

Zeus ValueConverters Ver. 1.3

Manual

Copyright © 2023 Χρήστος Μουρατίδης

Πίνακας περιεχομένων

ZEUS VALUECONVERTERS VERSION 1.3	3
ΤΙ ΝΕΟ ΥΠΑΡΧΕΙ ΣΤΗΝ ΕΚΔΟΣΗ 1.2	7
ΤΙ ΝΕΟ ΥΠΑΡΧΕΙ ΣΤΗΝ ΕΚΔΟΣΗ 1.3	8
ΟΙ ΟΡΙΣΜΟΙ ΤΩΝ ΚΛΑΣΕΩΝ	9
ΔΙΑΝΟΜΗ	19
ΕΠΙΚΟΙΝΩΝΙΑ	20
BOOLEANREVERSECONVERTER	21
BOOLEANTOBITCONVERTER	23
ΙΔΙΟΤΗΤΕΣ	25
<i>BitWhenTrue</i>	26
<i>BitWhenFalse</i>	28
BOOLEANTOBRUSHCONVERTER	30
ΙΔΙΟΤΗΤΕΣ	33
<i>BrushWhenTrue</i>	34
<i>BrushWhenFalse</i>	36
BOOLEANTOFONTSIZECONVERTER	38
ΙΔΙΟΤΗΤΕΣ	41
<i>FontSizeWhenTrue</i>	42
<i>FontSizeWhenFalse</i>	44
BOOLEANTOFONTSTYLECONVERTER	46
ΙΔΙΟΤΗΤΕΣ	49
<i>FontStyleWhenTrue</i>	50
<i>FontStyleWhenFalse</i>	52
BOOLEANTOFONTWEIGHTCONVERTER	54
ΙΔΙΟΤΗΤΕΣ	57
<i>FontWeightWhenTrue</i>	58
<i>FontWeightWhenFalse</i>	60
BOOLEANTOIMAGESOURCECONVERTER	62
ΙΔΙΟΤΗΤΕΣ	65
<i>ImageSourceWhenTrue</i>	66
<i>ImageSourceWhenFalse</i>	68
BOOLEANTOSTRINGCONVERTER	70
ΙΔΙΟΤΗΤΕΣ	73
<i>StringWhenTrue</i>	74
<i>StringWhenFalse</i>	76
BOOLEANTOVISIBILITYCONVERTER	78
ΙΔΙΟΤΗΤΕΣ	81
<i>VisibilityWhenTrue</i>	82
<i>VisibilityWhenFalse</i>	84

DATENULLABLETOSTRINGCONVERTER	86
ΙΔΙΟΤΗΤΕΣ.....	89
<i>TextWhenNothing</i>	90
DATENULLABLETOCURRENTDATECONVERTER	91
IMAGESOURCETOVISIBILITYCONVERTER	93
ΙΔΙΟΤΗΤΕΣ.....	96
<i>VisibilityWhenHasSource</i>	97
<i>VisibilityWhenHasNothing</i>	100
IMAGESOURCETOSTRINGCONVERTER	103
ΙΔΙΟΤΗΤΕΣ.....	106
<i>TextWhenHasSource</i>	107
<i>TextWhenHasNothing</i>	109
NUMBERNULLABLETOSTRINGCONVERTER	111
ΙΔΙΟΤΗΤΕΣ.....	116
<i>TextWhenHasNothing</i>	117
NUMBERASNULLABLETOSTRINGCONVERTER	119
ΙΔΙΟΤΗΤΕΣ.....	124
<i>NumberAsNothing</i>	125
<i>TextWhenNumberNull</i>	127

Zeus ValueConverters version 1.3



Κλάσεις: BooleanReverseConverter, BooleanToBitConverter, BooleanToBrushConverter, BooleanToFontSizeConverter, BooleanToFontStyleConverter, BooleanToFontWeightConverter, BooleanToImageSourceConverter, BooleanToStringConverter, BooleanToVisibilityConverter, DateNullableToStringConverter, DateNullableToCurrentDateConverter, ImageSourceToVisibilityConverter, ImageSourceToStringConverter, NumberNullableToStringConverter, NumberAsNullableToStringConverter.

Inherits: System.Windows.Data.IValueConverter

Namespace: Zeus.WPF.Classes.ValueConverters

Assembly: ZeusValueConverters (in ZeusValueConverters.dll)

Dependencies: -

Περιγραφή

Η βιβλιοθήκη **Zeus ValueConverters** περιλαμβάνει (στην έκδοση 1.3) **δεκαπέντε κλάσεις ValueConverters**. Τα ValueConverters αντικείμενα μπορούν να χρησιμοποιηθούν σε binding εκφράσεις, όταν απαιτείται μετατροπή τιμής από έναν τύπο σε άλλον.

Ως γνωστόν, μία binding έκφραση συνδέει δύο ιδιότητες μεταξύ τους, μία source και μία target. Αν οι ιδιότητες αυτές είναι διαφορετικού τύπου τότε χρειάζεται να χρησιμοποιήσουμε στην binding έκφραση ένα αντικείμενο ValueConverter. Το ποιο γνωστό παράδειγμα, είναι η σύνδεση της ιδιότητας Visibility ενός γραφικού element (π.χ. TextBlock) , με μία Boolean ιδιότητα ενός source αντικειμένου. Αν η boolean ιδιότητα είναι True, τότε η ιδιότητα Visibility να έχει την τιμή Visible.

Στην περίπτωση που το γραφικό αντικείμενο είναι ένα input element (π.χ. TextBox), αυτό σημαίνει ότι δέχεται τιμή από τον χρήστη. Έτσι, η έκφραση binding είναι two-way. Δηλαδή, μία αλλαγή τιμής στο target element (TextBox) αλλάζει την τιμή στην συνδεδεμένη ιδιότητα του source αντικειμένου. Η κλάση μετατροπής (ValueConverter), που χρησιμοποιείται στην binding έκφραση, πρέπει να υποστηρίζει το two-way mode. Τεχνικά μιλώντας, στην υλοποίησή της, πρέπει να έχει κώδικα μετατροπής στην μέθοδο Convert (one-way) αλλά και στην μέθοδο ConvertBack (two-way).

Παρακάτω, παρατίθενται οι **κλάσεις ValueConverter**.

VC	BooleanReverseConverter	<p>Μετατρέπει μία boolean τιμή στο αντίθετό της. Υποστηρίζει two-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε επιστρέφει False.</p>
VC	BooleanToBitConverter	<p>Μετατρέπει μία boolean τιμή σε 0 ή 1. Υποστηρίζει two-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει 1.</p>
VC	BooleanToBrushConverter	<p>Μετατρέπει μία boolean τιμή σε Brush. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει Brushes.Orange.</p>
VC	BooleanToFontSizeConverter	<p>Μετατρέπει μία boolean τιμή σε Double, που δηλώνει το font size. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει 16.0.</p>
VC	BooleanToFontStyleConverter	<p>Μετατρέπει μία boolean τιμή σε FontStyle. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει FontStyles.Italic.</p>
VC	BooleanToFontWeightConverter	<p>Μετατρέπει μία boolean τιμή σε FontWeight. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει FontWeights,Bold.</p>
VC	BooleanToImageSourceConverter	<p>Μετατρέπει μία boolean τιμή σε ImageSource. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει μία συγκεκριμένη εικόνα.</p>

VC	BooleanToStringConverter	<p>Μετατρέπει μία boolean τιμή σε String. Υποστηρίζει two-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει "Ναι".</p>
VC	BooleanToVisibilityConverter	<p>Μετατρέπει μία boolean τιμή σε Visibility. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι True, τότε να επιστρέφει Visibility.Visible.</p>
VC	<p>DateNullableToStringConverter</p> <p><i>(νέο στην έκδοση 1.2)</i></p>	<p>Μετατρέπει μία Nullable(Of Date) ή αλλιώς Date? τιμή σε String. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι Nothing, τότε να επιστρέφει String.Empty.</p>
VC	<p>DateNullableToCurrentDateConverter</p> <p><i>(νέο στην έκδοση 1.2)</i></p>	<p>Μετατρέπει μία Nullable(Of Date) ή αλλιώς Date? τιμή σε String. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι Nothing, τότε να επιστρέφει Date.Today.</p>
VC	<p>ImageSourceToVisibilityConverter</p> <p><i>(νέο στην έκδοση 1.2)</i></p>	<p>Μετατρέπει μία ImageSource τιμή σε Visibility. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι Nothing, τότε να επιστρέφει Visibility.Collapsed.</p>
VC	<p>ImageSourceToStringConverter</p> <p><i>(νέο στην έκδοση 1.2)</i></p>	<p>Μετατρέπει μία ImageSource τιμή σε String. Υποστηρίζει one-way binding mode.</p> <p>Για παράδειγμα, αν η source τιμή είναι Nothing, τότε να επιστρέφει "Δεν υπάρχει εικόνα".</p>

VC	NumberNullableToStringConverter <i>(νέο στην έκδοση 1.3)</i>	Μετατρέπει μία nullable αριθμητική τιμή (Byte, Short, Integer, Single κλπ) σε String . Υποστηρίζει two-way binding mode. Για παράδειγμα, αν η source τιμή είναι Nothing , τότε να επιστρέφει String.Empty .
VC	NumberAsNullableToStringConverter <i>(νέο στην έκδοση 1.3)</i>	Μετατρέπει μία μη nullable αριθμητική τιμή (Byte, Short, Integer, Single κλπ) σε String . Ορίζουμε μία τιμή-φρουρό που θα λειτουργεί ως Nothing μέσω της ιδιότητας NumberAsNothing . Όταν η source τιμή είναι ίση με την τιμή-φρουρό τότε καθορίζουμε αν θα εμφανίζεται κενό ή κάποιο άλλο κείμενο. Υποστηρίζει two-way binding mode. Για παράδειγμα, αν η source τιμή είναι -1 , τότε να επιστρέφει String.Empty .

Παρατηρήσεις

Περισσότερα ValueConverters θα προστίθενται στη βιβλιοθήκη σε μελλοντικές εκδόσεις.

