:40,5"  
    Language="el" ... />
```

- Για την κλάση **RequiredTextBox**:

Σύνταξη:

VB:

```
Public Class RequiredTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:RequiredTextBox ... />
```

- Για την κλάση **AlphabeticTextBox**:

Σύνταξη:

VB:

```
Public Class AlphabeticTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:AlphabeticTextBox AlphabeticCharacters="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnop  
qrstuvwxyz"  
    ExtraCharacters="!;,:~?()&- ' " ... />
```

- Για την κλάση **RequiredAlphabeticTextBox**:

Σύνταξη:

VB:

```
Public Class RequiredAlphabetic TextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:RequiredAlphabeticTextBox AlphabeticCharacters="ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδε
ζηθικλμνξοπρστυφχψωσάέήίόύϊϋ"
    ExtraCharacters=".!;, :?()&-' " ... />
```

-
- Για την κλάση **DateTextBox**:

Σύνταξη:

VB:

```
Public Class DateTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:DateTextBox ... />
```

- Για την κλάση **NullableDateTextBox**:

Σύνταξη:

VB:

```
Public Class NullableDateTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableDateTextBox ... />
```

- Για την κλάση **ZRichTextBox**:

Σύνταξη:

VB:

```
Public Class ZRichTextBox  
    Inherits RichTextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:ZRichTextBox ... />
```

Τί νέο υπάρχει στην έκδοση 1.1



- Έχει προστεθεί το [ZRichTextBox](#) control.

Διανομή

Κατά τη διανομή, στο φάκελο της εφαρμογής σας πρέπει να αντιγράψετε τα **assembly αρχεία** **ZeusValidations.dll** και **ZeusTextBoxControls.dll**.

Επικοινωνία

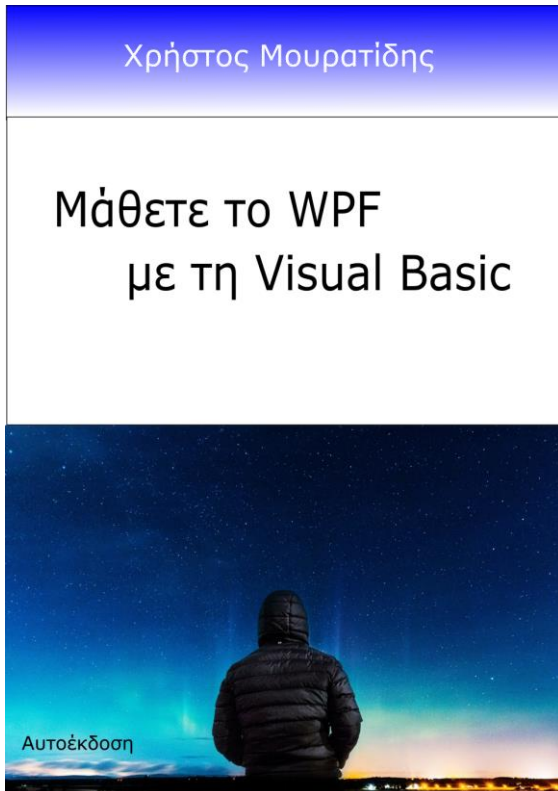
Για οποιαδήποτε πληροφορία ή διευκρίνιση παρακαλώ επικοινωνήστε στο :

mouratx@yahoo.com ή mouratx@hotmail.com



Χρήστος Μουρατίδης,
Πειραιάς, Σεπτέμβριος 2018, Ιούνιος 2020 (v1.1)

Υ.Γ. Μπορείτε να επικοινωνήσετε μαζί μου για να προμηθευτείτε το **βιβλίο** μου
"Μάθετε το WPF με τη Visual Basic" (1.333 σελίδες, Αυτοέκδοση 2018).





Αριθμητικοί τύποι

Στα επόμενα, παρουσιάζονται τα **TextBoxes** για τους βασικούς **αριθμητικούς τύπους** (**Byte, Short, Integer, Long, Single, Double** και **Decimal**). Η είσοδος του αριθμού είναι υποχρεωτική.

- [ByteTextBox](#)
- [ShortTextBox](#)
- [IntegerTextBox](#)
- [LongTextBox](#)
- [SingleTextBox](#)
- [DoubleTextBox](#)
- [DecimalTextBox](#)



ByteTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Byte**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class ByteTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:ByteTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **ByteTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Byte**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>

    <!-- The Validation Error ControlTemplate -->
    <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

            <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
                <AdornedElementPlaceholder x:Name="placeholder"/>
            </Border>

            <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
                <TextBlock.ToolTip >
                    <TextBlock Text="{Binding Path=/ErrorContent}"/>
                </TextBlock.ToolTip>
            </TextBlock>

        </StackPanel>

    </ControlTemplate>

    <!-- The style base for the TextBlocks -->
    <Style TargetType="{x:Type TextBlock}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>

    <!-- The style base for the Special TextBoxes -->
    <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Blue"/>

        <Style.Triggers>
            <Trigger Property="Control.IsKeyboardFocused" Value="True">
                <Setter Property="Control.Background" Value="Yellow"/>
            </Trigger>
            <Trigger Property="Validation.HasError" Value="True">
                <Setter Property="ToolTip"
                    Value="{Binding RelativeSource={x:Static RelativeSource
                        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
                <Setter Property="Foreground" Value="Red"/>
                <Setter Property="BorderBrush" Value="Red"/>
                <Setter Property="BorderThickness" Value="2"/>
            </Trigger>

        </Style.Triggers>

    </Style>

    <!-- A sample Customer resource object for the Bindings -->
    <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
        Amount1="0" />

</Window.Resources>
```

```
<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
  <!-- Amount1 (Byte type) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Byte): " />
  <zeus:ByteTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
    Language="el"
    ErrorMessageOnIncorrectValueType="Please, enter a valid Byte value!"
    Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
      StringFormat={}{0:C0}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give an amount">
  </zeus:ByteTextBox>
</Grid>
</Window>
```

Μία κανονική κατάσταση του **ByteTextBox**:

Amount 1 (Byte): 23 €

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Byte value!"

Amount 1 (Byte): 270 €



Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Byte.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Byte.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Byte.

XAML:

```
<zeus:ByteTextBox x:Name="txtOptions"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtOptions.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



ShortTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Short**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class ShortTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:ShortTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **ShortTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Short**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>

    <!-- The Validation Error ControlTemplate -->
    <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

            <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
                <AdornedElementPlaceholder x:Name="placeholder"/>
            </Border>

            <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
                <TextBlock.ToolTip >
                    <TextBlock Text="{Binding Path=/ErrorContent}"/>
                </TextBlock.ToolTip>
            </TextBlock>

        </StackPanel>

    </ControlTemplate>

    <!-- The style base for the TextBlocks -->
    <Style TargetType="{x:Type TextBlock}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>

    <!-- The style base for the Special TextBoxes -->
    <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Blue"/>

        <Style.Triggers>
            <Trigger Property="Control.IsKeyboardFocused" Value="True">
                <Setter Property="Control.Background" Value="Yellow"/>
            </Trigger>
            <Trigger Property="Validation.HasError" Value="True">
                <Setter Property="ToolTip"
                    Value="{Binding RelativeSource={x:Static RelativeSource
                        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
                <Setter Property="Foreground" Value="Red"/>
                <Setter Property="BorderBrush" Value="Red"/>
                <Setter Property="BorderThickness" Value="2"/>
            </Trigger>

        </Style.Triggers>

    </Style>

    <!-- A sample Customer resource object for the Bindings -->
    <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
        Amount1="0" />

</Window.Resources>
```



```
<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
  <!-- Amount1 (Short type) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Short): " />
  <zeus:ShortTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
    Language="el"
    ErrorMessageOnIncorrectValueType="Please enter a valid Short value!"
    Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
      StringFormat={}{0:C0}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give an amount">
  </zeus:ShortTextBox>
</Grid>
</Window>
```

Μία κανονική κατάσταση του **ShortTextBox**:

Amount 1 (Short):

Παρακάτω, βλέπουμε την **ένδειξη λάθους αφού ο χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Short value!"

Amount 1 (Short): 

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Short.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Short.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Short.

XAML:

```
<zeus:ShortTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



IntegerTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Integer**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class IntegerTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:IntegerTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **IntegerTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Integer**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```

<Window.Resources>

    <!-- The Validation Error ControlTemplate -->
    <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

            <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
                <AdornedElementPlaceholder x:Name="placeholder"/>
            </Border>

            <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
                <TextBlock.ToolTip >
                    <TextBlock Text="{Binding Path=/ErrorContent}"/>
                </TextBlock.ToolTip>
            </TextBlock>

        </StackPanel>

    </ControlTemplate>

    <!-- The style base for the TextBlocks -->
    <Style TargetType="{x:Type TextBlock}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>

    <!-- The style base for the Special TextBoxes -->
    <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Blue"/>

        <Style.Triggers>
            <Trigger Property="Control.IsKeyboardFocused" Value="True">
                <Setter Property="Control.Background" Value="Yellow"/>
            </Trigger>
            <Trigger Property="Validation.HasError" Value="True">
                <Setter Property="ToolTip"
                    Value="{Binding RelativeSource={x:Static RelativeSource
                        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
                <Setter Property="Foreground" Value="Red"/>
                <Setter Property="BorderBrush" Value="Red"/>
                <Setter Property="BorderThickness" Value="2"/>
            </Trigger>

        </Style.Triggers>

    </Style>

    <!-- A sample Customer resource object for the Bindings -->
    <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
        Amount1="0" />

</Window.Resources>

```

```
<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
  <!-- Amount1 (Integer type) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Integer): " />
  <zeus:IntegerTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
    Language="el"
    ErrorMessageOnIncorrectValueType="Please enter a valid Integer value!"
    Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
      StringFormat={}{0:C0}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give an amount">
  </zeus:IntegerTextBox>
</Grid>
</Window>
```

Μία κανονική κατάσταση του **IntegerTextBox**:

Amount 1 (Integer):

Παρακάτω, βλέπουμε την **ένδειξη λάθους αφού ο χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Integer value!"

Amount 1 (Integer): 

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Integer.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Integer.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Integer.

XAML:

```
<zeus:IntegerTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```




LongTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Long**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class LongTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:LongTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **LongTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Long**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
        ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>

  <!-- The Validation Error ControlTemplate -->
  <ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

      <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
              Background="#11FF0000" IsHitTestVisible="False" >
        <AdornedElementPlaceholder x:Name="placeholder"/>
      </Border>

      <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
              VerticalAlignment="Center" Margin="5,0,0,0">
        <TextBlock.ToolTip >
          <TextBlock Text="{Binding Path=/ErrorContent}"/>
        </TextBlock.ToolTip>
      </TextBlock>

    </StackPanel>

  </ControlTemplate>

  <!-- The style base for the TextBlocks -->
  <Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
  </Style>

  <!-- The style base for the Special TextBoxes -->
  <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
      <Trigger Property="Control.IsKeyboardFocused" Value="True">
        <Setter Property="Control.Background" Value="Yellow"/>
      </Trigger>
      <Trigger Property="Validation.HasError" Value="True">
        <Setter Property="ToolTip"
              Value="{Binding RelativeSource={x:Static RelativeSource
              .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
        <Setter Property="Foreground" Value="Red"/>
        <Setter Property="BorderBrush" Value="Red"/>
        <Setter Property="BorderThickness" Value="2"/>
      </Trigger>
    </Style.Triggers>
  </Style>

  <!-- A sample Customer resource object for the Bindings -->
  <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
              Amount1="0" />

```

```
</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

    <!-- Amount1 (Long type) -->
    <TextBlock Grid.Row="2" Text="Amount 1 (Long): " />
    <zeus:LongTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
        Language="el"
        ErrorMessageOnIncorrectValueType="Please enter a valid Long value!"
        Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
            StringFormat={{0:C0}}}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}"
        ToolTip="Please, give an amount">
    </zeus:LongTextBox>

</Grid>

</Window>
```

Μία κανονική κατάσταση του **LongTextBox**:

Amount 1 (Long):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Long value!"

Amount 1 (Long): 

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Long.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Long.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Long.

XAML:

```
<zeus:LongTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



SingleTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Single**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class SingleTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxControls"
```

Χρήση:

```
<zeus:SingleTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **SingleTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Single**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxControls"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>

  <!-- The Validation Error ControlTemplate -->
  <ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

      <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
              Background="#11FF0000" IsHitTestVisible="False" >
        <AdornedElementPlaceholder x:Name="placeholder"/>
      </Border>

      <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
              VerticalAlignment="Center" Margin="5,0,0,0">
        <TextBlock.ToolTip >
          <TextBlock Text="{Binding Path=/ErrorContent}"/>
        </TextBlock.ToolTip>
      </TextBlock>

    </StackPanel>

  </ControlTemplate>

  <!-- The style base for the TextBlocks -->
  <Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
  </Style>

  <!-- The style base for the Special TextBoxes -->
  <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
      <Trigger Property="Control.IsKeyboardFocused" Value="True">
        <Setter Property="Control.Background" Value="Yellow"/>
      </Trigger>
      <Trigger Property="Validation.HasError" Value="True">
        <Setter Property="ToolTip"
              Value="{Binding RelativeSource={x:Static RelativeSource
              .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
        <Setter Property="Foreground" Value="Red"/>
        <Setter Property="BorderBrush" Value="Red"/>
        <Setter Property="BorderThickness" Value="2"/>
      </Trigger>
    </Style.Triggers>
  </Style>

  <!-- A sample Customer resource object for the Bindings -->
  <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
              Amount1="0" />

```

```
</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
  <!-- Amount1 (Single type) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Single): " />
  <zeus:SingleTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
    Language="el"
    ErrorMessageOnIncorrectValueType="Please enter a valid Single value!"
    Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
      StringFormat={}{0:C2}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give an amount">
  </zeus:SingleTextBox>
</Grid>
</Window>
```

Μία κανονική κατάσταση του **SingleTextBox**:

Amount 1 (Single):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Single value!"

Amount 1 (Single):  Please enter a valid Single value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Single.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου `Single`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueTypeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου `Single`.

XAML:

```
<zeus:SingleTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



DoubleTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Double**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class DoubleTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:DoubleTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **DoubleTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Double**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
        ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>

    <!-- The Validation Error ControlTemplate -->
    <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

            <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
                <AdornedElementPlaceholder x:Name="placeholder"/>
            </Border>

            <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
                <TextBlock.ToolTip >
                    <TextBlock Text="{Binding Path=/ErrorContent}"/>
                </TextBlock.ToolTip>
            </TextBlock>

        </StackPanel>

    </ControlTemplate>

    <!-- The style base for the TextBlocks -->
    <Style TargetType="{x:Type TextBlock}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>

    <!-- The style base for the Special TextBoxes -->
    <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Blue"/>

        <Style.Triggers>
            <Trigger Property="Control.IsKeyboardFocused" Value="True">
                <Setter Property="Control.Background" Value="Yellow"/>
            </Trigger>
            <Trigger Property="Validation.HasError" Value="True">
                <Setter Property="ToolTip"
                    Value="{Binding RelativeSource={x:Static RelativeSource
                        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
                <Setter Property="Foreground" Value="Red"/>
                <Setter Property="BorderBrush" Value="Red"/>
                <Setter Property="BorderThickness" Value="2"/>
            </Trigger>
        </Style.Triggers>

    </Style>

    <!-- A sample Customer resource object for the Bindings -->
    <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
        Amount1="0" />

</Window.Resources>
```

```
<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
  <!-- Amount1 (Double type) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Double): " />
  <zeus:DoubleTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
    Language="el"
    ErrorMessageOnIncorrectValueType="Please enter a valid Double value!"
    Text="{Binding Path=Amount1, UpdateSourceTrigger=LostFocus,
      StringFormat={}{0:C2}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give an amount">
  </zeus:DoubleTextBox>
</Grid>
</Window>
```

Μία κανονική κατάσταση του **DoubleTextBox**:

Amount 1 (Double):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Double value!"

Amount 1 (Double): 


Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Double.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Double.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Double.

XAML:

```
<zeus:DoubleTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



DecimalTextBox

Ένα **TextBox** που επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου **Decimal**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class DecimalTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:DecimalTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **DecimalTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την ιδιότητα **Text** με την ιδιότητα **Amount1**, τύπου **Decimal**, ενός **Customer** αντικειμένου (source object). Η κλάση **Customer** ορίζεται στο project και συνπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην ιδιότητα **ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
        ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```



```
<Window.Resources>

  <!-- The Validation Error ControlTemplate -->
  <ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

      <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
              Background="#11FF0000" IsHitTestVisible="False" >
        <AdornedElementPlaceholder x:Name="placeholder"/>
      </Border>

      <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
              VerticalAlignment="Center" Margin="5,0,0,0">
        <TextBlock.ToolTip >
          <TextBlock Text="{Binding Path=/ErrorContent}"/>
        </TextBlock.ToolTip>
      </TextBlock>

    </StackPanel>

  </ControlTemplate>

  <!-- The style base for the TextBlocks -->
  <Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
  </Style>

  <!-- The style base for the Special TextBoxes -->
  <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
      <Trigger Property="Control.IsKeyboardFocused" Value="True">
        <Setter Property="Control.Background" Value="Yellow"/>
      </Trigger>
      <Trigger Property="Validation.HasError" Value="True">
        <Setter Property="ToolTip"
              Value="{Binding RelativeSource={x:Static RelativeSource
              .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
        <Setter Property="Foreground" Value="Red"/>
        <Setter Property="BorderBrush" Value="Red"/>
        <Setter Property="BorderThickness" Value="2"/>
      </Trigger>
    </Style.Triggers>
  </Style>

  <!-- A sample Customer resource object for the Bindings -->
  <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
            Amount1="0" />

</Window.Resources>
```

```
<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
    <!-- Amount1 (Decimal type) -->
    <TextBlock Grid.Row="2" Text="Amount 1 (Decimal): " />
    <zeus:DecimalTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
        Language="el"
        ErrorMessageOnIncorrectValueType="Please enter a valid Decimal value!"
        Text="{Binding Path=Amount1, UpdateSourceTrigger=LostFocus,
            StringFormat={}{0:C2}}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}"
        ToolTip="Please, give an amount">
    </zeus:DecimalTextBox>
</Grid>
</Window>
```

Μία κανονική κατάσταση του **DecimalTextBox**:



Amount 1 (Decimal): 265,00 €

Please, give an amount

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης εισήγε μία λάθος τιμή. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Decimal value!"



Amount 1 (Decimal): 2.654m,00 €

Please enter a valid Decimal value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Decimal.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου `Decimal`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueTypeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου `Decimal`.

XAML:

```
<zeus:DecimalTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



Nullable Αριθμητικοί τύποι

Στα επόμενα, παρουσιάζονται τα **TextBoxes** για τους βασικούς **Nullable Αριθμητικούς τύπους** (**Byte?**, **Short?**, **Integer?**, **Long?**, **Single?**, **Double?** και **Decimal?**). Η είσοδος του αριθμού δεν είναι υποχρεωτική.

- [NullableByteTextBox](#)
- [NullableShortTextBox](#)
- [NullableIntegerTextBox](#)
- [NullableLongTextBox](#)
- [NullableSingleTextBox](#)
- [NullableDoubleTextBox](#)
- [NullableDecimalTextBox](#)



NullableByteTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Byte**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableByteTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableByteTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableByteTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1Nullable**, τύπου **Nullable(Of Byte)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Byte** ή ένα κενό string (**Nothing**). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα **κενό string**.

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
```

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>

</Style>
```

```

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Byte)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Byte)): " />
  <zeus:NullableByteTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
                           Language="el"
                           ErrorMessageOnIncorrectValueType="Please enter a valid Byte value!"
                           Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=LostFocus,
                                           StringFormat={}{0:N0}, TargetNullValue=''}"
                           Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                           Style="{StaticResource TextBoxStyle}"
                           ToolTip="Please, give an amount">

  </zeus:NullableByteTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableByteTextBox**:

Amount 1 (Nullable(Of Byte)):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Byte value!"

Amount 1 (Nullable(Of Byte)):  

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Byte).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Byte).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Byte).

XAML:

```
<zeus:NullableByteTextBox x:Name="txtOptions"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtOptions.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableShortTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Short**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableShortTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableShortTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableShortTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1Nullable**, τύπου **Nullable(Of Short)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Short** ή ένα κενό string (**Nothing**). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα **κενό string**.

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxControls"

xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>
</Style>
```

```

</Style>

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Short)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Short)): " />
  <zeus:NullableShortTextBox Grid.Row="2" Grid.Column="1" x:Name="txtAmount1"
                           Language="el"
                           ErrorMessageOnIncorrectValueType="Please enter a valid Short value!"
                           Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=PropertyChanged,
                                           StringFormat={{0:N0}, TargetNullValue=''}}"
                           Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                           Style="{StaticResource TextBoxStyle}"
                           ToolTip="Please, give an amount">
  </zeus:NullableShortTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableShortTextBox**:

Amount 1 (Nullable(Of Short)):

Παρακάτω, βλέπουμε την **ένδειξη λάθους αφού ο χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Short value!"

Amount 1 (Nullable(Of Short)):  Please enter a valid Short value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Short).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Short).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Short).

XAML:

```
<zeus:NullableShortTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableIntegerTextBox

Ένα **TextBox** που επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου **Integer**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableIntegerTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableIntegerTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableIntegerTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την ιδιότητα **Text** με την ιδιότητα **Amount1Nullable**, τύπου **Nullable(Of Integer)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Integer** ή ένα κενό string (**Nothing**). Η κλάση **Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην ιδιότητα **ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα κενό string.

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
```



```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"

xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>
</Style>
```

```

</Style>

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Integer)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Integer)): " />
  <zeus:NullableIntegerTextBox Grid.Row="2" Grid.Column="1"
                             x:Name="txtAmount1" Language="el"
                             ErrorMessageOnIncorrectValueType="Please enter a valid Integer value!"
                             Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=PropertyChanged,
                                             StringFormat={}{0:N0}, TargetNullValue=''}"
                             Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                             Style="{StaticResource TextBoxStyle}"
                             ToolTip="Please, give an amount">
  </zeus:NullableIntegerTextBox>

</Grid>


</Window>

```

Μία κανονική κατάσταση του **NullableIntegerTextBox**:

Amount 1 (Nullable(Of Integer)):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Integer value!"

Amount 1 (Nullable(Of Integer)):  Please enter a valid Integer value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Integer).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Integer).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Integer).

XAML:

```
<zeus:NullableIntegerTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableLongTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Long**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableLongTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NullableLongTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableLongTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1Nullable**, τύπου **Nullable(Of Long)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Long** ή ένα κενό string (**Nothing**). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα **κενό string**.

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
```

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>

</Style>
```

```

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Long)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Long)): " />
  <zeus:NullableLongTextBox Grid.Row="2" Grid.Column="1"
                           x:Name="txtAmount1" Language="el"
                           ErrorMessageOnIncorrectValueType="Please enter a valid Long value!"
                           Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=PropertyChanged,
                                           StringFormat={}{0:N0}, TargetNullValue=''}"
                           Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                           Style="{StaticResource TextBoxStyle}"
                           ToolTip="Please, give an amount">
  </zeus:NullableLongTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableLongTextBox**:

Amount 1 (Nullable(Of Long)):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Long value!"

Amount 1 (Nullable(Of Long)): 
Please enter a valid Long value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Long).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Long).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Long).

XAML:

```
<zeus:NullableLongTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableSingleTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Single**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableSingleTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableSingleTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableSingleTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1Nullable**, τύπου **Nullable(Of Single)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Single** ή ένα κενό string (**Nothing**). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα κενό string.

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
```

```
xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">
```

```
    <StackPanel Orientation="Horizontal">
```

```
        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>
```

```
        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>
```

```
    </StackPanel>
```

```
</ControlTemplate>
```

```
<!-- The style base for the TextBlocks -->
```

```
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>
```

```
<!-- The style base for the Special TextBoxes -->
```

```
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>
</Style>
```

```
    <Style.Triggers>
```

```
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
```

```
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
```

```
    </Style.Triggers>
```

```
</Style.Triggers>
```

```
</Style>
```

```

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Single)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Single)): " />
  <zeus:NullableSingleTextBox Grid.Row="2" Grid.Column="1"
                             x:Name="txtAmount1" Language="el"
                             ErrorMessageOnIncorrectValueType="Please enter a valid Single value!"
                             Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=LostFocus,
                                             StringFormat={}{0:C2}, TargetNullValue=''}"
                             Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                             Style="{StaticResource TextBoxStyle}"
                             ToolTip="Please, give an amount">
  </zeus:NullableSingleTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableSingleTextBox**:

Amount 1 (Nullable(Of Single)): 1.230,50 €

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Single value!"

Amount 1 (Nullable(Of Single)): 1.230f,50 €

Please enter a valid Single value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Single).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Single).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Single).

XAML:

```
<zeus:NullableSingleTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableDoubleTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, τύπου Double**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableDoubleTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NullableDoubleTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableDoubleTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1Nullable**, τύπου **Nullable(Of Double)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Double** ή ένα κενό string (**Nothing**). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα **κενό string**.

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
```

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>

</Style>
```



```

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Double)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Double)): " />
  <zeus:NullableDoubleTextBox Grid.Row="2" Grid.Column="1"
                             x:Name="txtAmount1" Language="el"
                             ErrorMessageOnIncorrectValueType="Please enter a valid Double value!"
                             Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=LostFocus,
                                             StringFormat={}{0:C2}, TargetNullValue=''}"
                             Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                             Style="{StaticResource TextBoxStyle}"
                             ToolTip="Please, give an amount">
  </zeus:NullableDoubleTextBox>

</Grid>


</Window>

```

Μία κανονική κατάσταση του **NullableDoubleTextBox**:

Amount 1 (Nullable(Of Double)):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Double value!"

Amount 1 (Nullable(Of Double)):  Please enter a valid Double value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Double).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Double).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Double).

XAML:

```
<zeus:NullableDoubleTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableDecimalTextBox

Ένα **TextBox** που επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, τύπου **Decimal**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableDecimalTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NullableDecimalTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableDecimalTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την ιδιότητα **Text** με την ιδιότητα **Amount1Nullable**, τύπου **Nullable(Of Decimal)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Decimal** ή ένα κενό string (**Nothing**). Η κλάση **Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην ιδιότητα **ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα κενό string.

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
```

```
xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">
```

```
    <StackPanel Orientation="Horizontal">
```

```
        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>
```

```
        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>
```

```
    </StackPanel>
```

```
</ControlTemplate>
```

```
<!-- The style base for the TextBlocks -->
```

```
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>
```

```
<!-- The style base for the Special TextBoxes -->
```

```
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>
</Style>
```

```
    <Style.Triggers>
```

```
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
```

```
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
```

```
    </Style.Triggers>
```

```
</Style.Triggers>
```

```
</Style>
```

```

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Nullable(Of Decimal)) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Nullable(of Decimal)): " />
  <zeus:NullableDecimalTextBox Grid.Row="2" Grid.Column="1"
                             x:Name="txtAmount1" Language="el"
                             ErrorMessageOnIncorrectValueType="Please enter a valid Decimal value!"
                             Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=LostFocus,
                                           StringFormat={}{0:C2}, TargetNullValue=''}"
                             Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
                             Style="{StaticResource TextBoxStyle}"
                             ToolTip="Please, give an amount">
  </zeus:NullableDecimalTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableDecimalTextBox**:

Amount 1 (Nullable(Of Decimal)): 12,75 €

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης εισήγε μία λάθος τιμή. Αν πάει το δείκτη του ποντικού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Decimal value!"

Amount 1 (Nullable(Of Decimal)): A12,75 €

Please enter a valid Decimal value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Decimal).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable (Of Decimal).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable (Of Decimal).

XAML:

```
<zeus:NullableDecimalTextBox x:Name="txtAmount"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```




Γενικοί Αριθμητικοί τύποι

Στα επόμενα, παρουσιάζονται τα **TextBoxes** για την είσοδο δεδομένων **οποιοδήποτε αριθμητικού τύπου** καθορίζοντας την **ιδιότητα NumericType** (για **Byte, Short, Integer, Long, Single, Double** και **Decimal**).