Τί νέο υπάρχει στην έκδοση 1.2

Στην έκδοση 1.2 προστέθηκαν τα εξής **ValueConverters**:

- `DateNullableToStringConverter`
- `DateNullableToCurrentDateConverter`
- `ImageSourceToVisibilityConverter`
- `ImageSourceToStringConverter`

Τί νέο υπάρχει στην έκδοση 1.3

Στην **έκδοση 1.3** προστέθηκαν τα εξής **ValueConverters**:

- `NumberNullableToStringConverter`
- `NumberAsNullableToStringConverter`

Οι ορισμοί των κλάσεων

Οι κλάσεις έχουν οριστεί ως εξής:

- Για την κλάση **BooleanReverseConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Nullable(Of Boolean)))>  
Public Class BooleanReverseConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >
```

```
    <zeus:BooleanReverseConverter x:Key="booleanReverseConverter"/>
```

```
</container.Resources>
```

```
<Element property="{Binding sourcePropertyBooleanType,  
    Mode=OneOrTwoWay,  
    Converter={StaticResource booleanReverseConverter}}" />
```

-
- Για την κλάση **BooleanToBitConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Byte))>  
Public Class BooleanToBitConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >  
    <zeus:BooleanToBitConverter x:Key="booleanToBitConverter"/>  
</container.Resources>  
  
<Element property="{Binding sourcePropertyBooleanType,  
    Mode=OneOrTwoWay,  
    Converter={StaticResource booleanToBitConverter}}" />
```

- **ια την κλάση BooleanToBrushConverter:**

Σύνταξη:

VB:

```
Imports System.Windows.Data  
  
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Brush))>  
Public Class BooleanToBrushConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >  
    <zeus:BooleanToBrushConverter x:Key="booleanToBrushConverter"/>  
</container.Resources>  
  
<Element property="{Binding sourcePropertyBooleanType,
```

```
Converter={StaticResource booleanToBrushConverter}}" />
```

- Για την κλάση **BooleanToFontSizeConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Double))>  
Public Class BooleanToFontSizeConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >
```

```
    <zeus:BooleanToFontSizeConverter x:Key="booleanToFontSizeConverter"/>
```

```
</container.Resources>
```

```
<Element property="{Binding sourcePropertyBooleanType,  
                           Converter={StaticResource booleanToFontSizeConverter}}" />
```

-
- Για την κλάση **BooleanToFontStyleConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(FontStyle))>  
Public Class BooleanToFontStyleConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >  
    <zeus:BooleanToFontStyleConverter x:Key="booleanToFontStyleConverter"/>  
</container.Resources>  
  
<Element property="{Binding sourcePropertyBooleanType,  
    Converter={StaticResource booleanToFontStyleConverter}}" />
```

- Για την κλάση **BooleanToFontWeightConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(FontWeight))>  
Public Class BooleanToFontWeightConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >  
    <zeus:BooleanToFontWeightConverter x:Key="booleanToFontWeightConverter"/>  
</container.Resources>  
  
<Element property="{Binding sourcePropertyBooleanType,  
    Converter={StaticResource booleanToFontWeightConverter}}" />
```

- Για την κλάση **BooleanToImageSourceConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
Imports System.Windows.Media

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(ImageSource))>
Public Class BooleanToImageSourceConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToImageSourceConverter x:Key="booleanToImageSourceConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
    Converter={StaticResource booleanToImageSourceConverter}}" />
```

-
- Για την κλάση **BooleanToStringConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(String))>
Public Class BooleanToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >  
    <zeus:BooleanToStringConverter x:Key="booleanToStringConverter"/>  
  
</container.Resources>  
  
<Element property="{Binding sourcePropertyBooleanType,  
    Mode=OneOrTwoWay,  
    Converter={StaticResource booleanToStringConverter}}" />
```

- Για την κλάση **BooleanToVisibilityConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Visibility))>  
Public Class BooleanToVisibilityConverter  
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >  
    <zeus:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"/>  
  
</container.Resources>  
  
<Element property="{Binding sourcePropertyBooleanType,  
    Converter={StaticResource booleanToVisibilityConverter}}" />
```

- Για την κλάση **DateNullableToStringConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Date)), GetType(String))>
Public Class DateNullableToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:DateNullableToStringConverter x:Key="dateNullableToStringConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyDateNullableType,
    Converter={StaticResource dateNullableToStringConverter}}" />
```

-
- Για την κλάση **DateNullableToCurrentDateConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Date)), GetType(Date))>
Public Class DateNullableToCurrentDateConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:


```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:DateNullableToCurrentDateConverter x:Key="dateNullableToCurrentDateConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyDateNullableType, Converter={StaticResource dateNullableToCurrentDateConverter}}" />
```

- Για την κλάση **ImageSourceToVisibilityConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data
```

```
<ValueConversion(GetType(ImageSource), GetType(Visibility))>
Public Class ImageSourceToVisibilityConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:ImageSourceToVisibilityConverter x:Key="imageSourceToVisibilityConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyImageSourceType, Converter={StaticResource imageSourceToVisibilityConverter}}" />
```

- Για την κλάση **ImageSourceToStringConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(ImageSource), GetType(String))>
Public Class ImageSourceToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:ImageSourceToStringConverter x:Key="imageSourceToStringConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyImageSourceType,
    Converter={StaticResource imageSourceToStringConverter}}"/>
```

-
- Για την κλάση **NumberNullableToStringConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data

Public Class NumberNullableToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >
    <zeus:NumberNullableToStringConverter x:Key="numberNullableToStringConverter"/>
</container.Resources>

<Element property="{Binding sourcePropertyNullableNumberType,
    Converter={StaticResource numberNullableToStringConverter}}" />
```

- Για την κλάση **NumberAsNullableToStringConverter**:

Σύνταξη:

VB:

```
Imports System.Windows.Data

Public Class NumberAsNullableToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >
    <zeus:NumberAsNullableToStringConverter x:Key="numberAsNullableToStringConverter"/>
</container.Resources>

<Element property="{Binding sourcePropertyNumberType,
    Converter={StaticResource numberAsNullableToStringConverter}}" />
```

Διανομή

Κατά τη διανομή, στο φάκελο της εφαρμογής σας πρέπει να αντιγράψετε το **assembly αρχείο ZeusValueConverters.dll**.

Επικοινωνία

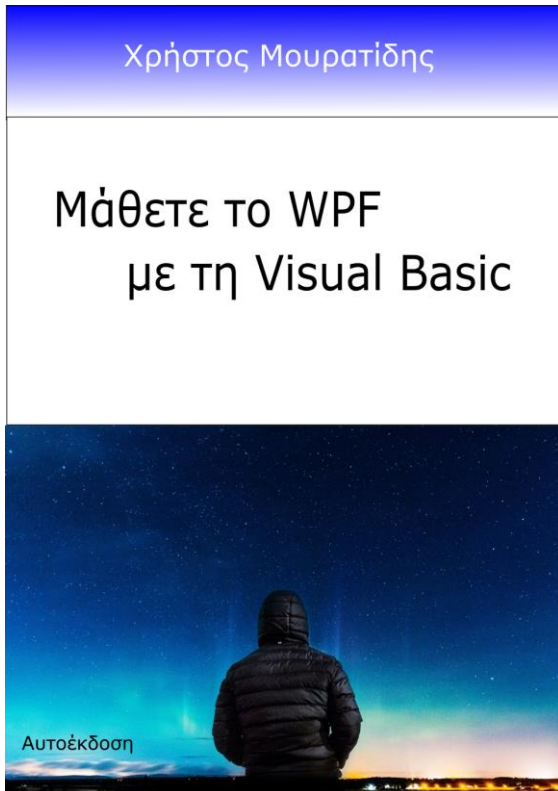
Για οποιαδήποτε πληροφορία ή διευκρίνιση παρακαλώ επικοινωνήστε στο :

mouratx@yahoo.com ή mouratx@hotmail.com



Χρήστος Μουρατίδης,
Πειραιάς, Απρίλιος 2023

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).





BooleanReverseConverter

Μετατρέπει μία **boolean** τιμή στο αντίθετό της. Υποστηρίζει **two-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Nullable(Of Boolean)))>
Public Class BooleanReverseConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanReverseConverter x:Key="booleanReverseConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
    Mode=OneOrTwoWay,
    Converter={StaticResource booleanReverseConverter}}" />
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanReverseConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα IsChecked** του **CheckBox** να γίνεται **False**. Στην σύνδεση με την binding έκφραση, βλέπουμε την αντιστροφή που κάνει το **BooleanReverseConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanReverseConverter x:Key="booleanReverseConverter"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />

    ...

    <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να μην διαγραφεί. -->
    <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

      <TextBlock Text="Να διαγραφεί;" />
      <CheckBox IsChecked="{Binding IsActive, Mode=OneWay,
        Converter={StaticResource booleanReverseConverter }}"
        Margin="15,0,0,0"/>

    </StackPanel>
  </Grid>

</Window>
```



BooleanToBitConverter

Μετατρέπει μία **boolean** τιμή σε **0** ή **1**. Υποστηρίζει **two-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Byte))>
Public Class BooleanToBitConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToBitConverter x:Key="booleanToBitConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
    Mode=OneOrTwoWay,
    Converter={StaticResource booleanToBitConverter}}" />
```

Παρατηρήσεις:

Η κλάση BooleanToBitConverter, εξ' ορισμού μετατρέπει την τιμή True σε 1 και την τιμή False σε 0. Αν θέλουμε να συμβαίνει το αντίθετο, τότε θέτουμε, την **ιδιότητα BitWhenTrue** σε 0 και την **ιδιότητα BitWhenFalse** σε 1.

```
<container.Resources >

    <zeus:BooleanToBitConverter x:Key="booleanToBitConverter"
```



```
BitWhenTrue="0" BitWhenFalse="1" />
```

```
</container.Resources>
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToBitConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται **1**, αλλιώς **0**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToBitConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanToBitConverter x:Key="booleanToBitConverter"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />

    ...

    <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να εμφανίζει την τιμή 1. -->
    <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

      <TextBlock Text="1 ή 0; " />
      <TextBlock Text="{Binding IsActive, Mode=OneWay,
        Converter={StaticResource booleanToBitConverter }}"
        Margin="15,0,0,0" />

    </StackPanel>

  </Grid>

</Window>
```

Ιδιότητες

Όνομα	Περιγραφή
BitWhenTrue	Καθορίζει την τιμή Bit (1 ή 0) όταν η source boolean τιμή είναι True .
BitWhenFalse	Καθορίζει την τιμή Bit (1 ή 0) όταν η source boolean τιμή είναι False .

BitWhenTrue

Καθορίζει την τιμή Bit (1 ή 0) όταν η source boolean τιμή είναι True.