Με άλλα λόγια, αποτελούν το ισοδύναμο των προηγούμενων. Για παράδειγμα, αν θέλουμε να χρησιμοποιήσουμε ένα **TextBox** για **Byte** αριθμούς έχουμε **δύο ισοδύναμες επιλογές**: Ένα **ByteTextBox** ή ένα **NumericTextBox** με την ιδιότητα **NumericType** σε "**ByteType**". Η διαφορά είναι σημασιολογικού χαρακτήρα.

Επιπλέον, εδώ έχουμε και δύο controls που **δέχονται έναν αριθμό οποιοδήποτε αριθμητικού τύπου σε ένα περιορισμένο εύρος τιμών (Range)**.

- [NumericTextBox](#)
- [NullableNumericTextBox](#)
- [NumericRangeTextBox](#)
- [NullableNumericRangeTextBox](#)



NumericTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου**. Αποτελεί ένα **γενικό αριθμητικό TextBox**. Μπορεί να χρησιμοποιηθεί ισοδύναμα αντί των εξειδικευμένων, ανά τύπο, TextBoxes (ByteTextBox, ShortTextBox κλπ). Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class NumericTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NumericTextBox NumericType="IntegerType" ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NumericTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Καθορίζουμε την **ιδιότητα NumericType** σε **"IntegerType"**. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1**, τύπου **Integer**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
        ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
```

```
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
```

```
<ControlTemplate x:Key="validationErrorTemplate">
```

```
    <StackPanel Orientation="Horizontal">
```

```
        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>
```

```
        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>
```

```
    </StackPanel>
```

```
</ControlTemplate>
```

```
<!-- The style base for the TextBlocks -->
```

```
<Style TargetType="{x:Type TextBlock}">
```

```
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>
```

```
<!-- The style base for the Special TextBoxes -->
```

```
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
```

```
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>
```

```
    <Style.Triggers>
```

```
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
```

```
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
```

```
    </Style.Triggers>
```

```
</Style>
```

```
</Style>
```

```
<!-- A sample Customer resource object for the Bindings -->
```

```
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
    Amount1="0" />
```

```

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

    <!-- Amount1 (Integer) -->
    <TextBlock Grid.Row="2" Text="Amount 1 (Integer): " />
    <zeus:NumericTextBox Grid.Row="2" Grid.Column="1"
        x:Name="txtAmount1" Language="el"
        NumericType="IntegerType"
        ErrorMessageOnIncorrectValueType="Please enter a valid Integer value!"
        Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
            StringFormat={}{0:N0}}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}"
        ToolTip="Please, give an amount">
    </zeus:NumericTextBox >

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NumericTextBox**:

Amount 1 (Integer):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Integer value!"

Amount 1 (Integer): !
Please enter a valid Integer value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με αυτήν που έχει οριστεί στην ιδιότητα NumericType.
NumericType	Καθορίζει τον τύπο των αριθμητικών δεδομένων (Byte, Short κλπ).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με τον τύπο που έχει καθοριστεί στην ιδιότητα NumericType.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή.

XAML:

```
<zeus:NumericTextBox x:Name="txtAmount" NumericType="IntegerType"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NumericValidationRule.NumericTypes.IntegerType  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```

NumericType

Καθορίζει τον αποδεκτό τύπο αριθμητικών δεδομένων για το NumericTextBox.

Σύνταξη:

VB:

```
Public Property NumericType As NumericValidationRule.NumericTypes
```

Τύπος: Zeus.WPF.Classes.Validations.NumericValidationRule.NumericTypes

Προσδιορίζουμε μία τιμή από την απαρίθμηση NumericTypes.

Η default τιμή είναι NumericValidationRule.NumericTypes.IntegerType

Dependency Property Information:

Identifier field: NumericTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή.

XAML:

```
<zeus:NumericTextBox x:Name="txtAmount" NumericType="IntegerType"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NumericValidationRule.NumericTypes.IntegerType  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NullableNumericTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου**. Αποτελεί ένα **γενικό αριθμητικό nullable TextBox**. Μπορεί να χρησιμοποιηθεί ισοδύναμα αντί των εξειδικευμένων, ανά τύπο, nullable TextBoxes (NullalbeByteTextBox, NullableShortTextBox κλπ). Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableNumericTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableNumericTextBox NumericType="IntegerType" ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableNumericTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Καθορίζουμε την **ιδιότητα NumericType** σε **"IntegerType"**. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα Amount1Nullable**, τύπου **Nullable(Of Integer)**, ενός **Customer** αντικειμένου (source object). Αυτό σημαίνει ότι η έγκυρη εισαχθείσα τιμή μπορεί να είναι αριθμός τύπου **Integer** ή ένα κενό string (**Nothing**). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα **κενό string**.

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```



```

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"

xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >

```

```
<Window.Resources>
```

```

<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>
</Style>

```

```

        </Style.Triggers>

    </Style>

    <!-- A sample Customer resource object for the Bindings -->
    <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
        Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

    <!-- Amount1 (Nullable (Of Integer)) -->
    <TextBlock Grid.Row="2" Text="Amount 1 (Nullable (Of Integer)): " />
    <zeus: NullableNumericTextBox Grid.Row="2" Grid.Column="1"
        x:Name="txtAmount1" Language="el"
        NumericType="IntegerType"
        ErrorMessageOnIncorrectValueType="Please enter a valid Integer value!"
        Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=PropertyChanged,
            StringFormat={}{0:N0}", TargetNullValue=''}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}"
        ToolTip="Please, give an amount">
    </zeus:NullableNumericTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableNumericTextBox**:

Amount 1 (Nullable of (Integer)):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Integer value!"

Amount 1 (Nullable of (Integer)): !
Please enter a valid Integer value!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με αυτήν που έχει οριστεί στην ιδιότητα NumericType.
NumericType	Καθορίζει τον τύπο των αριθμητικών δεδομένων (Byte, Short κλπ). Βεβαίως, στην περίπτωση αυτή, το control θα αποδεχθεί και το κενό string.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με τον τύπο που έχει καθοριστεί στην ιδιότητα `NumericType`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueTypeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο `Integer` και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή. Βεβαίως, το control, θα κάνει αποδεκτό και το κενό string.

XAML:

```
<zeus:NullableNumericTextBox x:Name="txtAmount" NumericType="IntegerType"
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NullableNumericValidationRule.NumericTypes.IntegerType
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```

NumericType

Καθορίζει τον **αποδεκτό τύπο αριθμητικών δεδομένων** για το NullableNumericTextBox.

Σύνταξη:

VB:

```
Public Property NumericType As NullableNumericValidationRule.NumericTypes
```

Τύπος: Zeus.WPF.Classes.Validations.NullableNumericValidationRule.NumericTypes

Προσδιορίζουμε μία τιμή από την απαρίθμηση NumericTypes.

Η default τιμή είναι NullableNumericValidationRule.NumericTypes.IntegerType

Dependency Property Information:

Identifier field: NumericTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή. Βεβαίως, το control, θα κάνει αποδεκτό και το κενό string.

XAML:

```
<zeus:NullableNumericTextBox x:Name="txtAmount" NumericType="IntegerType"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NullableNumericValidationRule.NumericTypes.IntegerType  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
```



NumericRangeTextBox

Ένα **TextBox** που επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου και σε συγκεκριμένο εύρος τιμών (**range**). Η τιμή απαιτείται (**required**).

Σύνταξη:

VB:

```
Public Class NumericRangeTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NumericRangeTextBox NumericType="IntegerType" Range="1:100" ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NumericRangeTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Καθορίζουμε την ιδιότητα **NumericType** σε **"IntegerType"** και την ιδιότητα **Range** σε **"1:100"** (αποδεκτοί αριθμοί από το 1 μέχρι και το 100). Συνδέουμε την ιδιότητα **Text** με την ιδιότητα **Amount1**, τύπου **Integer**, ενός **Customer** αντικειμένου (source object). Η κλάση **Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias **p**. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην ιδιότητα **ErrorMessageOnIncorrectValueType** ή στην **ErrorMessageValueOutOfRange**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
```

```
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>
```

```
<!-- The Validation Error ControlTemplate -->
```

```
<ControlTemplate x:Key="validationErrorTemplate">
```

```
    <StackPanel Orientation="Horizontal">
```

```
        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
```

```
            <AdornedElementPlaceholder x:Name="placeholder"/>
```

```
        </Border>
```

```
        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
```

```
            <TextBlock.ToolTip >
```

```
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
```

```
            </TextBlock.ToolTip>
```

```
        </TextBlock>
```

```
    </StackPanel>
```

```
</ControlTemplate>
```

```
<!-- The style base for the TextBlocks -->
```

```
<Style TargetType="{x:Type TextBlock}">
```

```
    <Setter Property="FontSize" Value="18"/>
```

```
    <Setter Property="VerticalAlignment" Value="Center" />
```

```
    <Setter Property="HorizontalAlignment" Value="Right"/>
```

```
</Style>
```

```
<!-- The style base for the Special TextBoxes -->
```

```
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
```

```
    <Setter Property="FontSize" Value="18"/>
```

```
    <Setter Property="VerticalContentAlignment" Value="Center" />
```

```
    <Setter Property="VerticalAlignment" Value="Center" />
```

```
    <Setter Property="Foreground" Value="Blue"/>
```

```
    <Style.Triggers>
```

```
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
```

```
            <Setter Property="Control.Background" Value="Yellow"/>
```

```
        </Trigger>
```

```
        <Trigger Property="Validation.HasError" Value="True">
```

```
            <Setter Property="ToolTip"
```

```
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
```

```
            <Setter Property="Foreground" Value="Red"/>
```

```
            <Setter Property="BorderBrush" Value="Red"/>
```

```
            <Setter Property="BorderThickness" Value="2"/>
```

```
        </Trigger>
```

```
    </Style.Triggers>
```

```
</Style>
```

```
<!-- A sample Customer resource object for the Bindings -->
```

```
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
            Amount1="0" />
```

```

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

    <!-- Amount1 (Integer) -->
    <TextBlock Grid.Row="2" Text="Amount 1 (Integer): " />
    <zeus:NumericRangeTextBox Grid.Row="2" Grid.Column="1"
        x:Name="txtAmount1" Language="el"
        NumericType="IntegerType" Range="1:100"
        ErrorMessageOnIncorrectValueType="Please enter a valid Integer value!"
        ErrorMessageOnValueOutOfRange="Please, enter an Integer
            value between 1 and 100!"
        Text="{Binding Path=Amount1, UpdateSourceTrigger=PropertyChanged,
            StringFormat={}{0:N0}}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}"
        ToolTip="Please, give an amount">
    </zeus:NumericRangeTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NumericRangeTextBox**:

Amount 1 (Integer):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Integer!"

Amount 1 (Integer): 

Please, enter a valid Integer!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με αυτήν που έχει οριστεί στην ιδιότητα NumericType.
ErrorMessageOnValueOutOfRange	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι στο αποδεκτό εύρος όπως καθορίζεται στην ιδιότητα Range.
NumericType	Καθορίζει τον τύπο των αριθμητικών δεδομένων (Byte, Short κλπ).
Range	Καθορίζει το εύρος των αποδεκτών τιμών . Το εύρος μπορεί να περιλαμβάνει πολλαπλές περιοχές . Για παράδειγμα, από το 1 μέχρι το 100 και από το 200 έως το 300.

ErrorMessageOnIncorrectValueType

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με τον τύπο που έχει καθοριστεί στην ιδιότητα `NumericType`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueTypeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο `Integer`, αποδεκτές τις τιμές στο εύρος 1 έως 100 και 400 έως 500 και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NumericRangeTextBox x:Name="txtAmount"  
    NumericType="IntegerType" Range="1:100;400:500"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"  
    ErrorMessageOnValueOutOfRange="Ο αριθμός δεν είναι μέσα στα όρια"  
    ToolTip="Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NumericRangeValidationRule.NumericTypes.IntegerType  
txtAmount.Range = "1:100;400:500"  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"  
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"  
txtAmount.ToolTip = "Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"
```

ErrorMessageOnValueOutOfRange

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι μέσα στο αποδεκτό εύρος τιμών, όπως αυτό ορίζεται στην ιδιότητα `Range`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnValueOutOfRange As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnValueOutOfRangeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο `Integer`, αποδεκτές τις τιμές στο εύρος 1 έως 100 και 400 έως 500 και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NumericRangeTextBox x:Name="txtAmount"  
    NumericType="IntegerType" Range="1:100;400:500"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"  
    ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"  
    ToolTip ="Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NumericRangeValidationRule.NumericTypes.IntegerType  
txtAmount.Range = "1:100;400:500"  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"  
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"  
txtAmount.ToolTip = "Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"
```

NumericType

Καθορίζει τον αποδεκτό τύπο αριθμητικών δεδομένων για το NumericRangeTextBox.

Σύνταξη:

VB:

```
Public Property NumericType As NumericRangeValidationRule.NumericTypes
```

Τύπος: Zeus.WPF.Classes.Validations.NumericRangeValidationRule.NumericTypes

Προσδιορίζουμε μία τιμή από την απαρίθμηση NumericTypes.

Η default τιμή είναι NumericRangeValidationRule.NumericTypes.IntegerType

Dependency Property Information:

Identifier field: NumericTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer, αποδεκτές τις τιμές στο εύρος 1 έως τη μέγιστη τιμή του τύπου Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NumericRangeTextBox x:Name="txtAmount" NumericType="IntegerType" Range="1:-"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"  
    ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα  
        όρια" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NumericRangeValidationRule.NumericTypes.IntegerType  
txtAmount.Range = "1:-"  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"  
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"
```

Range

Καθορίζει τον αποδεκτό εύρος των αποδεκτών τιμών για το NumericRangeTextBox. Το εύρος μπορεί να περιλαμβάνει πολλαπλές περιοχές. Η τιμή που θέτουμε είναι ένα string που έχει μορφή παρόμοια με το range στο Excel. Οι περιοχές διαχωρίζονται με το ελληνικό ερωτηματικό.

Σύνταξη:

VB:

```
Public Property Range As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή εύρους σε μορφή παρόμοια με το range του Excel. Η default τιμή είναι ένα κενό string, που υποδηλώνει αποδοχή σε όλο το εύρος του αριθμητικού τύπου.

Dependency Property Information:

Identifier field: RangeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer, αποδεκτές τις τιμές στο εύρος 1 έως τη μέγιστη τιμή του τύπου Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NumericRangeTextBox x:Name="txtAmount" NumericType="IntegerType" Range="1:-"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"  
    ErrorMessageOnValueOutOfRange="Ο αριθμός δεν είναι μέσα στα  
        όρια" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NumericRangeValidationRule.NumericTypes.IntegerType  
txtAmount.Range = "1:-"  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"  
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"
```

Παραδείγματα καθορισμού της τιμής Range (για δεδομένα Integer):

"1:100"	1 έως 100
"1:100;200:400"	1 έως 100 και 200 έως 400
"-100:100"	-100 έως 100
"-:0"	Integer.MinValue έως 0
"0:100;500:-"	0 έως 100 και 500 έως Integer.MaxValue



NullableNumericRangeTextBox

Ένα `TextBox` που επιτρέπει την είσοδο μόνο αριθμητικών δεδομένων, συγκεκριμένου τύπου και σε συγκεκριμένο εύρος τιμών (*range*). Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableNumericRangeTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:NullableNumericRangeTextBox NumericType="IntegerType" Range="1:100" ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα `NullableNumericRangeTextBox` μέσα σε ένα `Grid` ενός παραθύρου. Καθορίζουμε την ιδιότητα `NumericType` σε `"IntegerType"` και την ιδιότητα `Range` σε `"1:20;60:80"` (αποδεκτοί αριθμοί από το 1 μέχρι και το 20 και από το 60 μέχρι και το 80). Συνδέουμε την ιδιότητα `Text` με την ιδιότητα `Amount1Nullable`, τύπου `Nullable (Of Integer)`, ενός `Customer` αντικειμένου (source object). Η κλάση `Customer` ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias `p`. Επίσης, ορίζουμε ένα `ErrorTemplate` για το `Validation` (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην ιδιότητα `ErrorMessageOnIncorrectValueType` ή στην `ErrorMessageValueOutOfRange`).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την `TargetNullValue` σε ένα κενό string.

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
```

```

xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"

xmlns:p="clr-namespace:TextBoxControlsTestProject"
mc:Ignorable="d"
Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
<Window.Resources>

    <!-- The Validation Error ControlTemplate -->
    <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

            <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
                <AdornedElementPlaceholder x:Name="placeholder"/>
            </Border>

            <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
                <TextBlock.ToolTip >
                    <TextBlock Text="{Binding Path=/ErrorContent}"/>
                </TextBlock.ToolTip>
            </TextBlock>

        </StackPanel>

    </ControlTemplate>

    <!-- The style base for the TextBlocks -->
    <Style TargetType="{x:Type TextBlock}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>

    <!-- The style base for the Special TextBoxes -->
    <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Blue"/>

        <Style.Triggers>
            <Trigger Property="Control.IsKeyboardFocused" Value="True">
                <Setter Property="Control.Background" Value="Yellow"/>
            </Trigger>
            <Trigger Property="Validation.HasError" Value="True">
                <Setter Property="ToolTip"
                    Value="{Binding RelativeSource={x:Static RelativeSource
                        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
                <Setter Property="Foreground" Value="Red"/>
                <Setter Property="BorderBrush" Value="Red"/>
                <Setter Property="BorderThickness" Value="2"/>
            </Trigger>
        </Style.Triggers>
    </Style>

```



```

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
           Amount1Nullable="0" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- Amount1 (Integer) -->
  <TextBlock Grid.Row="2" Text="Amount 1 (Integer): " />
  <zeus:NullableNumericRangeTextBox Grid.Row="2" Grid.Column="1"
    x:Name="txtAmount1" Language="el"
    NumericType="IntegerType" Range="1:20;60:80"
    ErrorMessageOnIncorrectValueType="Please enter a valid Integer value!"
    ErrorMessageOnValueOutOfRange="Please, enter an Integer value
    between 1 and 20 and between 60 and 80!"
    Text="{Binding Path=Amount1Nullable, UpdateSourceTrigger=PropertyChanged,
    StringFormat={}{0:N0}, TargetNullValue=''}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give an amount">
  </zeus:NullableNumericRangeTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση του **NullableNumericRangeTextBox**:

Amount 1 (Integer):

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης εισήγε μία λάθος τιμή**. Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please enter a valid Integer!"

Amount 1 (Integer): 

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με αυτήν που έχει οριστεί στην ιδιότητα NumericType.
ErrorMessageOnValueOutOfRange	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι στο αποδεκτό εύρος όπως καθορίζεται στην ιδιότητα Range.
NumericType	Καθορίζει τον τύπο των αριθμητικών δεδομένων (Byte, Short κλπ).
Range	Καθορίζει το εύρος των αποδεκτών τιμών . Το εύρος μπορεί να περιλαμβάνει πολλαπλές περιοχές . Για παράδειγμα, από το 1 μέχρι το 100 και από το 200 έως το 300.

ErrorMessageOnIncorrectValueType

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι σύμφωνη με τον τύπο που έχει καθοριστεί στην ιδιότητα `NumericType`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueTypeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο `Integer`, αποδεκτές τις τιμές στο εύρος 1 έως 100 και 400 έως 500 και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NullableNumericRangeTextBox x:Name="txtAmount"
    NumericType="IntegerType" Range="1:100;400:500"
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"
    ErrorMessageOnValueOutOfRange="Ο αριθμός δεν είναι μέσα στα όρια"
    ToolTip="Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NullableNumericRangeValidationRule.NumericTypes.IntegerType
txtAmount.Range = "1:100;400:500"
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"
txtAmount.ToolTip = "Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"
```

ErrorMessageOnValueOutOfRange

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι μέσα στο αποδεκτό εύρος τιμών, όπως αυτό ορίζεται στην ιδιότητα `Range`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnValueOutOfRange As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: `ErrorMessageOnValueOutOfRangeProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο `Integer`, αποδεκτές τις τιμές στο εύρος 1 έως 100 και 400 έως 500 και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NullableNumericRangeTextBox x:Name="txtAmount"  
    NumericType="IntegerType" Range="1:100;400:500"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"  
    ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"  
    ToolTip = "Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NullableNumericRangeValidationRule.NumericTypes.IntegerType  
txtAmount.Range = "1:100;400:500"  
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"  
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"  
txtAmount.ToolTip = "Παρακαλώ, δώστε έναν αριθμό από 1-100 και 400-500"
```

NumericType

Καθορίζει τον αποδεκτό τύπο αριθμητικών δεδομένων για το NullableNumericRangeTextBox.

Σύνταξη:

VB:

```
Public Property NumericType As NullableNumericRangeValidationRule.NumericTypes
```

Τύπος: Zeus.WPF.Classes.Validations.NullableNumericRangeValidationRule.NumericTypes

Προσδιορίζουμε μία τιμή από την απαρίθμηση NumericTypes.

Η default τιμή είναι NullableNumericRangeValidationRule.NumericTypes.IntegerType

Dependency Property Information:

Identifier field: NumericTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer, αποδεκτές τις τιμές στο εύρος 1 έως τη μέγιστη τιμή του τύπου Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NullableNumericRangeTextBox x:Name="txtAmount"
    NumericType="IntegerType" Range="1:-"
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"
    ErrorMessageOnValueOutOfRange="Ο αριθμός δεν είναι μέσα στα
    όρια" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NullableNumericRangeValidationRule.NumericTypes.IntegerType
txtAmount.Range = "1:-"
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"
```

Range

Καθορίζει τον αποδεκτό εύρος των αποδεκτών τιμών για το NullableNumericRangeTextBox. Το εύρος μπορεί να περιλαμβάνει πολλαπλές περιοχές. Η τιμή που θέτουμε είναι ένα string που έχει μορφή παρόμοια με το range στο Excel. Οι περιοχές διαχωρίζονται με το ελληνικό ερωτηματικό.

Σύνταξη:

VB:

```
Public Property Range As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή εύρους σε μορφή παρόμοια με το range του Excel. Η default τιμή είναι ένα κενό string, που υποδηλώνει αποδοχή σε όλο το εύρος του αριθμητικού τύπου.

Dependency Property Information:

Identifier field: RangeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτό τον τύπο Integer, αποδεκτές τις τιμές στο εύρος 1 έως τη μέγιστη τιμή του τύπου Integer και το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης εισάγει μία λανθασμένη τιμή και το μήνυμα "Ο αριθμός δεν είναι μέσα στα όρια" όταν ο χρήστης εισάγει μία τιμή εκτός του αποδεκτού εύρους.

XAML:

```
<zeus:NullableNumericRangeTextBox x:Name="txtAmount"
    NumericType="IntegerType" Range="1:-"
    ErrorMessageOnIncorrectValueType="Μη έγκυρη τιμή"
    ErrorMessageOnValueOutOfRange="Ο αριθμός δεν είναι μέσα στα
    όρια" ... />
```

VB:

```
Imports Zeus.WPF.Classes.Validations
```

```
txtAmount.NumericType = NullableNumericRangeValidationRule.NumericTypes.IntegerType
txtAmount.Range = "1:-"
txtAmount.ErrorMessageOnIncorrectValueType = "Μη έγκυρη τιμή"
txtAmount.ErrorMessageOnValueOutOfRange = "Ο αριθμός δεν είναι μέσα στα όρια"
```

Παραδείγματα καθορισμού της τιμής Range (για δεδομένα Double, culture="el"):

"1,5:100"	1,5 έως 100
"1,5:100,8;200:400"	1,5 έως 100,8 και 200 έως 400
"-100:100"	-100 έως 100
"-:0"	Double.MinValue έως 0
"0,1:99,9;500,1:-"	0,1 έως 99,9 και 500,1 έως Double.MaxValue



TextBoxes για Αλφαριθμητικά και Αλφαβητικά δεδομένα

Στα επόμενα, παρουσιάζονται τα **TextBoxes** για **είσοδο αλφαριθμητικών και αλφαβητικών δεδομένων**.

- [RequiredTextBox](#)
- [AlphabeticTextBox](#)
- [RequiredAlphabeticTextBox](#)



RequiredTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο αλφαριθμητικών δεδομένων (string)** όπου η τιμή **απαιτείται (required)**. Με άλλα λόγια, το **TextBox** δεν μπορεί να είναι κενό.

Σύνταξη:

VB:

```
Public Class RequiredTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:RequiredTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **RequiredTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα City**, τύπου **String**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnEmpty**).

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo  
ntrols"  
  
    xmlns:p="clr-namespace:TextBoxControlsTestProject"  
    mc:Ignorable="d"  
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >  
  
<Window.Resources>
```

```

<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>

</Style>

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
    Amount1="0" City="Athens"/>

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

```

```
<!-- City (String type) -->
<TextBlock Grid.Row="2" Text="City: " />
<zeus:RequiredTextBox Grid.Row="2" Grid.Column="1" x:Name="txtCity"
    ErrorMessageOnEmpty="The field cannot be empty!"
    Text="{Binding Path=City, UpdateSourceTrigger=PropertyChanged}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please give a city name"
    Width="200" HorizontalAlignment="Left">
</zeus:RequiredTextBox>

</Grid>

</Window>
```

Μία κανονική κατάσταση του **RequiredTextBox**:

City: Athens

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο **χρήστης δεν έχει εισάγει μία τιμή** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "The field cannot be empty!"

City: 

The field cannot be empty!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnEmpty	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης δεν εισάγει κάποια τιμή στο RequiredTextBox.

ErrorMessageOnEmpty

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης δεν εισάγει κάποια τιμή στο **RequiredTextBox**.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnEmpty As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για την περίπτωση που το RequiredTextBox είναι κενό.

Dependency Property Information:

Identifier field: ErrorMessageOnEmptyProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Απαιτείται" να εμφανίζεται όταν ο χρήστης δεν έχει δώσει κάποια τιμή.

XAML:

```
<zeus:RequiredTextBox x:Name="txtCity" ErrorMessageOnEmpty="Απαιτείται" ... />
```

VB:

```
txtCity.ErrorMessageOnEmpty = "Απαιτείται"
```



AlphabeticTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο αλφαβητικών χαρακτήρων**. Μπορούμε να καθορίσουμε ποιό θα είναι αυτοί οι χαρακτήρες. Επιπλέον, μπορούμε να καθορίσουμε να γίνονται αποδεκτοί και κάποιοι **έξτρα χαρακτήρες** (κενό, θαυμαστικό, παύλα, τελεία κλπ). Η τιμή δεν απαιτείται (can be empty).