Σύνταξη:

VB:

```
Public Property BitWhenTrue As Byte = 1
```

Τύπος: System.Byte

Προσδιορίζουμε μία τιμή Byte (1 ή 0) που θα αποτελεί την μετατρεπόμενη τιμή όταν η source boolean τιμή είναι True. Η default τιμή είναι 1.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToBitConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται **0**, αλλιώς **1**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToBitConverter** αντικείμενο έχοντας προσδιορίσει τις μετατρεπόμενες τιμές στις **ιδιότητες BitWhenTrue** και **BitWhenFalse**.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToBitConverter x:Key="booleanToBitConverter"
            BitWhenTrue="0" BitWhenFalse="1"/>

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να εμφανίζει την τιμή 0 αλλιώς την 1. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="1 ή 0; " />
    <TextBlock Text="{Binding IsActive, Mode=OneWay,
        Converter={StaticResource booleanToBitConverter }}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

BitWhenFalse

Καθορίζει την τιμή Bit (1 ή 0) όταν η source boolean τιμή είναι True.

Σύνταξη:

VB:

```
Public Property BitWhenFalse As Byte = 0
```

Τύπος: System.Byte

Προσδιορίζουμε μία τιμή Byte (1 ή 0) που θα αποτελεί την μετατρεπόμενη τιμή όταν η source boolean τιμή είναι False. Η default τιμή είναι 0.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToBitConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται **0**, αλλιώς **1**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToBitConverter** αντικείμενο έχοντας προσδιορίσει τις μετατρεπόμενες τιμές στις **ιδιότητες BitWhenTrue** και **BitWhenFalse**.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToBitConverter x:Key="booleanToBitConverter"
            BitWhenTrue="0" BitWhenFalse="1"/>

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να εμφανίζει την τιμή 0 αλλιώς την 1. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="1 ή 0; " />
    <TextBlock Text="{Binding IsActive, Mode=OneWay,
        Converter={StaticResource booleanToBitConverter }}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```



BooleanToBrushConverter

Μετατρέπει μία **boolean** τιμή σε **Brush**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Brush))>
Public Class BooleanToBrushConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToBrushConverter x:Key="booleanToBrushConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
                           Converter={StaticResource booleanToBrushConverter}}" />
```

Παρατηρήσεις:

Η κλάση BooleanToBrushConverter, εξ' ορισμού μετατρέπει την τιμή True σε Brushes.Transparent και την τιμή False επίσης σε Brushes.Transparent. Για να προσδιορίσουμε ποιά Brush θα επιστραφεί για την τιμή True θα θέσουμε κατάλληλα την **ιδιότητα BrushWhenTrue**. Παρόμοια, για να προσδιορίσουμε ποιά Brush θα επιστραφεί για την τιμή False θα θέσουμε κατάλληλα την **ιδιότητα BrushWhenFalse**.

```
<container.Resources >
```

```
<zeus:BooleanToBrushConverter x:Key="booleanToBrushConverter"
    BrushWhenTrue="Green" BrushWhenFalse="Red" />
```

```
</container.Resources>
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToBrushConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Foreground** του **TextBlock** να γίνεται **"Green"**, αλλιώς **"Red"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToBrushConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToBrushConverter x:Key="booleanToBrushForegroundConverter"
            BrushWhenTrue="Green" BrushWhenFalse="Red"/>

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
        να έχει πράσινο χρώμα αλλιώς κόκκινο. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <TextBlock Text="Σήμανση: " />
            <TextBlock Foreground="{Binding IsActive,
                Converter={StaticResource booleanToBrushForegroundConverter}}"
                Margin="15,0,0,0" />

        </StackPanel>

    </Grid>
```



```
</Window>
```

Φυσικά, αν επιθυμούσαμε μία ανάλογη χρωματική σήμανση και για το background θα δημιουργούσαμε άλλο ένα resource BooleanToBrushConverter, για παράδειγμα:

```
<zeus:BooleanToBrushConverter x:Key="booleanToBrushBackgroundConverter"
    BrushWhenTrue="Yellow"
    BrushWhenFalse="Orange"/>
```

και θα θέταμε :

```
<TextBlock Foreground="{Binding IsActive,
    Converter={StaticResource booleanToBrushForegroundConverter}}"

    Background="{Binding IsActive,
    Converter={StaticResource booleanToBrushBackgroundConverter}}"

    Margin="15,0,0,0" />
```

Ιδιότητες

Όνομα	Περιγραφή
BrushWhenTrue	Καθορίζει το αντικείμενο Brush όταν η source boolean τιμή είναι True .
BrushWhenFalse	Καθορίζει το αντικείμενο Brush όταν η source boolean τιμή είναι False .

BrushWhenTrue

Καθορίζει το αντικείμενο **Brush** όταν η **source boolean** τιμή είναι **True**.

Σύνταξη:

VB:

```
Public Property BrushWhenTrue As Brush = Brushes.Transparent
```

Τύπος: **System.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο **Brush** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **Brushes.Transparent**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToBrushConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Foreground** του **TextBlock** να γίνεται **"Green"**, αλλιώς **"Red"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToBrushConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToBrushConverter x:Key="booleanToBrushForegroundConverter"
            BrushWhenTrue="Green"
            BrushWhenFalse="Red"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει πράσινο χρώμα αλλιώς κόκκινο. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Σήμανση: " />
    <TextBlock Foreground="{Binding IsActive,
        Converter={StaticResource booleanToBrushForegroundConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

BrushWhenFalse

Καθορίζει το αντικείμενο **Brush** όταν η **source boolean** τιμή είναι **False**.

Σύνταξη:

VB:

```
Public Property BrushWhenFalse As Brush = Brushes.Transparent
```

Τύπος: **System.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο **Brush** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **False**. Η **default** τιμή είναι **Brushes.Transparent**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToBrushConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Foreground** του **TextBlock** να γίνεται **"Green"**, αλλιώς **"Red"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToBrushConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToBrushConverter x:Key="booleanToBrushForegroundConverter"
            BrushWhenTrue="Green"
            BrushWhenFalse="Red"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει πράσινο χρώμα αλλιώς κόκκινο. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Σήμανση: " />
    <TextBlock Foreground="{Binding IsActive,
        Converter={StaticResource booleanToBrushForegroundConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```



BooleanToFontSizeConverter

Μετατρέπει μία **boolean** τιμή σε **Double**, που δηλώνει το **font size**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Double))>
Public Class BooleanToFontSizeConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToFontSizeConverter x:Key="booleanToFontSizeConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
                          Converter={StaticResource booleanToFontSizeConverter}}" />
```

Παρατηρήσεις:

Η κλάση BooleanToFontSizeConverter, εξ' ορισμού μετατρέπει την τιμή True σε 14.0 και την τιμή False σε 12.0. Για να προσδιορίσουμε ποιά FontSize θα επιστραφεί για την τιμή True θα θέσουμε κατάλληλα την **ιδιότητα FontSizeWhenTrue**. Παρόμοια, για να προσδιορίσουμε ποιά FontSize θα επιστραφεί για την τιμή False θα θέσουμε κατάλληλα την **ιδιότητα FontSizeWhenFalse**.

```

<container.Resources >

    <zeus:BooleanToFontSizeConverter x:Key="booleanToFontSizeConverter"
        FontSizeWhenTrue="16.0"
        FontSizeFalse="14.0" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToFontSizeConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontSize** του **TextBlock** να γίνεται **16.0**, αλλιώς **14.0**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToFontSizeConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToFontSizeConverter x:Key="booleanToFontSizeConverter"
            FontSizeWhenTrue="16.0"
            FontSizeWhenFalse="14.0"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει μεγαλύτερο font size. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <TextBlock Text="Μέγεθος γραμματοσειράς: " />
            <TextBlock FontSize="{Binding IsActive,
                Converter={StaticResource booleanToFontSizeConverter}}"
                Margin="15,0,0,0" />

        </StackPanel>

```


</Grid>

</Window>

Ιδιότητες

Όνομα	Περιγραφή
FontSizeWhenTrue	Καθορίζει το μέγεθος γραμματοσειράς, τύπου Double , όταν η source boolean τιμή είναι True .
FontSizeWhenFalse	Καθορίζει το μέγεθος γραμματοσειράς, τύπου Double , όταν η source boolean τιμή είναι False .

FontSizeWhenTrue

Καθορίζει το μέγεθος γραμματοσειράς, τύπου `Double`, όταν η `source boolean` τιμή είναι `True`.

Σύνταξη:

VB:

```
Public Property FontSizeWhenTrue As Double = 14.0
```

Τύπος: `System.Double`

Προσδιορίζουμε ένα μέγεθος γραμματοσειράς (`FontSize`) που θα αποτελεί την μετατρεπόμενη τιμή όταν η `source boolean` τιμή είναι `True`. Η `default` τιμή είναι `14.0`.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως `resource`, ένα `BooleanToFontSizeConverter` αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του `source` αντικειμένου `Student` είναι η ιδιότητα `IsActive`, τύπου `Boolean`. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η ιδιότητα `IsActive` είναι `True`, τότε η ιδιότητα `FontSize` του `TextBlock` να γίνεται **16.0**, αλλιώς **14.0**. Στην σύνδεση με την `binding` έκφραση, την δουλειά αυτή κάνει το `BooleanToFontSizeConverter` αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanToFontSizeConverter x:Key="booleanToFontSizeConverter"
      FontSizeWhenTrue="16.0"
      FontSizeWhenFalse="14.0"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει μεγαλύτερο font size. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Μέγεθος γραμματοσειράς: " />
    <TextBlock FontSize="{Binding IsActive,
        Converter={StaticResource booleanToFontSizeConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

FontSizeWhenFalse

Καθορίζει το μέγεθος γραμματοσειράς, τύπου **Double**, όταν η **source boolean** τιμή είναι **False**.

Σύνταξη:

VB:

```
Public Property FontSizeWhenFalse As Double = 12.0
```

Τύπος: **System.Double**

Προσδιορίζουμε ένα μέγεθος γραμματοσειράς (**FontSize**) που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **12.0**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToFontSizeConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontSize** του **TextBlock** να γίνεται **16.0**, αλλιώς **14.0**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToFontSizeConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanToFontSizeConverter x:Key="booleanToFontSizeConverter"
      FontSizeWhenTrue="16.0"
      FontSizeWhenFalse="14.0"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει μεγαλύτερο font size. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Μέγεθος γραμματοσειράς: " />
    <TextBlock FontSize="{Binding IsActive,
        Converter={StaticResource booleanToFontSizeConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```



BooleanToFontStyleConverter

Μετατρέπει μία **boolean** τιμή σε **FontStyle**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(FontStyle))>
Public Class BooleanToFontStyleConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToFontStyleConverter x:Key="booleanToFontStyleConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
                          Converter={StaticResource booleanToFontStyleConverter}}"/>
```

Παρατηρήσεις:

Η κλάση BooleanToFontStyleConverter, εξ' ορισμού μετατρέπει την τιμή True σε FontStyles.Italic και την τιμή False σε FontStyles.Normal. Για να προσδιορίσουμε ποιά FontStyle θα επιστραφεί για την τιμή True θα θέσουμε κατάλληλα την **ιδιότητα FontStyleWhenTrue**. Παρόμοια, για να προσδιορίσουμε ποιά FontStyle θα επιστραφεί για την τιμή False θα θέσουμε κατάλληλα την **ιδιότητα FontStyleWhenFalse**.