Σύνταξη:

VB:

```
Public Class AlphabeticTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:AlphabeticTextBox  
    AlphabeticCharacters="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"  
    ExtraCharacters=".;,:?()-' " ... />
```

Παρατηρήσεις:

Το `AlphabeticTextBox` μπορεί να χρησιμοποιηθεί γενικότερα αν θέλουμε να περιορίσουμε την είσοδο σε συγκεκριμένους αλφαριθμητικούς χαρακτήρες μόνο. Για παράδειγμα, αν θέλουμε να περιορίσουμε την είσοδο μόνο στους χαρακτήρες A B C και τα ψηφία 1 2 3 4 τότε μπορούμε να θέσουμε τις ιδιότητες `AlphabeticCharacters="ABC1234"` και `ExtraCharacters=""` ή `AlphabeticCharacters="ABC"` και `ExtraCharacters="1234"`.

Η διαφορά μεταξύ των ιδιοτήτων `AlphabeticCharacters` και `ExtraCharacters` είναι σημασιολογική. Λειτουργικά, ό,τι χαρακτήρες θέτουμε σε αυτές τις δύο ιδιότητες γίνεται αποδεκτοί.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **RequiredAlphabeticTextBox** κι ένα **AlphabeticTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** του

RequiredAlphabeticTextBox με την **ιδιότητα LastName**, τύπου **String**, ενός **Customer** αντικειμένου (source object). Παρόμοια, Συνδέουμε την **ιδιότητα Text** του **AlphabeticTextBox** με την **ιδιότητα FirstName**, τύπου **String**. Η διαφορά είναι ότι στο πρώτο είναι υποχρεωτικό να δώσουμε ένα Επώνυμο ενώ το δεύτερο μπορεί να είναι κενό. Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnEmpty** ή στην **ErrorMessageOnIncorrectValue**).

```
<Window x:Class="MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
  xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
ntrols"

  xmlns:p="clr-namespace:TextBoxControlsTestProject"
  mc:Ignorable="d"
  Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >

<Window.Resources>

  <!-- The Validation Error ControlTemplate -->
  <ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

      <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
        Background="#11FF0000" IsHitTestVisible="False" >
        <AdornedElementPlaceholder x:Name="placeholder"/>
      </Border>

      <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
        VerticalAlignment="Center" Margin="5,0,0,0">
        <TextBlock.ToolTip >
          <TextBlock Text="{Binding Path=/ErrorContent}"/>
        </TextBlock.ToolTip>
      </TextBlock>

    </StackPanel>

  </ControlTemplate>

  <!-- The style base for the TextBlocks -->
  <Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
  </Style>

  <!-- The style base for the Special TextBoxes -->
  <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
```

```

        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
    Amount1="0" City="Athens" />

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

    <!-- LastName (String type). The value is required. -->
    <TextBlock Text="LastName: " />
    <zeus:RequiredAlphabeticTextBox Grid.Column="1" x:Name="txtLastName"
        AlphabeticCharacters="ABCDEFGHIJKLMNOPQRSTUVWXYZ
            cdefghijklmnopqrstuvwxyz"
        ExtraCharacters=".!;,:~()-' "
        ErrorMessageOnEmpty="The field cannot be empty!"
        ErrorMessageOnIncorrectValue="Please, give a valid Lastname!"
        Text="{Binding Path=LastName, UpdateSourceTrigger=PropertyChanged}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}">
    </zeus:RequiredAlphabeticTextBox >

    <!-- FirstName (String type). The value is not required. -->
    <TextBlock Grid.Row="1" Text="FirstName: " />
    <zeus:AlphabeticTextBox Grid.Row="1" Grid.Column="1" x:Name="txtFirstName"
        ExtraCharacters="- . "
        ErrorMessageOnIncorrectValue="Please, give a valid Firstname!"
        Text="{Binding Path=FirstName, UpdateSourceTrigger=PropertyChanged}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}">
    </zeus:AlphabeticTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση των **RequiredAlphaTextBox** και **AlphaTextBox** :

LastName : FirstName :

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης δεν έχει εισάγει μία τιμή **LastName** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "The field cannot be empty!"

LastName :

FirstName :

The field cannot be empty!

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης έχει εισάγει έναν μη έγκυρο χαρακτήρα **για το LastName** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please, give a valid Lastname!"

LastName :

FirstName :

Please, give a valid Lastname!

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης έχει εισάγει έναν μη έγκυρο χαρακτήρα **για το FirstName** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please, give a valid Firstname!"

LastName :

FirstName :

Please, give a valid Firstname!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValue	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει κάποια μη αποδεκτή τιμή στο AlphabeticTextBox .
AlphabeticCharacters	Καθορίζει τους αποδεκτούς αλφαβητικούς χαρακτήρες.
ExtraCharacters	Καθορίζει επιπλέον αποδεκτούς χαρακτήρες εκτός των αλφαβητικών (κενό, θαυμαστικό, παύλα, τελεία κλπ).

ErrorMessageOnIncorrectValue

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει κάποια μη αποδεκτή τιμή στο `AlphabeticTextBox`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValue As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα λάθους για την περίπτωση που στο `AlphabeticTextBox` έχει εισαχθεί μία μη αποδεκτή τιμή.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης δώσει κάποια λανθασμένη τιμή.

XAML:

```
<zeus:AlphabeticTextBox x:Name="txtFirstName"  
    ErrorMessageOnIncorrectValue="Μη έγκυρη τιμή" ... />
```

VB:

```
txtFirstName.ErrorMessageOnIncorrectValue = "Μη έγκυρη τιμή"
```

AlphabeticCharacters

Καθορίζει τους αποδεκτούς αλφαβητικούς χαρακτήρες για το `AlphabeticTextBox`.

Σύνταξη:

VB:

```
Public Property AlphabeticCharacters As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα περιέχει το σύνολο των αλφαβητικών χαρακτήρων που θα είναι αποδεκτοί στο `AlphabeticTextBox`.

By default, αποδεκτοί χαρακτήρες είναι αυτοί του Αγγλικού αλφάβητου, δηλαδή η τιμή `"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"`

Παρατηρήσεις:

Γενικότερα, στην ιδιότητα `AlphabeticCharacters` μπορούμε να θέσουμε κι άλλους μη αλφαβητικούς χαρακτήρες ή και όλους τους χαρακτήρες που θα είναι αποδεκτοί χωρίς να χρησιμοποιήσουμε την ιδιότητα `ExtraCharacters`. Προτείνεται όμως, για σημασιολογικούς και διαχειριστικούς λόγους να ξεχωρίσουμε τους χαρακτήρες σε αλφαβητικούς και μή ώστε να τους θέτουμε στις αντίστοιχες ιδιότητες.

Dependency Property Information:

Identifier field: `AlphabeticCharactersProperty`

Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτούς μόνο τους χαρακτήρες του Ελληνικού αλφάβητου.

XAML:

```
<zeus:AlphabeticTextBox x:Name="txtLastName"  
    AlphabeticCharacters="ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζηθι  
        κλμνξοπρστυφχψωάέύίώ"  
    ErrorMessageOnIncorrectValue="Μη έγκυρη τιμή" ... />
```

VB:

```
txtLastName.AlphabeticCharacters = _  
    ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζηθικλμνξοπρστυφχψωάέύίώ"
```

Παράδειγμα 2:

Στο επόμενο παράδειγμα, καθορίζουμε αποδεκτούς τους Αγγλικούς και Ελληνικούς χαρακτήρες, σε run-time mode. Οι Αγγλικοί χαρακτήρες είναι καθορισμένοι εξ' ορισμού, οπότε απλά προσθέτουμε δυναμικά το string που περιέχει το σύνολο των Ελληνικών χαρακτήρων.

VB:

```
Const GreekAlphabeticCharacters As String = _  
    "ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζηθικλμνξοπρστυφχψωάέύίώ"
```

...

```
txtLastName.AlphabeticCharacters += GreekAlphabeticCharacters
```

ExtraCharacters

Καθορίζει **επιπλέον αποδεκτούς χαρακτήρες εκτός των αλφαβητικών** (κενό, θαυμαστικό, παύλα, τελεία κλπ) για το **AlphabeticTextBox**.

Σύνταξη:

VB:

```
Public Property ExtraCharacters As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα περιέχει το σύνολο των έξτρα χαρακτήρων που θα είναι αποδεκτοί στο AlphabeticTextBox.

By default, έξτρα αποδεκτοί χαρακτήρες είναι οι εξής : `.!;, :?()&- '` (τελεία, θαυμαστικό, κόμμα, άνω-κάτω τελεία, αγγλικό ερωτηματικό, αριστερή παρένθεση, δεξιά παρένθεση, αγγλικό και, παύλα, κενό, απόστροφος).

Παρατηρήσεις:

Γενικότερα, στην **ιδιότητα AlphabeticCharacters** μπορούμε να θέσουμε κι άλλους μη αλφαβητικούς χαρακτήρες ή και όλους τους χαρακτήρες που θα είναι αποδεκτοί χωρίς να χρησιμοποιήσουμε την **ιδιότητα ExtraCharacters**. Προτείνεται όμως, για σημασιολογικούς και διαχειριστικούς λόγους να ξεχωρίσουμε τους χαρακτήρες σε αλφαβητικούς και μή ώστε να τους θέτουμε στις αντίστοιχες ιδιότητες.

Dependency Property Information:

Identifier field: ExtraCharactersProperty

Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτούς χαρακτήρες μόνο τα ψηφία 0 1 2 3 4 5 6 7 8 και 9 και τίποτα άλλο.

XAML:

```
<zeus:AlphabeticTextBox x:Name="txtZipCode"
    AlphabeticCharacters=""
    ExtraCharacters="0123456789"
    ErrorMessageOnIncorrectValue="Μη έγκυρος κωδικός περιοχής " ... />
```

VB:

```
txtZipCode.AlphabeticCharacters = ""
```

```
txtZipCode.ExtraCharacters = "0123456789"  
txtZipCode.ErrorMessageOnIncorrectValue = "Μη έγκυρος κωδικός περιοχής"
```

Παράδειγμα 2:

Στο επόμενο παράδειγμα, καθορίζουμε έξτρα αποδεκτούς χαρακτήρες, σε run-time mode. Πέρα από τους εξ' ορισμού έξτρα, απλά προσθέτουμε δυναμικά το string που περιέχει και τα ψηφία 0 1 2 3 4 5 6 7 8 και 9.

VB:

```
Const NumberExtraCharacters As String = "0123456789"
```

...

```
txtField.ExtraCharacters += NumberExtraCharacters
```



RequiredAlphabeticTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο αλφαβητικών χαρακτήρων**. Μπορούμε να καθορίσουμε ποιοί θα είναι αυτοί οι χαρακτήρες. Επιπλέον, μπορούμε να καθορίσουμε να γίνονται αποδεκτοί και κάποιοι **έξτρα χαρακτήρες** (κενό, θαυμαστικό, παύλα, τελεία κλπ). Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class RequiredAlphabetic TextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:RequiredRequiredAlphabeticTextBox
    AlphabeticCharacters="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    ExtraCharacters=".!;,;:~()-' " ... />
```

Παρατηρήσεις:

Το RequiredAlphabeticTextBox μπορεί να χρησιμοποιηθεί γενικότερα αν θέλουμε να περιορίσουμε την είσοδο σε συγκεκριμένους αλφαριθμητικούς χαρακτήρες μόνο. Για παράδειγμα, αν θέλουμε να περιορίσουμε την είσοδο μόνο στους χαρακτήρες A B C και τα ψηφία 1 2 3 4 τότε μπορούμε να θέσουμε τις ιδιότητες AlphabeticCharacters="ABC1234" και ExtraCharacters="" ή AlphabeticCharacters="ABC" και ExtraCharacters="1234".