```
<container.Resources >
```

```

<zeus:BooleanToFontStyleConverter x:Key="booleanToFontStyleConverter"
    FontStyleWhenTrue="Oblique"
    FontStyleFalse="Normal" />

</container.Resources>

```

Σε γενικές γραμμές, δεν χρειάζεται να προσδιορίσουμε αυτές τις ιδιότητες αφού τα font styles είναι περιορισμένα: Italic, Oblique και Normal. Συνεπώς, οι default τιμές τους είναι αρκετές.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToFontStyleConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontStyle** του **TextBlock** να γίνεται **"Oblique"**, αλλιώς **"Normal"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToFontStyleConverter** αντικείμενο. Να σημειώσουμε ότι την ιδιότητα FontStyleWhenFalse θα μπορούσαμε να την παραλείψουμε αφού ήδη η default τιμή είναι "Normal".

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToFontStyleConverter x:Key="booleanToFontStyleConverter"
            FontStyleWhenTrue="Oblique"
            FontStyleWhenFalse="Normal"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει font style "Oblique". -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

```



```
<TextBlock Text="Στυλ κειμένου: " />
<TextBlock FontStyle="{Binding IsActive,
Converter={StaticResource booleanToFontStyleConverter}}"
Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

Ιδιότητες

Όνομα	Περιγραφή
FontStyleWhenTrue	Καθορίζει το FontStyle όταν η source boolean τιμή είναι True .
FontStyleWhenFalse	Καθορίζει το FontStyle όταν η source boolean τιμή είναι False .

FontStyleWhenTrue

Καθορίζει το **FontStyle** όταν η **source boolean** τιμή είναι **True**.

Σύνταξη:

VB:

```
Public Property FontStyleWhenTrue As FontStyle = FontStyles.Italic
```

Τύπος: **System.Windows.FontStyle**

Προσδιορίζουμε ένα **FontStyle** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **FontStyles.Italic**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToFontStyleConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontStyle** του **TextBlock** να γίνεται **"Oblique"**, αλλιώς **"Normal"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToFontStyleConverter** αντικείμενο. Να σημειώσουμε ότι την ιδιότητα **FontStyleWhenFalse** θα μπορούσαμε να την παραλείψουμε αφού ήδη η **default** τιμή είναι **"Normal"**.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToFontStyleConverter x:Key="booleanToFontStyleConverter"
            FontStyleWhenTrue="Oblique"
            FontStyleWhenFalse="Normal"/>

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">
```

```
<!-- Title-->
<TextBlock Text="Μαθητής" />

...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει font style "Oblique". -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Στυλ κειμένου: " />
    <TextBlock FontStyle="{Binding IsActive,
        Converter={StaticResource booleanToFontStyleConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

FontStyleWhenFalse

Καθορίζει το **FontStyle** όταν η **source boolean** τιμή είναι **False**.

Σύνταξη:

VB:

```
Public Property FontStyleWhenFalse As FontStyle = FontStyles.Normal
```

Τύπος: **System.Windows.FontStyle**

Προσδιορίζουμε ένα **FontStyle** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **False**. Η **default** τιμή είναι **FontStyles.Normal**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToFontStyleConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontStyle** του **TextBlock** να γίνεται **"Oblique"**, αλλιώς **"Normal"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToFontStyleConverter** αντικείμενο. Να σημειώσουμε ότι την ιδιότητα **FontStyleWhenFalse** θα μπορούσαμε να την παραλείψουμε αφού ήδη η **default** τιμή είναι **"Normal"**.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanToFontStyleConverter x:Key="booleanToFontStyleConverter"
      FontStyleWhenTrue="Oblique"
      FontStyleWhenFalse="Normal"/>

  </Window.Resources >

  <Grid Margin="10" Name="grdStudent">
```

```
<!-- Title-->
<TextBlock Text="Μαθητής" />

...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει font style "Oblique". -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Στυλ κειμένου: " />
    <TextBlock FontStyle="{Binding IsActive,
        Converter={StaticResource booleanToFontStyleConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```



BooleanToFontWeightConverter

Μετατρέπει μία **boolean** τιμή σε **FontWeight**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(FontWeight))>
Public Class BooleanToFontWeightConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToFontWeightConverter x:Key="booleanToFontWeightConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
                          Converter={StaticResource booleanToFontWeightConverter}}" />
```

Παρατηρήσεις:

Η κλάση BooleanToFontWeightConverter, εξ' ορισμού μετατρέπει την τιμή True σε FontWeights.Bold και την τιμή False σε FontWeights.Normal. Για να προσδιορίσουμε ποιά FontWeight θα επιστραφεί για την τιμή True θα θέσουμε κατάλληλα την **ιδιότητα FontWeightWhenTrue**. Παρόμοια, για να προσδιορίσουμε ποιά FontWeight θα επιστραφεί για την τιμή False θα θέσουμε κατάλληλα την **ιδιότητα FontWeightWhenFalse**.

```
<container.Resources >
```

```

<zeus:BooleanToFontWeightConverter x:Key="booleanToFontWeightConverter"
    FontWeightWhenTrue="ExtraBold"
    FontWeightFalse="ExtraLight" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToFontWeightConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontWeight** του **TextBlock** να γίνεται **"ExtraBold"**, αλλιώς **"ExtraLight"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToFontWeightConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToFontWeightConverter x:Key="booleanToFontWeightConverter"
            FontWeightWhenTrue="ExtraBold"
            FontWeightWhenFalse="ExtraLight"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει πολύ έντονη γραφή αλλιώς πολύ λεπτή γραφή. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <TextBlock Text="Ένταση γραφής κειμένου: " />
            <TextBlock FontWeight="{ Binding IsActive,
                Converter={StaticResource booleanToFontWeightConverter}}"
                Margin="15,0,0,0" />

        </StackPanel>

```


</Grid>

</Window>

Ιδιότητες

Όνομα	Περιγραφή
FontWeightWhenTrue	Καθορίζει το FontWeight όταν η source boolean τιμή είναι True .
FontWeightWhenFalse	Καθορίζει το FontWeight όταν η source boolean τιμή είναι False .

FontWeightWhenTrue

Καθορίζει το **FontWeight** όταν η **source boolean** τιμή είναι **True**.

Σύνταξη:

VB:

```
Public Property FontWeightWhenTrue As FontWeight = FontWeights.Bold
```

Τύπος: **System.Windows.FontWeight**

Προσδιορίζουμε ένα **FontWeight** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **FontWeights.Bold**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToFontWeightConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontWeight** του **TextBlock** να γίνεται **"ExtraBold"**, αλλιώς **"ExtraLight"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToFontWeightConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToFontWeightConverter x:Key="booleanToFontWeightConverter"
            FontWeightWhenTrue="ExtraBold"
            FontWeightWhenFalse="ExtraLight"/>

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει πολύ έντονη γραφή αλλιώς πολύ λεπτή γραφή. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Ένταση γραφής κειμένου: " />
    <TextBlock FontWeight="{ Binding IsActive,
        Converter={StaticResource booleanToFontWeightConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

FontWeightWhenFalse

Καθορίζει το **FontWeight** όταν η **source boolean** τιμή είναι **False**.

Σύνταξη:

VB:

```
Public Property FontWeightWhenFalse As FontWeight = FontWeights.Normal
```

Τύπος: **System.Windows.FontWeight**

Προσδιορίζουμε ένα **FontWeight** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **FontWeights.Normal**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToFontWeightConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα FontWeight** του **TextBlock** να γίνεται **"ExtraBold"**, αλλιώς **"ExtraLight"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToFontWeightConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanToFontWeightConverter x:Key="booleanToFontWeightConverter"
      FontWeightWhenTrue="ExtraBold"
      FontWeightWhenFalse="ExtraLight"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
```

```
<TextBlock Text="Μαθητής" />

...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να έχει πολύ έντονη γραφή αλλιώς πολύ λεπτή γραφή. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Ένταση γραφής κειμένου: " />
    <TextBlock FontWeight="{ Binding IsActive,
        Converter={StaticResource booleanToFontWeightConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```



BooleanToImageSourceConverter

Μετατρέπει μία **boolean** τιμή σε **ImageSource**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data
Imports System.Windows.Media

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(ImageSource))>
Public Class BooleanToImageSourceConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToImageSourceConverter x:Key="booleanToImageSourceConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
    Converter={StaticResource booleanToImageSourceConverter}}" />
```

Παρατηρήσεις:

Η κλάση `BooleanToImageSourceConverter`, εξ' ορισμού μετατρέπει την τιμή `True` σε `Nothing`, όπως και για την τιμή `False`. Για να προσδιορίσουμε ποιό `ImageSource` επιστραφεί για την τιμή `True` θα θέσουμε κατάλληλα την ιδιότητα **`ImageSourceWhenTrue`**. Παρόμοια, για να προσδιορίσουμε ποιό `ImageSource` θα επιστραφεί για την τιμή `False` θα θέσουμε κατάλληλα την ιδιότητα **`ImageSourceWhenFalse`**.

```
<container.Resources >
```

```
<zeus:BooleanToImageSourceConverter x:Key="booleanToImageSourceConverter"
    ImageSourceWhenTrue="Images/Happy.png"
    ImageSourceFalse="Images/Unhappy.png" />
```

```
</container.Resources>
```

Υποθέτουμε ότι στην εφαρμογή έχουμε προσθέσει έναν φάκελο με όνομα *Images* κι έχουμε τοποθετήσει δύο εικόνες: την *Happy.png* και την *Unhappy.png*. Έτσι, οι παραπάνω τιμές στις ιδιότητες είναι το σχετικό *Uri* προς αυτά τα αρχεία.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα `BooleanToImageSourceConverter` αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου `Student` είναι η ιδιότητα `IsActive`, τύπου `Boolean`. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα `IsActive`** είναι **`True`**, τότε η **ιδιότητα `Source`** του **`Image`** να γίνεται **`"Images/Happy.png"`** (χαρούμενο πρόσωπο), αλλιώς **`"Images/Unhappy.png"`** (λυπημένο πρόσωπο). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **`BooleanToImageSourceConverter`** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToImageSourceConverter x:Key="booleanToImageSourceConverter"
            ImageSourceWhenTrue="Images/Happy.png"
            ImageSourceWhenFalse="Images/Unhappy.png" />

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
        τότε να εμφανίζει ένα χαρούμενο προσωπάκι αλλιώς ένα λυπημένο. -->
```



```
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >
    <TextBlock Text="Γραφικό: " />
    <Viewbox Width="50" Height="50" Stretch="Fill" >
        <Image Source="{Binding IsActive,
            Converter={StaticResource booleanToImageSourceConverter}}" />
    </Viewbox>
</StackPanel>
</Grid>
</Window>
```

- Υποθέτουμε ότι στην εφαρμογή έχουμε προσθέσει έναν φάκελο με όνομα Images κι έχουμε τοποθετήσει δύο εικόνες: την Happy.png και την Unhappy.png. Έτσι, οι παραπάνω τιμές στις ιδιότητες είναι το σχετικό Uri προς αυτά τα αρχεία.

Ιδιότητες

Όνομα	Περιγραφή
ImageSourceWhenTrue	Καθορίζει το ImageSource όταν η source boolean τιμή είναι True .
ImageSourceWhenFalse	Καθορίζει το ImageSource όταν η source boolean τιμή είναι False .

ImageSourceWhenTrue

Καθορίζει το `ImageSource` όταν η `source` boolean τιμή είναι `True`.

Σύνταξη:

VB:

```
Public Property ImageSourceWhenTrue As ImageSource = Nothing
```

Τύπος: `System.Windows.Media.ImageSource`

Προσδιορίζουμε ένα `ImageSource` που θα αποτελεί την μετατρεπόμενη τιμή όταν η `source` boolean τιμή είναι `True`. Η default τιμή είναι `Nothing`.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα `BooleanToImageSourceConverter` αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του `source` αντικειμένου `Student` είναι η ιδιότητα `IsActive`, τύπου `Boolean`. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η ιδιότητα `IsActive` είναι `True`, τότε η ιδιότητα `Source` του `Image` να γίνεται `"Images/Happy.png"` (χαρούμενο πρόσωπο), αλλιώς `"Images/Unhappy.png"` (λυπημένο πρόσωπο). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το `BooleanToImageSourceConverter` αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToImageSourceConverter x:Key="booleanToImageSourceConverter"
            ImageSourceWhenTrue="Images/Happy.png"
            ImageSourceWhenFalse="Images/Unhappy.png"/>

    </Window.Resources >

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

...

```
<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
τότε να εμφανίζει ένα χαρούμενο προσωπάκι αλλιώς ένα λυπημένο. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Γραφικό: " />
    <Viewbox Width="50" Height="50" Stretch="Fill" >
        <Image Source="{Binding IsActive,
            Converter={StaticResource booleanToImageSourceConverter}}" />
    </Viewbox>

</StackPanel>

</Grid>

</Window>
```

- Υποθέτουμε ότι στην εφαρμογή έχουμε προσθέσει έναν φάκελο με όνομα Images κι έχουμε τοποθετήσει δύο εικόνες: την Happy.png και την Unhappy.png. Έτσι, οι παραπάνω τιμές στις ιδιότητες είναι το σχετικό Uri προς αυτά τα αρχεία.

ImageSourceWhenFalse

Καθορίζει το `ImageSource` όταν η `source` boolean τιμή είναι `False`.

Σύνταξη:

VB:

```
Public Property ImageSourceWhenFalse As ImageSource = Nothing
```

Τύπος: `System.Windows.Media.ImageSource`

Προσδιορίζουμε ένα `ImageSource` που θα αποτελεί την μετατρεπόμενη τιμή όταν η `source` boolean τιμή είναι `False`. Η default τιμή είναι `Nothing`.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα `BooleanToImageSourceConverter` αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του `source` αντικειμένου `Student` είναι η ιδιότητα `IsActive`, τύπου `Boolean`. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η ιδιότητα `IsActive` είναι `True`, τότε η ιδιότητα `Source` του `Image` να γίνεται `"Images/Happy.png"` (χαρούμενο πρόσωπο), αλλιώς `"Images/Unhappy.png"` (λυπημένο πρόσωπο). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το `BooleanToImageSourceConverter` αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToImageSourceConverter x:Key="booleanToImageSourceConverter"
            ImageSourceWhenTrue="Images/Happy.png"
            ImageSourceWhenFalse="Images/Unhappy.png"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

...

```
<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
τότε να εμφανίζει ένα χαρούμενο προσωπάκι αλλιώς ένα λυπημένο. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Γραφικό: " />
    <Viewbox Width="50" Height="50" Stretch="Fill" >
        <Image Source="{Binding IsActive,
            Converter={StaticResource booleanToImageSourceConverter}}" />
    </Viewbox>

</StackPanel>

</Grid>

</Window>
```

- Υποθέτουμε ότι στην εφαρμογή έχουμε προσθέσει έναν φάκελο με όνομα Images κι έχουμε τοποθετήσει δύο εικόνες: την Happy.png και την Unhappy.png. Έτσι, οι παραπάνω τιμές στις ιδιότητες είναι το σχετικό Uri προς αυτά τα αρχεία.



BooleanToStringConverter

Μετατρέπει μία **boolean** τιμή σε **String**. Υποστηρίζει **two-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(String))>
Public Class BooleanToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToStringConverter x:Key="booleanToStringConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
    Mode=OneOrTwoWay,
    Converter={StaticResource booleanToStringConverter}}" />
```

Παρατηρήσεις:

Η κλάση BooleanToStringConverter, εξ' ορισμού μετατρέπει την τιμή True σε "Ναι" και την τιμή False σε "Όχι". Για να προσδιορίσουμε ποιό String θα επιστραφεί για την τιμή True θα θέσουμε κατάλληλα την **ιδιότητα StringWhenTrue**. Παρόμοια, για να προσδιορίσουμε ποιό String θα επιστραφεί για την τιμή False θα θέσουμε κατάλληλα την **ιδιότητα StringWhenFalse**.

```
<container.Resources >
```

```
<zeus:BooleanToStringConverter x:Key="booleanToStringConverter"
    StringWhenTrue="Ισχύει"
    StringFalse="Δεν ισχύει" />
```

```
</container.Resources>
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToStringConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται **"Είναι ενεργός"**, αλλιώς **"Έχει διακόψει"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToStringConverter x:Key="booleanToStringConverter"
            StringWhenTrue="Είναι ενεργός"
            StringWhenFalse="Έχει διακόψει"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να επιστρέφει το "Είναι ενεργός" αλλιώς "Έχει διακόψει". -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <TextBlock Text="Κατάσταση μαθητή: " />
            <TextBlock Text="{Binding IsActive,
                Converter={StaticResource booleanToStringConverter}}"
                Margin="15,0,0,0" />

        </StackPanel>
```


</Grid>

</Window>

Ιδιότητες

Όνομα	Περιγραφή
StringWhenTrue	Καθορίζει το String όταν η source boolean τιμή είναι True .
StringWhenFalse	Καθορίζει το String όταν η source boolean τιμή είναι False .

StringWhenTrue

Καθορίζει το **String** όταν η **source boolean** τιμή είναι **True**.

Σύνταξη:

VB:

```
Public Property StringWhenTrue As String = "Ναι"
```

Τύπος: **System.String**

Προσδιορίζουμε ένα **String** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **"Ναι"**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται **"v"**, αλλιώς **"x"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
  WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:BooleanToStringConverter x:Key="booleanToStringConverter"
      StringWhenTrue="v"
      StringWhenFalse="x"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να επιστρέφει το "v" αλλιώς "x". -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Είναι ενεργός; " />
    <TextBlock Text="{Binding IsActive,
        Converter={StaticResource booleanToStringConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

StringWhenFalse

Καθορίζει το **String** όταν η **source boolean** τιμή είναι **False**.

Σύνταξη:

VB:

```
Public Property StringWhenFalse As String = "Όχι"
```

Τύπος: System.String

Προσδιορίζουμε ένα String που θα αποτελεί την μετατρεπόμενη τιμή όταν η source boolean τιμή είναι False. Η default τιμή είναι "Όχι".

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToStringConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται "v", αλλιώς "x". Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToStringConverter x:Key="booleanToStringConverter"
            StringWhenTrue="v"
            StringWhenFalse="x"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να επιστρέφει το "v" αλλιώς "x". -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Είναι ενεργός; " />
    <TextBlock Text="{Binding IsActive,
        Converter={StaticResource booleanToStringConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```



BooleanToVisibilityConverter

Μετατρέπει μία **boolean** τιμή σε **Visibility**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Boolean)), GetType(Visibility))>
Public Class BooleanToVisibilityConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyBooleanType,
    Converter={StaticResource booleanToVisibilityConverter}}" />
```

Παρατηρήσεις:

Η κλάση BooleanToVisibilityConverter, εξ' ορισμού μετατρέπει την τιμή True σε Visibility.Visible και την τιμή False σε Visibility.Collapsed. Για να προσδιορίσουμε ποιά String θα επιστραφεί για την τιμή True θα θέσουμε κατάλληλα την **ιδιότητα VisibilityWhenTrue**. Παρόμοια, για να προσδιορίσουμε ποιά String θα επιστραφεί για την τιμή False θα θέσουμε κατάλληλα την **ιδιότητα VisibilityWhenFalse**.

```

<container.Resources >

    <zeus:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"
        VisibilityWhenTrue="Visible"
        VisibilityWhenFalse="Hidden" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα BooleanToVisibilityConverter αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα IsActive, τύπου Boolean. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Visibility** του **TextBlock** να γίνεται **"Visible"**, αλλιώς **"Hidden"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **BooleanToVisibilityConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"
            VisibilityWhenTrue="Visible"
            VisibilityWhenFalse="Hidden"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να εμφανίζεται αλλιώς να αποκρύπτεται. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <TextBlock Text="Εμφάνιση στοιχείων μαθητή: " />
            <TextBlock Visibility="{ Binding IsActive,
                Converter={StaticResource booleanToVisibilityConverter}}"
                Margin="15,0,0,0" />

        </StackPanel>

```


</Grid>

</Window>

Ιδιότητες

Όνομα	Περιγραφή
VisibilityWhenTrue	Καθορίζει την Visibility όταν η source boolean τιμή είναι True .
VisibilityWhenFalse	Καθορίζει την Visibility όταν η source boolean τιμή είναι False .

VisibilityWhenTrue

Καθορίζει την **Visibility** όταν η **source boolean** τιμή είναι **True**.