Η διαφορά μεταξύ των ιδιοτήτων AlphabeticCharacters και ExtraCharacters είναι σημασιολογική. Λειτουργικά, ό,τι χαρακτήρες θέτουμε σε αυτές τις δύο ιδιότητες γίνεται αποδεκτοί.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **RequiredAlphabeticTextBox** κι ένα **AlphabeticTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** του **RequiredAlphabeticTextBox** με την **ιδιότητα LastName**, τύπου **String**, ενός **Customer** αντικειμένου (source object). Παρόμοια, Συνδέουμε την **ιδιότητα Text** του **AlphabeticTextBox** με την **ιδιότητα FirstName**, τύπου **String**. Η διαφορά είναι ότι στο πρώτο είναι υποχρεωτικό να δώσουμε ένα Επώνυμο ενώ το δεύτερο μπορεί να είναι κενό. Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnEmpty** ή στην **ErrorMessageOnIncorrectValue**

```
<Window x:Class="MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
  xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxControls"
  xmlns:p="clr-namespace:TextBoxControlsTestProject"
  mc:Ignorable="d"
  Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >

<Window.Resources>

  <!-- The Validation Error ControlTemplate -->
  <ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

      <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
        Background="#11FF0000" IsHitTestVisible="False" >
        <AdornedElementPlaceholder x:Name="placeholder"/>
      </Border>

      <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
        VerticalAlignment="Center" Margin="5,0,0,0">
        <TextBlock.ToolTip >
          <TextBlock Text="{Binding Path=/ErrorContent}"/>
        </TextBlock.ToolTip>
      </TextBlock>

    </StackPanel>

  </ControlTemplate>

  <!-- The style base for the TextBlocks -->
  <Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
  </Style>

  <!-- The style base for the Special TextBoxes -->
  <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
```

```

<Setter Property="VerticalAlignment" Value="Center" />
<Setter Property="Foreground" Value="Blue"/>

<Style.Triggers>
  <Trigger Property="Control.IsKeyboardFocused" Value="True">
    <Setter Property="Control.Background" Value="Yellow"/>
  </Trigger>
  <Trigger Property="Validation.HasError" Value="True">
    <Setter Property="ToolTip"
      Value="{Binding RelativeSource={x:Static RelativeSource
        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
    <Setter Property="Foreground" Value="Red"/>
    <Setter Property="BorderBrush" Value="Red"/>
    <Setter Property="BorderThickness" Value="2"/>

  </Trigger>

</Style.Triggers>

</Style>

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
  Amount1="0" City="Athens"/>

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

  <!-- LastName (String type). The value is required. -->
  <TextBlock Text="LastName: " />
  <zeus:RequiredAlphabeticTextBox Grid.Column="1" x:Name="txtLastName"
    AlphabeticCharacters="ABCDEFGHIJKLMNOPQRSTUVWXYZab
      cdefghijklmnopqrstuvwxyz"
    ExtraCharacters=".!;, :?()-' "
    ErrorMessageOnEmpty="The field cannot be empty!"
    ErrorMessageOnIncorrectValue="Please, give a valid Lastname!"
    Text="{Binding Path=LastName, UpdateSourceTrigger=PropertyChanged}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}">
  </zeus:RequiredAlphabeticTextBox >

  <!-- FirstName (String type). The value is not required. -->
  <TextBlock Grid.Row="1" Text="FirstName: " />
  <zeus:AlphabeticTextBox Grid.Row="1" Grid.Column="1" x:Name="txtFirstName"
    ExtraCharacters="- . "
    ErrorMessageOnIncorrectValue="Please, give a valid Firstname!"
    Text="{Binding Path=FirstName, UpdateSourceTrigger=PropertyChanged}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}">
  </zeus:AlphabeticTextBox>

</Grid>

</Window>

```

Μία κανονική κατάσταση των **RequiredAlphaTextBox** και **AlphaTextBox** :

LastName :

FirstName :

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης **δεν έχει εισάγει μία τιμή LastName** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "The field cannot be empty!"

LastName :

FirstName :

The field cannot be empty!

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης **έχει εισάγει έναν μη έγκυρο χαρακτήρα για το LastName** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please, give a valid Lastname!"

LastName :

FirstName :

Please, give a valid Lastname!

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης **έχει εισάγει έναν μη έγκυρο χαρακτήρα για το FirstName** . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please, give a valid Firstname!"

LastName :

FirstName :

Please, give a valid Firstname!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnEmpty	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης αφήνει το <code>RequiredAlphabeticTextBox</code> κενό.
ErrorMessageOnIncorrectValue	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει κάποια μη αποδεκτή τιμή στο <code>RequiredAlphabeticTextBox</code> .
AlphabeticCharacters	Καθορίζει τους αποδεκτούς αλφαβητικούς χαρακτήρες.
ExtraCharacters	Καθορίζει επιπλέον αποδεκτούς χαρακτήρες εκτός των αλφαβητικών (κενό, θαυμαστικό, παύλα, τελεία κλπ).

ErrorMessageOnEmpty

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης αφήνει κενό το `RequiredAlphabeticTextBox`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnEmpty As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα λάθους για την περίπτωση που το `RequiredAlphabeticTextBox` είναι κενό.

Dependency Property Information:

Identifier field: `ErrorMessageOnEmptyProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Απαιτείται να δώσετε όνομα" να εμφανίζεται όταν ο χρήστης δεν έχει εισάγει κάποια τιμή.

XAML:

```
<zeus:RequiredAlphabeticTextBox x:Name="txtFirstName"  
    ErrorMessageOnEmpty="Απαιτείται να δώσετε όνομα" ... />
```

VB:

```
txtFirstName.ErrorMessageOnEmpty = "Απαιτείται να δώσετε όνομα"
```

ErrorMessageOnIncorrectValue

Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει κάποια μη αποδεκτή τιμή στο `RequiredAlphabeticTextBox`.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValue As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το μήνυμα λάθους για την περίπτωση που στο `RequiredAlphabeticTextBox` έχει εισαχθεί μία μη αποδεκτή τιμή.

Dependency Property Information:

Identifier field: `ErrorMessageOnIncorrectValueProperty`

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη τιμή" να εμφανίζεται όταν ο χρήστης δώσει κάποια λανθασμένη τιμή.

XAML:

```
<zeus:RequiredAlphabeticTextBox x:Name="txtFirstName"  
    ErrorMessageOnIncorrectValue="Μη έγκυρη τιμή" ... />
```

VB:

```
txtFirstName.ErrorMessageOnIncorrectValue = "Μη έγκυρη τιμή"
```

AlphabeticCharacters

Καθορίζει τους αποδεκτούς αλφαβητικούς χαρακτήρες για το RequiredAlphabeticTextBox.

Σύνταξη:

VB:

```
Public Property AlphabeticCharacters As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα περιέχει το σύνολο των αλφαβητικών χαρακτήρων που θα είναι αποδεκτοί στο RequiredAlphabeticTextBox.

By default, αποδεκτοί χαρακτήρες είναι αυτοί του Αγγλικού αλφάβητου, δηλαδή η τιμή "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"

Παρατηρήσεις:

Γενικότερα, στην ιδιότητα **AlphabeticCharacters** μπορούμε να θέσουμε κι άλλους μη αλφαβητικούς χαρακτήρες ή και όλους τους χαρακτήρες που θα είναι αποδεκτοί χωρίς να χρησιμοποιήσουμε την ιδιότητα **ExtraCharacters**. Προτείνεται όμως, για σημασιολογικούς και διαχειριστικούς λόγους να ξεχωρίσουμε τους χαρακτήρες σε αλφαβητικούς και μή ώστε να τους θέτουμε στις αντίστοιχες ιδιότητες.

Dependency Property Information:

Identifier field: AlphabeticCharactersProperty

Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτούς μόνο τους χαρακτήρες του Ελληνικού αλφάβητου.

XAML:

```
<zeus:RequiredAlphabeticTextBox x:Name="txtLastName"
    AlphabeticCharacters="ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζη
        θικλμνξοπρστυφχψωάέύίώ"
    ErrorMessageOnEmpty="Το Επώνυμο απαιτείται!"
    ErrorMessageOnIncorrectValue="Μη έγκυρη τιμή" ... />
```

VB:

```
txtLastName.AlphabeticCharacters = _
    ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζηθικλμνξοπρστυφχψωάέύίώ"
```

Παράδειγμα 2:

Στο επόμενο παράδειγμα, καθορίζουμε αποδεκτούς τους Αγγλικούς και Ελληνικούς χαρακτήρες, σε run-time mode. Οι Αγγλικοί χαρακτήρες είναι καθορισμένοι εξ' ορισμού, οπότε απλά προσθέτουμε δυναμικά το string που περιέχει το σύνολο των Ελληνικών χαρακτήρων.

VB:

```
Const GreekAlphabeticCharacters As String = _  
    "ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζηθικλμνξοπρστυφχψωάέύίώ"
```

...

```
txtLastName.AlphabeticCharacters += GreekAlphabeticCharacters
```


ExtraCharacters

Καθορίζει **επιπλέον αποδεκτούς χαρακτήρες εκτός των αλφαβητικών** (κενό, θαυμαστικό, παύλα, τελεία κλπ) **για το RequiredAlphabeticTextBox**.

Σύνταξη:

VB:

```
Public Property ExtraCharacters As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα περιέχει το σύνολο των έξτρα χαρακτήρων που θα είναι αποδεκτοί στο RequiredAlphabeticTextBox.

By default, έξτρα αποδεκτοί χαρακτήρες είναι οι εξής : `.!;, :?()&- '` (τελεία, θαυμαστικό, κόμμα, άνω-κάτω τελεία, αγγλικό ερωτηματικό, αριστερή παρένθεση, δεξιά παρένθεση, αγγλικό και, παύλα, κενό, απόστροφος).

Παρατηρήσεις:

Γενικότερα, στην **ιδιότητα AlphabeticCharacters** μπορούμε να θέσουμε κι άλλους μη αλφαβητικούς χαρακτήρες ή και όλους τους χαρακτήρες που θα είναι αποδεκτοί χωρίς να χρησιμοποιήσουμε την **ιδιότητα ExtraCharacters**. Προτείνεται όμως, για σημασιολογικούς και διαχειριστικούς λόγους να ξεχωρίσουμε τους χαρακτήρες σε αλφαβητικούς και μή ώστε να τους θέτουμε στις αντίστοιχες ιδιότητες.

Dependency Property Information:

Identifier field: ExtraCharactersProperty

Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε ως αποδεκτούς χαρακτήρες μόνο τα ψηφία 0 1 2 3 4 5 6 7 8 και 9 και τίποτα άλλο.

XAML:

```
<zeus:AlphabeticTextBox x:Name="txtZipCode"
    AlphabeticCharacters=""
    ExtraCharacters="0123456789"
    ErrorMessageOnIncorrectValue="Μη έγκυρος κωδικός περιοχής " ... />
```

VB:

```
txtZipCode.AlphabeticCharacters = ""
```

```
txtZipCode.ExtraCharacters = "0123456789"  
txtZipCode.ErrorMessageOnIncorrectValue = "Μη έγκυρος κωδικός περιοχής"
```

Παράδειγμα 2:

Στο επόμενο παράδειγμα, καθορίζουμε έξτρα αποδεκτούς χαρακτήρες, σε run-time mode. Πέρα από τους εξ' ορισμού έξτρα, απλά προσθέτουμε δυναμικά το string που περιέχει και τα ψηφία 0 1 2 3 4 5 6 7 8 και 9.

VB:

```
Const NumberExtraCharacters As String = "0123456789"
```

...

```
txtField.ExtraCharacters += NumberExtraCharacters
```



TextBoxes για Ημερομηνίες

Στα επόμενα, παρουσιάζονται τα **TextBoxes** για είσοδο ημερομηνιακών δεδομένων.

- [DateTextBox](#)
- [NullableDateTextBox](#)



DateTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο δεδομένων δεδομένων ημερομηνίας (Date type)**. Η τιμή απαιτείται (required).

Σύνταξη:

VB:

```
Public Class DateTextBox
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon
trols"
```

Χρήση:

```
<zeus:DateTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **DateTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα DateInsertion**, τύπου **Date**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCo
        ntrols"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
<Window.Resources>
```

```

<!-- The Validation Error ControlTemplate -->
<ControlTemplate x:Key="validationErrorTemplate">

    <StackPanel Orientation="Horizontal">

        <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
            Background="#11FF0000" IsHitTestVisible="False" >
            <AdornedElementPlaceholder x:Name="placeholder"/>
        </Border>

        <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
            VerticalAlignment="Center" Margin="5,0,0,0">
            <TextBlock.ToolTip >
                <TextBlock Text="{Binding Path=/ErrorContent}"/>
            </TextBlock.ToolTip>
        </TextBlock>

    </StackPanel>

</ControlTemplate>

<!-- The style base for the TextBlocks -->
<Style TargetType="{x:Type TextBlock}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Right"/>
</Style>

<!-- The style base for the Special TextBoxes -->
<Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
    <Setter Property="FontSize" Value="18"/>
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Foreground" Value="Blue"/>

    <Style.Triggers>
        <Trigger Property="Control.IsKeyboardFocused" Value="True">
            <Setter Property="Control.Background" Value="Yellow"/>
        </Trigger>
        <Trigger Property="Validation.HasError" Value="True">
            <Setter Property="ToolTip"
                Value="{Binding RelativeSource={x:Static RelativeSource
                    .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
            <Setter Property="Foreground" Value="Red"/>
            <Setter Property="BorderBrush" Value="Red"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>

</Style>

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
    Amount1="0" City="Athens"/>

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

```

```
<!-- DateInsertion field (Date type) -->
<TextBlock Grid.Row="2" Text="Date insertion (Date): " />
<zeus:DateTextBox Grid.Row="2" Grid.Column="1"
    x:Name="txtDateInsertion" Language="el"
    ErrorMessageOnIncorrectValueType="Please, enter a valid Date!"
    Text="{Binding Path=DateInsertion, UpdateSourceTrigger=LostFocus,
        StringFormat={}{0:d}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
    Style="{StaticResource TextBoxStyle}"
    ToolTip="Please, give the insertion date."
    Width="200" HorizontalAlignment="Left">

</Grid>

</Window>
```

Μία κανονική κατάσταση του **DateTextBox (culture "el")**:

Date insertion (Date) : 12/3/2018

Παρακάτω, βλέπουμε την **ένδειξη λάθους** αφού ο χρήστης έχει εισάγει μία λανθασμένη τιμή . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please, enter a valid Date!"

Date insertion (Date) : 6/13/2018



Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Date.

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Date.

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη ημερομηνία" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Date.

XAML:

```
<zeus:DateTextBox x:Name="txtPaymentDate"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη ημερομηνία" ... />
```

VB:

```
txtPaymentDate.ErrorMessageOnIncorrectValueType = "Μη έγκυρη ημερομηνία"
```




NullableDateTextBox

Ένα **TextBox** που επιτρέπει την **είσοδο δεδομένων δεδομένων ημερομηνίας (Date type)**. Η τιμή δεν απαιτείται (can be null).