Σύνταξη:

VB:

```
Public Property VisibilityWhenTrue As Visibility = Visibility.Visible
```

Τύπος: **System.Windows.Visibility**

Προσδιορίζουμε την **Visibility** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **Visibility.Visible**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToVisibilityConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Visibility** του **TextBlock** να γίνεται **"Visible"**, αλλιώς **"Hidden"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToVisibilityConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"
            VisibilityWhenTrue="Visible"
            VisibilityWhenFalse="Hidden"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...

<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός
να εμφανίζεται αλλιώς να αποκρύπτεται. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Εμφάνιση στοιχείων μαθητή: " />
    <TextBlock Visibility="{ Binding IsActive,
        Converter={StaticResource booleanToVisibilityConverter}}"
        Margin="15,0,0,0" />

</StackPanel>

</Grid>

</Window>
```

- Στο resource BooleanToVisibilityConverter αντικείμενο, θα μπορούσαμε να παραλείψουμε την ιδιότητα VisibilityWhenTrue αφού η default τιμή είναι Visibility.Visible.

VisibilityWhenFalse

Καθορίζει την **Visibility** όταν η **source boolean** τιμή είναι **False**.

Σύνταξη:

VB:

```
Public Property VisibilityWhenFalse As Visibility = Visibility.Collapsed
```

Τύπος: **System.Windows.Visibility**

Προσδιορίζουμε την **Visibility** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source boolean** τιμή είναι **True**. Η **default** τιμή είναι **Visibility.Collapsed**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **BooleanToVisibilityConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **IsActive**, τύπου **Boolean**. Αυτή η ιδιότητα υποδηλώνει ότι ο μαθητής είναι ενεργός (δεν έχει διακόψει τη φοίτησή του). Θέλουμε όταν η **ιδιότητα IsActive** είναι **True**, τότε η **ιδιότητα Visibility** του **TextBlock** να γίνεται **"Visible"**, αλλιώς **"Hidden"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **BooleanToVisibilityConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project" Height="467.775" Width="799.064"
    WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"
            VisibilityWhenTrue="Visible"
            VisibilityWhenFalse="Hidden"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```
...  
  
<!-- Ενεργός μαθητής. Source ιδιότητα IsActive. Αν ο μαθητής είναι ενεργός  
να εμφανίζεται αλλιώς να αποκρύπτεται. -->  
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >  
  
    <TextBlock Text="Εμφάνιση στοιχείων μαθητή: " />  
    <TextBlock Visibility="{ Binding IsActive,  
        Converter={StaticResource booleanToVisibilityConverter}}"  
        Margin="15,0,0,0" />  
  
</StackPanel>  
  
</Grid>  
  
</Window>
```

- Στο resource BooleanToVisibilityConverter αντικείμενο, θα μπορούσαμε να παραλείψουμε την ιδιότητα VisibilityWhenTrue αφού η default τιμή είναι Visibility.Visible.



DateNullableToStringConverter

Μετατρέπει μία **Nullable(Of Date)** ή αλλιώς **Date?** τιμή σε **String**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Date)), GetType(String))>
Public Class DateNullableToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:DateNullableToStringConverter x:Key="dateNullableToStringConverter"/>

</container.Resources>

<Element
property="{Binding sourcePropertyDateNullableType, Mode=OneWay, Converter={StaticResource dateNullableToStringConverter}}" />
```

Παρατηρήσεις:

Η κλάση `DateNullableToStringConverter`, εξ' ορισμού μετατρέπει την τιμή `Nothing` ή `Date.MinValue` σε `String.Empty`. Για να προσδιορίσουμε ποιά `String` θα επιστραφεί για την τιμή `Nothing` θα θέσουμε κατάλληλα την **ιδιότητα `TextWhenNothing`**.

```
<container.Resources >
```

```

<zeus:DateNullableToStringConverter x:Key="dateNullableToStringConverter"
    TextWhenNothing="Απαιτείται ημερομηνία" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **DateNullableToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικείμενου Student είναι η ιδιότητα **RegistrationDate**, τύπου **Nullable(of Date)**. Αυτή η ιδιότητα καθορίζει την ημερομηνία εγγραφής του μαθητή. Θέλουμε όταν η **ιδιότητα RegistrationDate** είναι **Nothing**, τότε η **ιδιότητα Text** του **TextBlock** να γίνεται **"Απαιτείται ημερομηνία"** (φυσικά, αν δεν είναι **Nothing** τότε επιστρέφει την *date string*). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **DateNullableToStringConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
        nverters"

    mc:Ignorable="d"
    Title="ValueConverters Sample Project"
    Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:DateNullableToStringConverter x:Key="dateNullableToStringConverter"
            TextWhenNothing="Απαιτείται ημερομηνία" />

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Source ιδιότητα RegistrationDate . -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <TextBlock Text="Ημερομηνία εγγραφής: " />
            <TextBlock Text="{Binding RegistrationDate,
                StringFormat=d, ConverterCulture=el,
                Converter={StaticResource dateNullableToStringConverter}}"
                Margin="15,0,0,0" />

        </StackPanel>

    </Grid>

```


</Window>

Ιδιότητες

Όνομα	Περιγραφή
TextWhenNothing	Καθορίζει το String όταν η source Date? τιμή είναι Nothing .

TextWhenNothing

Καθορίζει το **String** όταν η source **Date?** τιμή είναι **Nothing**.

Σύνταξη:

VB:

```
Public Property TextWhenNothing As String = String.Empty
```

Τύπος: System.String

Προσδιορίζουμε ένα **String** που θα αποτελεί την μετατρεπόμενη τιμή όταν η source **Date?** τιμή είναι **Nothing**. Η default τιμή είναι **String.Empty**.



DateNullableToCurrentDateConverter

Μετατρέπει μία **Nullable(Of Date)** ή αλλιώς **Date?** τιμή σε **Date.Today**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(Nullable(Of Date)), GetType(Date))>
Public Class DateNullableToCurrentDateConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:DateNullableToCurrentDateConverter x:Key="dateNullableToCurrentDateConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyDateNullableType, Mode=OneWay, Converter={StaticResource dateNullableToCurrentDateConverter}}" />
```

Παρατηρήσεις:

Η κλάση `DateNullableToCurrentDateConverter` μετατρέπει την τιμή `Nothing` ή `Date.MinValue` σε `Date.Today`.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **DateNullableToCurrentDateConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα RegistrationDate, τύπου Nullable(of Date). Αυτή η ιδιότητα καθορίζει την ημερομηνία εγγραφής του μαθητή. Θέλουμε όταν η **ιδιότητα RegistrationDate** είναι **Nothing**, τότε η **ιδιότητα DisplayDate** του **DatePicker** να εμφανίζει την τρέχουσα ημερομηνία (*φυσικά, αν δεν είναι Nothing τότε επιστρέφει την date value*). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **DateNullableToCurrentDateConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:DateNullableToCurrentDateConverter
      x:Key="dateNullableToCurrentDateConverter" />

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />

    ...

    <!-- Source ιδιότητα RegistrationDate . -->
    <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

      <TextBlock Text="Ημερομηνία εγγραφής: " />
      <DatePicker SelectedDate="{Binding RegistrationDate, Mode=TwoWay}"
        DisplayDate="{Binding RegistrationDate, Mode=OneWay
          Converter={StaticResource dateNullableToCurrentDateConverter}}"
        Margin="15,0,0,0" />

    </StackPanel>

  </Grid>

</Window>
```



ImageSourceToVisibilityConverter

Μετατρέπει μία **ImageSource** τιμή σε **Visibility**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(ImageSource), GetType(Visibility))>
Public Class ImageSourceToVisibilityConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:ImageSourceToVisibilityConverter
        x:Key="imageSourceToVisibilityConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyImageSourceType,
    Converter={StaticResource imageSourceToVisibilityConverter}}" />
```

Παρατηρήσεις:

Η κλάση ImageSourceToVisibilityConverter, εξ' ορισμού μετατρέπει την τιμή Nothing σε Visibility.Collapsed και όταν έχει όντως source σε Visibility.Visible. Για να προσδιορίσουμε ποιά Visibility θα επιστραφεί για την τιμή Nothing θα θέσουμε κατάλληλα την **ιδιότητα VisibilityWhenHasNothing**. Παρόμοια, για να προσδιορίσουμε ποιά Visibility θα επιστραφεί όταν έχει source θα θέσουμε κατάλληλα την **ιδιότητα VisibilityWhenHasSource**.

```

<container.Resources >

    <zeus:ImageSourceToVisibilityConverter
        x:Key="imageSourceToVisibilityConverter"
        VisibilityWhenHasSource="Visible"
        VisibilityWhenHasNothing="Hidden" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **ImageSourceToVisibilityConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα Photo, τύπου ImageSource. Αυτή η ιδιότητα περιέχει τα δεδομένα της εικόνας του μαθητή. Θέλουμε όταν η **ιδιότητα Photo** έχει δεδομένα, τότε η **ιδιότητα Visibility** του **TextBlock** να γίνεται **"Visible"**, **αλλιώς "Collapsed"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **ImageSourceToVisibilityConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project"
    Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:ImageSourceToVisibilityConverter
            x:Key="imageSourceToVisibilityConverter"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Source ιδιότητα Photo. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <!-- Το TextBlock θα εμφανίζεται μόνο όταν υπάρχει source στην ιδιότητα
            Photo του student object -->
            <TextBlock Text="Φωτό:" Style="{StaticResource RowLabelsStyle }"
                HorizontalAlignment="Center"
                Visibility="{Binding Photo, Mode=OneWay,
                Converter={StaticResource imageSourceToVisibilityConverter} }"/>

```

```
        <Image Source="{Binding Photo, Mode=OneWay}" Margin="5,15,5,5"/>
    </StackPanel>
</Grid>
</Window>
```


Ιδιότητες

Όνομα	Περιγραφή
VisibilityWhenHasSource	Καθορίζει την Visibility όταν η source ImageSource τιμή έχει δεδομένα .
VisibilityWhenHasNothing	Καθορίζει την Visibility όταν η source ImageSource τιμή είναι Nothing .

VisibilityWhenHasSource

Καθορίζει την **Visibility** όταν η **source ImageSource** τιμή έχει δεδομένα.

Σύνταξη:

VB:

```
Public Property VisibilityWhenHasSource As Visibility = Visibility.Visible
```

Τύπος: **System.Windows.Visibility**

Προσδιορίζουμε την **Visibility** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source ImageSource** τιμή έχει δεδομένα. Η **default** τιμή είναι **Visibility.Visible**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **ImageSourceToVisibilityConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικείμενου **Student** είναι η ιδιότητα **Photo**, τύπου **ImageSource**. Αυτή η ιδιότητα περιέχει τα δεδομένα της εικόνας του μαθητή. Θέλουμε όταν η **ιδιότητα Photo** έχει δεδομένα, τότε η **ιδιότητα Visibility** του **TextBlock** να γίνεται **"Visible"**, αλλιώς **"Collapsed"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **ImageSourceToVisibilityConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:ImageSourceToVisibilityConverter
            x:Key="imageSourceToVisibilityConverter"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...
```

```

<!-- Source ιδιότητα Photo. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <!-- Το TextBlock θα εμφανίζεται μόνο όταν υπάρχει source στην ιδιότητα
    Photo του student object -->
    <TextBlock Text="Φωτό:" Style="{StaticResource RowLabelsStyle }"
    HorizontalAlignment="Center"
    Visibility="{Binding Photo, Mode=OneWay,
    Converter={StaticResource imageSourceToVisibilityConverter} }"/>

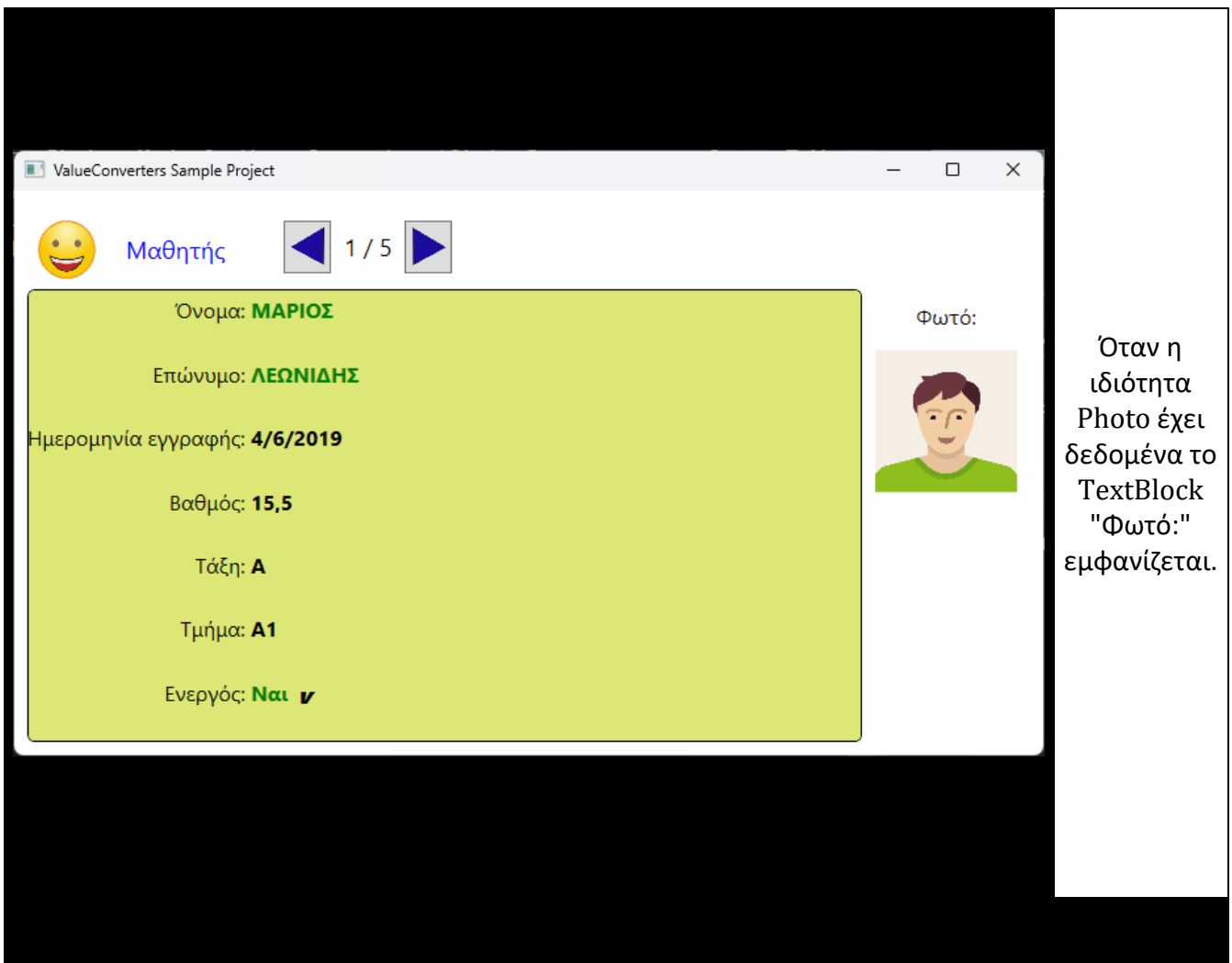
    <Image Source="{Binding Photo, Mode=OneWay}" Margin="5,15,5,5"/>

</StackPanel>

</Grid>

</Window>

```



Όταν η ιδιότητα Photo δεν έχει δεδομένα το TextBlock "Φωτό:" δεν εμφανίζεται.

VisibilityWhenHasNothing

Καθορίζει την **Visibility** όταν η source **ImageSource** τιμή δεν έχει δεδομένα (τιμή **Nothing**).

Σύνταξη:

VB:

```
Public Property VisibilityWhenHasNothing As Visibility = Visibility.Collapsed
```

Τύπος: **System.Windows.Visibility**

Προσδιορίζουμε την **Visibility** που θα αποτελεί την μετατρεπόμενη τιμή όταν η source **ImageSource** τιμή δεν έχει δεδομένα. Η default τιμή είναι **Visibility.Collapsed**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **ImageSourceToVisibilityConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικείμενου **Student** είναι η ιδιότητα **Photo**, τύπου **ImageSource**. Αυτή η ιδιότητα περιέχει τα δεδομένα της εικόνας του μαθητή. Θέλουμε όταν η **ιδιότητα Photo** έχει δεδομένα, τότε η **ιδιότητα Visibility** του **TextBlock** να γίνεται **"Visible"**, αλλιώς **"Collapsed"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **ImageSourceToVisibilityConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:ImageSourceToVisibilityConverter
            x:Key="imageSourceToVisibilityConverter"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...
```

```

<!-- Source ιδιότητα Photo. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <!-- Το TextBlock θα εμφανίζεται μόνο όταν υπάρχει source στην ιδιότητα
    Photo του student object -->
    <TextBlock Text="Φωτό:" Style="{StaticResource RowLabelsStyle }"
    HorizontalAlignment="Center"
    Visibility="{Binding Photo, Mode=OneWay,
    Converter={StaticResource imageSourceToVisibilityConverter} }"/>

    <Image Source="{Binding Photo, Mode=OneWay}" Margin="5,15,5,5"/>

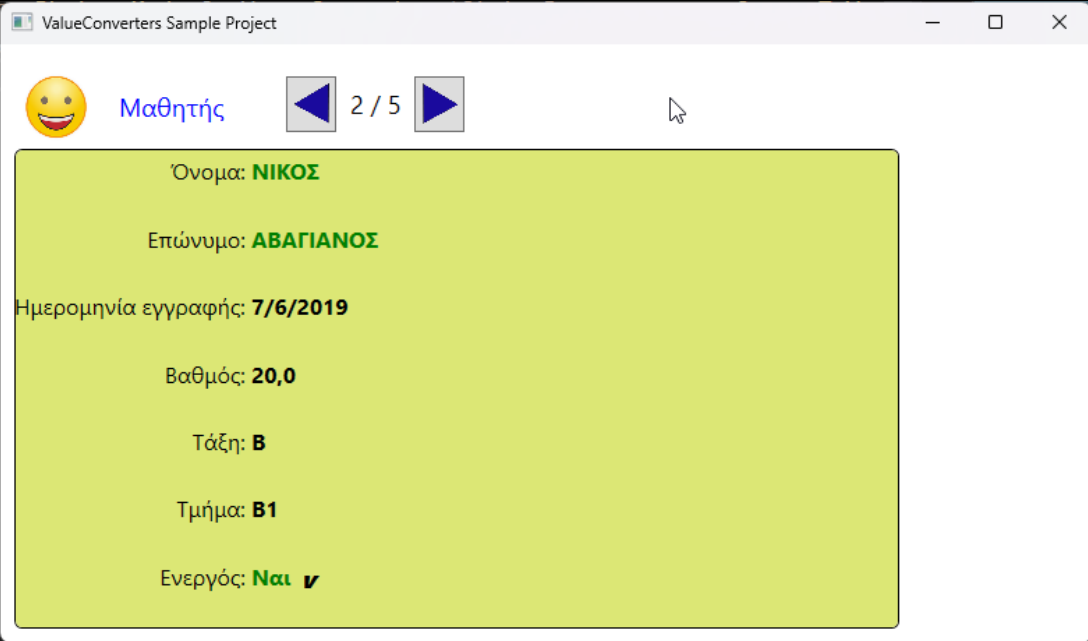
</StackPanel>

</Grid>

</Window>

```

Όταν η ιδιότητα Photo έχει δεδομένα το TextBlock "Φωτό:" εμφανίζεται.



Όταν η ιδιότητα Photo δεν έχει δεδομένα το TextBlock "Φωτό:" δεν εμφανίζεται.



ImageSourceToStringConverter

Μετατρέπει μία **ImageSource** τιμή σε **String**. Υποστηρίζει **one-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

<ValueConversion(GetType(ImageSource), GetType(String))>
Public Class ImageSourceToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:ImageSourceToStringConverter x:Key="imageSourceToStringConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyImageSourceType,
    Converter={StaticResource imageSourceToStringConverter}}" />
```

Παρατηρήσεις:

Η κλάση **ImageSourceToStringConverter**, εξ' ορισμού μετατρέπει την τιμή **Nothing** στο string **"No Image"**. και όταν έχει όντως source στο string **"Has image"**. Για να προσδιορίσουμε ποιό string θα επιστραφεί για την τιμή **Nothing** θα θέσουμε κατάλληλα την **ιδιότητα TextWhenHasNothing**. Παρόμοια, για να προσδιορίσουμε ποιό string θα επιστραφεί όταν έχει source θα θέσουμε κατάλληλα την **ιδιότητα TextWhenHasSource**.