Σύνταξη:

VB:

```
Public Class NullableDateTextBox  
    Inherits TextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
trols"
```

Χρήση:

```
<zeus:NullableDateTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε ένα **NullableDateTextBox** μέσα σε ένα **Grid** ενός παραθύρου. Συνδέουμε την **ιδιότητα Text** με την **ιδιότητα DateInsertionNullable**, τύπου **Nullable(of Date)**, ενός **Customer** αντικειμένου (source object). Η **κλάση Customer** ορίζεται στο project και συνεπώς πρέπει να εισάγουμε και το αντίστοιχο namespace με alias p. Επίσης, ορίζουμε ένα **ErrorTemplate** για το **Validation** (όπου μέσω αυτού θα εμφανιστεί το μήνυμα λάθους που ορίζουμε στην **ιδιότητα ErrorMessageOnIncorrectValueType**).

Ιδιαίτερη προσοχή πρέπει να δοθεί στην **binding έκφραση της ιδιότητας Text** όπου πρέπει να θέσουμε την **TargetNullValue** σε ένα **κενό string**.

```
<Window x:Class="MainWindow"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"  
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxCon  
ntrols"  
  
    xmlns:p="clr-namespace:TextBoxControlsTestProject"  
    mc:Ignorable="d"
```

```

Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
<Window.Resources>

    <!-- The Validation Error ControlTemplate -->
    <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

            <Border BorderBrush="#FFCB2E2E" BorderThickness="1"
                Background="#11FF0000" IsHitTestVisible="False" >
                <AdornedElementPlaceholder x:Name="placeholder"/>
            </Border>

            <TextBlock Grid.Column="1" Foreground="Red" FontSize="20" Text="!"
                VerticalAlignment="Center" Margin="5,0,0,0">
                <TextBlock.ToolTip >
                    <TextBlock Text="{Binding Path=/ErrorContent}"/>
                </TextBlock.ToolTip>
            </TextBlock>

        </StackPanel>

    </ControlTemplate>

    <!-- The style base for the TextBlocks -->
    <Style TargetType="{x:Type TextBlock}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="HorizontalAlignment" Value="Right"/>
    </Style>

    <!-- The style base for the Special TextBoxes -->
    <Style x:Key="TextBoxStyle" TargetType="{x:Type TextBox}">
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="VerticalContentAlignment" Value="Center" />
        <Setter Property="VerticalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Blue"/>

        <Style.Triggers>
            <Trigger Property="Control.IsKeyboardFocused" Value="True">
                <Setter Property="Control.Background" Value="Yellow"/>
            </Trigger>
            <Trigger Property="Validation.HasError" Value="True">
                <Setter Property="ToolTip"
                    Value="{Binding RelativeSource={x:Static RelativeSource
                        .Self}, Path=(Validation.Errors)[0].ErrorContent}"/>
                <Setter Property="Foreground" Value="Red"/>
                <Setter Property="BorderBrush" Value="Red"/>
                <Setter Property="BorderThickness" Value="2"/>
            </Trigger>
        </Style.Triggers>
    </Style>

    <!-- A sample Customer resource object for the Bindings -->
    <p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
        Amount1="0" City="Athens" />

```

```

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">
    <!-- DateInsertion field (Nullable(Of Date) type) -->
    <TextBlock Grid.Row="2" Text="Date insertion (Nullable(Of Date)): " />
    <zeus:NullableDateTextBox Grid.Row="2" Grid.Column="1"
        x:Name="txtDateInsertion" Language="el"
        ErrorMessageOnIncorrectValueType="Please, enter a valid Date!"
        Text="{Binding Path=DateInsertionNullable, UpdateSourceTrigger=LostFocus,
            StringFormat={}{0:d}, TargetNullValue=''}"
        Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"
        Style="{StaticResource TextBoxStyle}"
        Tooltip="Please, give the insertion date."
        Width="200" HorizontalAlignment="Left">
    </zeus:NullableDateTextBox >


</Grid>
</Window>

```

Μία κανονική κατάσταση του **NullableDateTextBox (culture "el")**:

Date insertion (Nullable(Of Date)) :

Παρακάτω, βλέπουμε την ένδειξη λάθους αφού ο χρήστης έχει εισάγει μία λανθασμένη τιμή . Αν πάει το δείκτη του ποντικιού πάνω στο **θαυμαστικό** θα εμφανιστεί το μήνυμα λάθους "Please, enter a valid Date!"

Date insertion (Nullable(Of Date)) :  Please, enter a valid Date!

Ιδιότητες

Όνομα	Περιγραφή
ErrorMessageOnIncorrectValueType	Καθορίζει το μήνυμα λάθους που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable(Of Date).

ErrorMessageOnIncorrectValueType

Καθορίζει το **μήνυμα λάθους** που θα εμφανίζεται όταν ο χρήστης εισάγει τιμή που δεν είναι τύπου Nullable(Of Date).

Σύνταξη:

VB:

```
Public Property ErrorMessageOnIncorrectValueType As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το μήνυμα για λανθασμένη εισαγωγή τιμής.

Dependency Property Information:

Identifier field: ErrorMessageOnIncorrectValueTypeProperty

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το μήνυμα "Μη έγκυρη ημερομηνία" να εμφανίζεται όταν ο χρήστης εισάγει μία τιμή που δεν είναι τύπου Nullable(Of Date).

XAML:

```
<zeus:NullableDateTextBox x:Name="txtPaymentDate"  
    ErrorMessageOnIncorrectValueType="Μη έγκυρη ημερομηνία" ... />
```

VB:

```
txtPaymentDate.ErrorMessageOnIncorrectValueType = "Μη έγκυρη ημερομηνία"
```



ZRichTextBox

Ένα **RichTextBox** που επιτρέπει την **επεξεργασία εμπλουτισμένου κειμένου (rtf)**. Πρόκειται για ένα κανονικό RichTextBox με τη διαφορά ότι **εμμέσως μπορούμε να κάνουμε bind την ιδιότητα Document**.

Σύνταξη:

VB:

```
Public Class ZRichTextBox
    Inherits RichTextBox
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxControls"
```

Χρήση:

```
<zeus:ZRichTextBox ... />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε τοποθετήσει ένα **ZRichTextBox** control στο παράθυρο κι έχουμε συνδέσει την **ιδιότητα DocumentXamlText** με την ιδιότητα Notes του αντικειμένου Customer. Η ιδιότητα Notes (τύπου String), περιέχει ένα xaml text που αναπαριστά ένα FlowDocument. Με αυτόν τον τρόπο, θέτουμε περιεχόμενο στην ιδιότητα Document του ZRichTextBox.

```
<Window x:Class="MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:val="clr-namespace:Zeus.WPF.Classes.Validations;assembly=ZeusValidations"
    xmlns:zeus="clr-namespace:Zeus.WPF.Controls.TextBoxControls;assembly=ZeusTextBoxControls"
    xmlns:p="clr-namespace:TextBoxControlsTestProject"
    mc:Ignorable="d"
    Title="Sample Project for the Special TextBoxes" Height="431.915" Width="696.277" >
```

```
<Window.Resources>

...

<!-- A sample Customer resource object for the Bindings -->
<p:Customer x:Key="customer" LastName="Mouratidis" FirstName="Christos"
            Amount1="0" City="Athens"/>

</Window.Resources>

<Grid Margin="20" DataContext="{Binding Source={StaticResource customer}}">

...

<!-- Notes -->
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"
                  DocumentXamlText="{Binding Notes}" />

</Grid>

</Window>
```

Ένα παράδειγμα τιμής (xaml text) που περιέχει η ιδιότητα **Notes** του customer object είναι η εξής:

```
<FlowDocument FontFamily="" Arial"" FontStyle="" Normal"" FontWeight="" Normal""
FontSize="" 12"" TextAlignment="" Left"" PagePadding="" 5,0,5,0"" AllowDrop="" True""
xmlns="" http://schemas.microsoft.com/winfx/2006/xaml/presentation""></FlowDocument>
```

- Αυτή η τιμή τίθεται μέσω binding στην ιδιότητα DocumentXamlText και αυτομάτως διαμορφώνει την ιδιότητα Document.
- Αντίστροφα, αν επεξεργαστούμε το περιεχόμενο του ZRichTextBox (μορφοποιήσουμε έντονα, πλάγια κλπ) τότε αντίστοιχα διαμορφώνεται το xaml text της ιδιότητας DocumentXamlText και συνεπώς αποθηκεύεται αντίστοιχα στην ιδιότητα Notes. Έτσι, πετυχαίνουμε read/save εμπλουτισμένου κειμένου από/σε ένα source object.

Ιδιότητες

Όνομα	Περιγραφή
DocumentXamlText	Καθορίζει το <code>xaml text</code> , που αναπαριστά ένα <code>FlowDocument</code> , μέσω του οποίου τίθεται το περιεχόμενο του <code>ZRichTextBox</code> .
EmptyDocumentXamlText	Καθορίζει το <code>xaml text</code> , που αναπαριστά ένα κενό <code>FlowDocument</code> .
IsDocumentEmpty	Επιστρέφει True αν το <code>ZRichTextBox</code> είναι κενό (κενό <code>FlowDocument</code>).

DocumentXamlText

Καθορίζει το `xaml text`, που αναπαριστά ένα `FlowDocument`, μέσω του οποίου τίθεται το περιεχόμενο του `ZRichTextBox`.

Σύνταξη:

VB:

```
Public Property DocumentXamlText As String
```

Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που αναπαριστά ένα `FlowDocument`, το οποίο θέτει το περιεχόμενο του `ZRichTextBox`. Με αυτόν τον τρόπο, θέτουμε εμμέσως την τιμή της ιδιότητας `Document`.

Η default τιμή είναι ένα κενό `FlowDocument` με τα εξής χαρακτηριστικά:

```
<FlowDocument FontFamily="Arial" FontStyle="Normal" FontWeight="Normal"
FontSize="12" TextAlignment="Left" PagePadding="5,0,5,0"
AllowDrop="True"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"></FlowDo
cument>
```

Dependency Property Information:

Identifier field: `DocumentXamlTextProperty`

Παρατηρήσεις:

Ο σκοπός αυτής της ιδιότητας είναι να γίνει bind με μία ιδιότητα ενός source αντικειμένου. Επειδή, η ιδιότητα `Document` δεν μπορεί να γίνει bind (καθώς δεν είναι dependency), χρησιμοποιούμε την `DocumentXamlText`, η οποία θέτει εμμέσως ένα `FlowDocument` αντικείμενο στην `Document`.

Σε ένα σενάριο, μπορούμε να έχουμε ένα πεδίο π.χ. `Notes` μίας Βάσης Δεδομένων, όπου αποθηκεύουμε εμπλουτισμένο κείμενο σε μορφή `xaml text`. Για να το εμφανίσουμε και επεξεργαστούμε στην εφαρμογή, χρησιμοποιούμε ένα `ZRichTextBox` και συνδέουμε την ιδιότητα `DocumentXamlText` με την ιδιότητα `Notes` του source αντικειμένου.

Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε συνδέσει την **ιδιότητα `DocumentXamlText`** με την ιδιότητα `Notes` του αντικειμένου `Customer`. Η ιδιότητα `Notes` (τύπου `String`), περιέχει ένα `xaml text` που αναπαριστά ένα `FlowDocument`.

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176"  
Padding="10" DocumentXamlText="{Binding Notes}" />
```

- Η επεξεργασία του περιεχομένου του ZRichTextBox από τον χρήστη αλλάζει το FlowDocument και αυτόματα επαναδιαμορφώνεται το xaml text της ιδιότητας DocumentXamlText. Τελικά, η source ιδιότητα Notes θα έχει το επεξεργασμένο FlowDocument, το οποίο θα αποθηκευτεί στο αντίστοιχο πεδίο της Βάσης Δεδομένων. Με άλλα λόγια, **το binding είναι two-way.**

EmptyDocumentXamlText

Καθορίζει το `xaml text`, που περιγράφει ένα κενό `FlowDocument`, το οποίο θα τίθεται στο `ZRichTextBox` αν το τελευταίο είναι κενό.

Σύνταξη:

VB:

```
Public Property EmptyDocumentXamlText As String
```

Τύπος: System.String

Προσδιορίζουμε μία τιμή `String` που αναπαριστά ένα κενό `FlowDocument`, το οποίο θέτει το περιεχόμενο του `ZRichTextBox`, αν το τελευταίο είναι κενό.

Η default τιμή είναι ένα κενό `FlowDocument` με τα εξής χαρακτηριστικά:

```
"<FlowDocument FontFamily=""Arial"" FontStyle=""Normal"" FontWeight=""Normal""  
FontSize=""12"" TextAlignment=""Left"" PagePadding=""5,0,5,0""  
AllowDrop=""True""  
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation""></FlowDo  
cument>"
```

Dependency Property Information:

Identifier field: `EmptyDocumentXamlTextProperty`

Παράδειγμα:

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176"  
    Padding="10"  
    DocumentXamlText="{Binding Notes}" />
```

VB:

```
rtfBox.EmptyDocumentXamlText = "<FlowDocument FontFamily=""Calibri""  
FontStyle=""Normal"" FontWeight=""Normal"" FontSize=""12"" TextAlignment=""Left""  
PagePadding=""5,0,5,0"" AllowDrop=""True""  
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation"" />"
```

IsDocumentEmpty

Επιστρέφει **True** αν το **ZRichTextBox** είναι κενό (κενό **FlowDocument**).

Σύνταξη:

VB:

```
Public ReadOnly Property IsDocumentEmpty As Boolean
```

Τύπος: **System.Boolean**

Επιστρέφει μία τιμή Boolean που δηλώνει αν το ZRichTextBox είναι κενό. Η default τιμή είναι True.

Dependency Property Information:

Identifier field: IsDocumentEmptyProperty

Παράδειγμα:

Στο παρακάτω παράδειγμα, συνδέουμε την ιδιότητα **Visibility** ενός **TextBlock** με την ιδιότητα **IsDocumentEmpty** του **ZRichTextBox**, με τη βοήθεια ενός **Converter** που μετατρέπει μία τιμή Boolean σε **Visibility**. Αν το **ZRichTextBox** είναι κενό τότε εμφανίζεται με κόκκινα γράμματα το σχετικό μήνυμα;

XAML:

```
<Window.Resources>
    <p:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"/>
</Window.Resources>
<!-- Notes -->
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"
    DocumentXamlText="{Binding Notes}" />
<TextBlock Text="To ZRichTextBox είναι κενό"
    Visibility="{Binding ElementName=rtfBox, Path=IsDocumentEmpty, _
        Mode=OneWay, _
        Converter={StaticResource booleanToVisibilityConverter}}"
    Padding="5" Margin="5,0,0,0" FontSize="9" Foreground="Red"/>
```

Μέθοδοι

Όνομα	Περιγραφή
ClearDocument() As Boolean	<p>Καθαρίζει το περιεχόμενο περιεχόμενο του ZRichTextBox. Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές.</p> <p>Σημείωση: Το περιεχόμενο ενός RichTextBox είναι ένα FlowDocument. Έτσι, το καθάρισμα σημαίνει ότι τίθεται ένα κενό FlowDocument. Αυτό το FlowDocument περιγράφεται από την ιδιότητα EmptyDocumentXamlText.</p>
CopyDocument() As FlowDocument	<p>Αντιγράφει το περιεχόμενο του ZRichTextBox. Επιστρέφει ένα νέο FlowDocument.</p> <p>Μέσω αυτής της μεθόδου μπορούμε να αντιγράψουμε με ασφάλεια το Document του ZRichTextBox σε ένα άλλο RichTextBox/ZRichTextBox.</p>
GetDocumentAsXamlText() As String	<p>Επιστρέφει το περιεχόμενο του ZRichTextBox ως ένα xaml text, το οποίο περιγράφει το FlowDocument.</p>
GetPlainText() As String	<p>Επιστρέφει το περιεχόμενο του ZRichTextBox ως ένα plain text, δηλαδή χωρίς τις μορφοποιήσεις.</p>
SetDocument(newDocument As FlowDocument) As Boolean	<p>Θέτει ένα FlowDocument ως περιεχόμενο του ZRichTextBox. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.</p>

[SetDocumentFromXamlText\(flowDocumentXamlText As String\) As Boolean](#)

Θέτει ένα xaml text με root element το **<FlowDocument>** ως περιεχόμενο του **ZRichTextBox**. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

ClearDocument()

Καθαρίζει το περιεχόμενο περιεχόμενο του ZRichTextBox. Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές.

Σύνταξη:

VB :

```
Public Function ClearDocument()  
    As Boolean
```

Επιστρεφόμενη τιμή

Τύπος: System.Boolean

Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές. Τεχνικά μιλώντας, τίθεται ένα κενό FlowDocument με βάση την ιδιότητα EmptyDocumentXamlText .

Παρατηρήσεις:

Το περιεχόμενο ενός ZRichTextBox είναι ένα FlowDocument.

Παράδειγμα:

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"  
    DocumentXamlText="{Binding Notes}" />
```

VB:

```
Imports Zeus.WPF.Controls.TextBoxControls
```

...

```
rtfBox.ClearDocument()
```

CopyDocument()

Καθαρίζει το περιεχόμενο περιεχόμενο του ZRichTextBox. Επιστρέφει True, αν το καθάρισμα ήταν επιτυχές.

Σύνταξη:

VB :

```
Public Function CopyDocument()  
    As FlowDocument
```

Επιστρεφόμενη τιμή

Τύπος: System.Windows.Documents.FlowDocument

Επιστρέφει το περιεχόμενο FlowDocument.

Παρατηρήσεις:

Το περιεχόμενο ενός ZRichTextBox είναι ένα FlowDocument. Δεν μπορούμε να θέσουμε άμεσα την ιδιότητα Document ενός ZRichTextBox σε ένα Document που ανήκει σε άλλο ZRichTextBox/RichTextBox. Με άλλα λόγια, δεν μπορούν δύο RichTextboxes/ZRichTextboxes να αναφέρονται (point) στο ίδιο αντικείμενο FlowDocument. Έτσι, ο παρακάτω VB κώδικας:

```
richTextBox2.Document = richTextBox1.Document
```

θα προκαλέσει run-time error. Η χρήση της μεθόδου CopyDocument κάνει εύκολη την αντιγραφή καθώς, τεχνικά μιλώντας, κάνει xaml parse το αντικείμενο FlowDocument του source ZRichTextBox, το επαναδημιουργεί σε ένα νέο αντικείμενο FlowDocument και το αποθέτει στο target RichTextBox/ZRichTextBox.

Παράδειγμα:

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"  
    DocumentXamlText="{Binding Notes}" />  
<zeus:ZRichTextBox x:Name="rtfBox2" />
```

VB:

```
Imports Zeus.WPF.Controls.TextBoxControls
```

...

```
rtfBox2.Document= rtfBox.CopyDocument()
```


GetDocumentAsXamlText()

Επιστρέφει το περιεχόμενο του ZRichTextBox ως ένα xaml text, το οποίο περιγράφει το εσωτερικό FlowDocument.

Σύνταξη:

VB :

```
Public Function GetDocumentAsXamlText()  
    As String
```

Επιστρεφόμενη τιμή

Τύπος: System.String

Επιστρέφει το xaml text που περιγράφει το περιεχόμενο (εσωτερικό) FlowDocument.

Παρατηρήσεις:

Τεχνικά μιλώντας, η μέθοδος κάνει parsing το περιεχόμενο FlowDocument και το επιστρέφει ως xaml text με root element το <FlowDocument>. Ένα τέτοιο xaml text μπορεί να αποθηκευτεί σε ένα πεδίο μίας Βάσης Δεδομένων, το οποίο να κρατάει μορφοποιημένες σημειώσεις.

Παράδειγμα:

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"  
    DocumentXamlText="{Binding Notes}" />
```

VB:

```
Imports Zeus.WPF.Controls.TextBoxControls
```

...

```
Dim xamlText As String = rtfBox.GetDocumentAsXamlText()
```

GetPlainText()

Επιστρέφει το περιεχόμενο του ZRichTextBox ως ένα plain text, δηλαδή χωρίς τις μορφοποιήσεις.

Σύνταξη:

VB :

```
Public Function GetPlainText()  
    As String
```

Επιστρεφόμενη τιμή

Τύπος: System.String

Επιστρέφει το plain text (μόνο κείμενο) άνευ μορφοποιήσεων.

Παράδειγμα:

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"  
    DocumentXamlText="{Binding Notes}" />
```

VB:

```
Imports Zeus.WPF.Controls.TextBoxControls
```

...

```
Dim plainText As String = rtfBox.GetPlainText()
```

SetDocument(newDocument)

Θέτει ένα **FlowDocument** ως περιεχόμενο του **ZRichTextBox**. Επιστρέφει **True**, αν η απόθεση ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Function SetDocument(  
    newDocument As FlowDocument  
)  
    As Boolean
```

Παράμετροι:

newDocument

Τύπος: `System.Windows.Documents.FlowDocument`

Το αντικείμενο `FlowDocument`, το οποίο θα τεθεί ως περιεχόμενο.

Επιστρεφόμενη τιμή

Τύπος: `System.Boolean`

Επιστρέφει `True`, αν η απόθεση ήταν επιτυχής.

Παρατηρήσεις:

Το περιεχόμενο ενός `ZRichTextBox` είναι ένα `FlowDocument`.

Παράδειγμα:

Στο παρακάτω παράδειγμα, θέτουμε ως περιεχόμενο στο `rtfBox` ένα νέο `FlowDocument`. Το τελευταίο, μπορούμε να το "χτίσουμε" στον VB κώδικα εκτός από το να θέσουμε άμεσα ένα xaml string:

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"  
    DocumentXamlText="{Binding Notes}" />
```

VB:

```
Imports Zeus.WPF.Controls.TextBoxControls
```

...

```
Dim flowDoc As New FlowDocument() With {.Foreground = Brushes.Orange, _  
    .FontFamily = New FontFamily("Courier New"), _  
    .FontSize = 22}
```

```
With flowDoc
```

```
    .Blocks.Add(New Paragraph(New Run( "Δεν υπάρχει περιεχόμενο.")))
```

```
End With
```

```
If rtfBox.SetDocument(flowDoc) Then MessageBox.Show("Η απόθεση ήταν επιτυχής")
```

SetDocumentFromXamlText(flowDocumentXamlText)

Θέτει ένα xaml text με root element το <FlowDocument> ως περιεχόμενο του ZRichTextBox. Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Σύνταξη:

VB :

```
Public Function SetDocument(  
    flowDocumentXamlText As String  
)  
    As Boolean
```

Παράμετροι:

flowDocumentXamlText
Τύπος: System.String
Το xaml string με root element το <FlowDocument>.

Επιστρεφόμενη τιμή
Τύπος: System.Boolean
Επιστρέφει True, αν η απόθεση ήταν επιτυχής.

Παράδειγμα:

Στο παρακάτω παράδειγμα, θέτουμε ως περιεχόμενο στο rtfBox ένα νέο FlowDocument με βάση ένα xaml string.

XAML:

```
<!-- Notes -->  
<zeus:ZRichTextBox x:Name="rtfBox" Background="#FFF176" Padding="10"  
    DocumentXamlText="{Binding Notes}" />
```

VB:

```
Imports Zeus.WPF.Controls.TextBoxControls
```

...

```
Dim xamlString As String = "<FlowDocument Name=""richlowDoc2""  
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation""><Paragraph  
FontSize=""16"" Foreground=""Orange"">Δεν υπάρχει  
περιεχόμενο.</Paragraph></FlowDocument>"
```

```
If rtfBox.SetDocument(xamlString) Then MessageBox.Show("Η απόθεση ήταν επιτυχής")
```

Παράρτημα: Η κλάση Customer

Στα παραδείγματα της Βοήθειας κάνουμε εκτενή χρήση της **κλάσης Customer**. Οι **ιδιότητες** αυτής της κλάσης (πεδία) γίνονται binding με την ιδιότητα Text των TextBoxControls.

Σημείωση: Εδώ, η ιδιότητα Amount1 είναι τύπου Decimal και η ιδιότητα Amount1Nullable είναι τύπου Nullable(of Decimal). Σε παραδείγματα του Help όπου απαιτείται η ιδιότητα Amount1 να είναι Byte ή Integer κλπ. μπορείτε να θεωρήσετε ότι είναι του αυτού τύπου. Δεν αλλάζει κάτι άλλο. Παρόμοια ισχύει και για την ιδιότητα Amount1Nullable για τους αντίστοιχους nullable τύπους.

Η κλάση αυτή έχει οριστεί ως εξής:

```
Imports System.ComponentModel

'*****
'***** The class for the Data *****
'*****

Public Class Customer
    Implements INotifyPropertyChanged

    Public Event PropertyChanged As PropertyChangedEventHandler
        Implements INotifyPropertyChanged.PropertyChanged

    Public Sub OnPropertyChanged(ByVal sender As Object, ByVal e As PropertyChangedEventArgs)
        If e IsNot Nothing Then RaiseEvent PropertyChanged(Me, e)
    End Sub

    'Private backing fields.
    Private _lastName As String
    Private _firstName As String
    Private _amount1 As Decimal
    Private _amount1Nullable As Nullable(Of Decimal)
    Private _city As String
    Private _dateInsertion As Date
    Private _dateInsertionNullable As Nullable(Of Date)
    Private _notes As String

    'CLR properties - data fields.
    Public Property LastName As String
        Get
            Return _lastName
        End Get
        Set(ByVal value As String)
            _lastName = value
            OnPropertyChanged(Me, New PropertyChangedEventArgs("LastName"))
        End Set
    End Property

    Public Property FirstName As String
        Get
            Return _firstName
        End Get
        Set(ByVal value As String)
```

```

        _firstName = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("FirstName"))
    End Set
End Property

Public Property Amount1 As Decimal
    Get
        Return _amount1
    End Get
    Set(ByVal value As Decimal)
        _amount1 = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("Amount1"))
    End Set
End Property

Public Property Amount1Nullable As Nullable(Of Decimal)
    Get
        Return _amount1Nullable
    End Get
    Set(ByVal value As Nullable(Of Decimal))
        _amount1Nullable = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("Amount1Nullable"))
    End Set
End Property

Public Property City As String
    Get
        Return _city
    End Get
    Set(ByVal value As String)
        _city = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("City"))
    End Set
End Property

Public Property DateInsertion As Date
    Get
        Return _dateInsertion
    End Get
    Set(ByVal value As Date)
        _dateInsertion = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("DateInsertion"))
    End Set
End Property

Public Property DateInsertionNullable As Nullable(Of Date)
    Get
        Return _dateInsertionNullable
    End Get
    Set(ByVal value As Nullable(Of Date))
        _dateInsertionNullable = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("DateInsertionNullable"))
    End Set
End Property

Public Property Notes As String
    Get
        Return _notes
    End Get
    Set(ByVal value As String)
        _notes = value
        OnPropertyChanged(Me, New PropertyChangedEventArgs("Notes"))
    End Set
End Property

```

```
End Set  
End Property
```

```
'Constructor.  
Public Sub New()
```

```
    _lastName = String.Empty  
    _firstName = String.Empty  
    _amount1 = 0  
    _amount1Nullable = 0  
    _city= String.Empty  
    _dateInsertion = DateTime.Today  
    _dateInsertionNullable = Nothing  
    _notes = "<FlowDocument FontFamily=""Arial"" FontStyle=""Normal""  
FontWeight=""Normal"" FontSize=""12"" TextAlignment=""Left"" PagePadding=""5,0,5,0""  
AllowDrop=""True""  
xmlns=""http://schemas.microsoft.com/winfx/2006/xaml/presentation""></FlowDocument>"
```

```
End Sub
```

```
End Class
```