```

<container.Resources >

    <zeus:ImageSourceToStringConverter x:Key="imageSourceToStringConverter"
        TextWhenHasSource=""
        TextWhenHasNothing="Δεν έχει φωτό" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **ImageSourceToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα Photo, τύπου ImageSource. Αυτή η ιδιότητα περιέχει τα δεδομένα της εικόνας του μαθητή. Θέλουμε όταν η **ιδιότητα Photo** έχει δεδομένα, τότε η **ιδιότητα Text** ενός ενημερωτικού **TextBlock** να γίνεται "" (κενό string), **αλλιώς "Δεν έχει φωτό"**. Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **ImageSourceToStringConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:ImageSourceToStringConverter x:Key="imageSourceToStringConverter"
            TextWhenHasSource=""
            TextWhenHasNothing="Δεν έχει φωτό"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Source ιδιότητα Photo. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

            <!-- Το TextBlock θα εμφανίζει ένα ενημερωτικό κείμενο μόνο όταν δεν
                υπάρχει source στην ιδιότητα Photo του student object -->
            <Image Source="{Binding Photo, Mode=OneWay}" Margin="5,15,5,5"/>

            <TextBlock Style="{StaticResource RowLabelsStyle }"
                HorizontalAlignment="Center"

```

```
Text="{Binding Photo, Mode=OneWay,  
Converter={StaticResource imageSourceToStringConverter}}" />  
  
    </StackPanel>  
  
    </Grid>  
  
</Window>
```

Ιδιότητες

Όνομα	Περιγραφή
TextWhenHasSource	Καθορίζει το String όταν η source ImageSource τιμή έχει δεδομένα .
TextWhenHasNothing	Καθορίζει το String όταν η source ImageSource τιμή είναι Nothing .

TextWhenHasSource

Καθορίζει το **String** όταν η **source ImageSource** τιμή έχει δεδομένα.

Σύνταξη:

VB:

```
Public Property TextWhenHasSource As String = "Has image"
```

Τύπος: System.String

Προσδιορίζουμε το **String** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source ImageSource** τιμή έχει δεδομένα. Η **default** τιμή είναι "Has image".

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **ImageSourceToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικείμενου **Student** είναι η ιδιότητα **Photo**, τύπου **ImageSource**. Αυτή η ιδιότητα περιέχει τα δεδομένα της εικόνας του μαθητή. Θέλουμε όταν η **ιδιότητα Photo** έχει δεδομένα, τότε η **ιδιότητα Text** ενός ενημερωτικού **TextBlock** να γίνεται "" (κενό string), **αλλιώς "Δεν έχει φωτό"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **ImageSourceToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:ImageSourceToStringConverter x:Key="imageSourceToStringConverter"
      TextWhenHasSource=""
      TextWhenHasNothing="Δεν έχει φωτό"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />
```

```
...  
  
<!-- Source ιδιότητα Photo. -->  
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >  
  
    <!-- Το TextBlock θα εμφανίζει ένα ενημερωτικό κείμενο μόνο όταν δεν  
        υπάρχει source στην ιδιότητα Photo του student object -->  
    <Image Source="{Binding Photo, Mode=OneWay}" Margin="5,15,5,5"/>  
  
    <TextBlock Style="{StaticResource RowLabelsStyle }"  
        HorizontalAlignment="Center"  
        Text="{Binding Photo, Mode=OneWay,  
            Converter={StaticResource imageSourceToStringConverter}}"/>  
  
    </StackPanel>  
  
</Grid>  
  
</Window>
```

TextWhenHasNothing

Καθορίζει το **String** όταν η **source ImageSource** τιμή είναι **Nothing**.

Σύνταξη:

VB:

```
Public Property TextWhenHasNothing As String = "No image"
```

Τύπος: System.String

Προσδιορίζουμε το **String** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source ImageSource** τιμή είναι **Nothing**. Η **default** τιμή είναι **"No image"**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **ImageSourceToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικειμένου **Student** είναι η ιδιότητα **Photo**, τύπου **ImageSource**. Αυτή η ιδιότητα περιέχει τα δεδομένα της εικόνας του μαθητή. Θέλουμε όταν η **ιδιότητα Photo** έχει δεδομένα, τότε η **ιδιότητα Text** ενός ενημερωτικού **TextBlock** να γίνεται "" (κενό string), **αλλιώς "Δεν έχει φωτό"**. Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **ImageSourceToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:ImageSourceToStringConverter x:Key="imageSourceToStringConverter"
      TextWhenHasSource=""
      TextWhenHasNothing="Δεν έχει φωτό"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />
```

```

...

<!-- Source ιδιότητα Photo. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <!-- Το TextBlock θα εμφανίζει ένα ενημερωτικό κείμενο μόνο όταν δεν
    υπάρχει source στην ιδιότητα Photo του student object -->
    <Image Source="{Binding Photo, Mode=OneWay}" Margin="5,15,5,5"/>

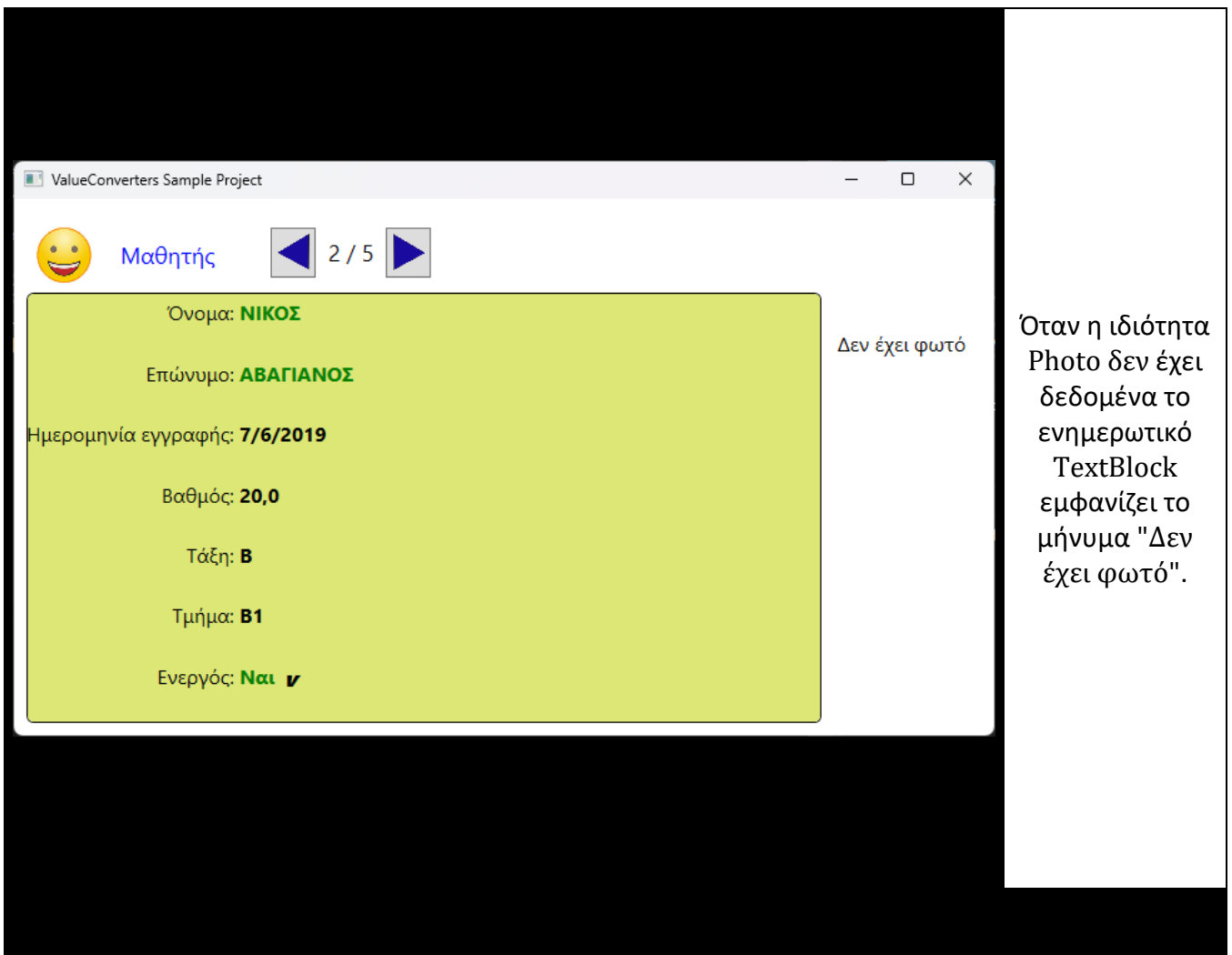
    <TextBlock Style="{StaticResource RowLabelsStyle }"
        HorizontalAlignment="Center"
        Text="{Binding Photo, Mode=OneWay,
            Converter={StaticResource imageSourceToStringConverter}}" />

</StackPanel>

</Grid>

</Window>

```





NumberNullableToStringConverter

Μετατρέπει μία **Nullable αριθμητική τιμή** (π.χ. **Byte?** ή **Short?** ή **Integer?** ή **Single?** κλπ) σε **String**. Υποστηρίζει **two-way binding mode**.

Σύνταξη:

VB:

```
Imports System.Windows.Data

Public Class NumberNullableToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:NumberNullableToStringConverter x:Key="numberNullableToStringConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyNullableNumberType,
    Converter={StaticResource numberNullableToStringConverter}}" />
```

Παρατηρήσεις:

Η κλάση **NumberNullableToStringConverter**, εξ' ορισμού μετατρέπει την τιμή **Nothing** σε **String.Empty**. Για να προσδιορίσουμε ποιά **String** θα επιστραφεί για την τιμή **Nothing** θα θέσουμε κατάλληλα την **ιδιότητα TextWhenHasNothing**.

```
<container.Resources >

    <zeus:NumberNullableToStringConverter x:Key="numberNullableToStringConverter"
        TextWhenHasNothing="Απαιτείται τιμή" />
```



```
</container.Resources>
```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **NumberNullableToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικείμενου Student είναι η ιδιότητα Absences, τύπου Nullable(of Integer). Αυτή η ιδιότητα καθορίζει τις απουσίες του μαθητή. Θέλουμε όταν η **ιδιότητα Absences** είναι **Nothing**, τότε η **ιδιότητα Text** του **TextBlock** να έχει κενό (φυσικά, αν δεν είναι *Nothing* τότε επιστρέφει το number string, σύμφωνα με την *StringFormat*). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **NumberNullableToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project"
  Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:NumberNullableToStringConverter
      x:Key="numberNullableToStringConverter" />

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />

    ...

    <!-- Source ιδιότητα Absences. -->
    <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

      <TextBlock Text="Απουσίες: " />
      <TextBlock Text="{Binding Absences, StringFormat=N0,
        ConverterCulture=el,
        Converter={StaticResource numberNullableToStringConverter}}"
        Margin="15,0,0,0" />

    </StackPanel>

  </Grid>

</Window>
```

ValueConverters Sample Project

😊 Μαθητής 1 / 5

Όνομα: **ΜΑΡΙΟΣ**

Επώνυμο: **ΛΕΩΝΙΔΗΣ**

Ημερομηνία εγγραφής: **4/6/2019**

Βαθμός: **15,5**

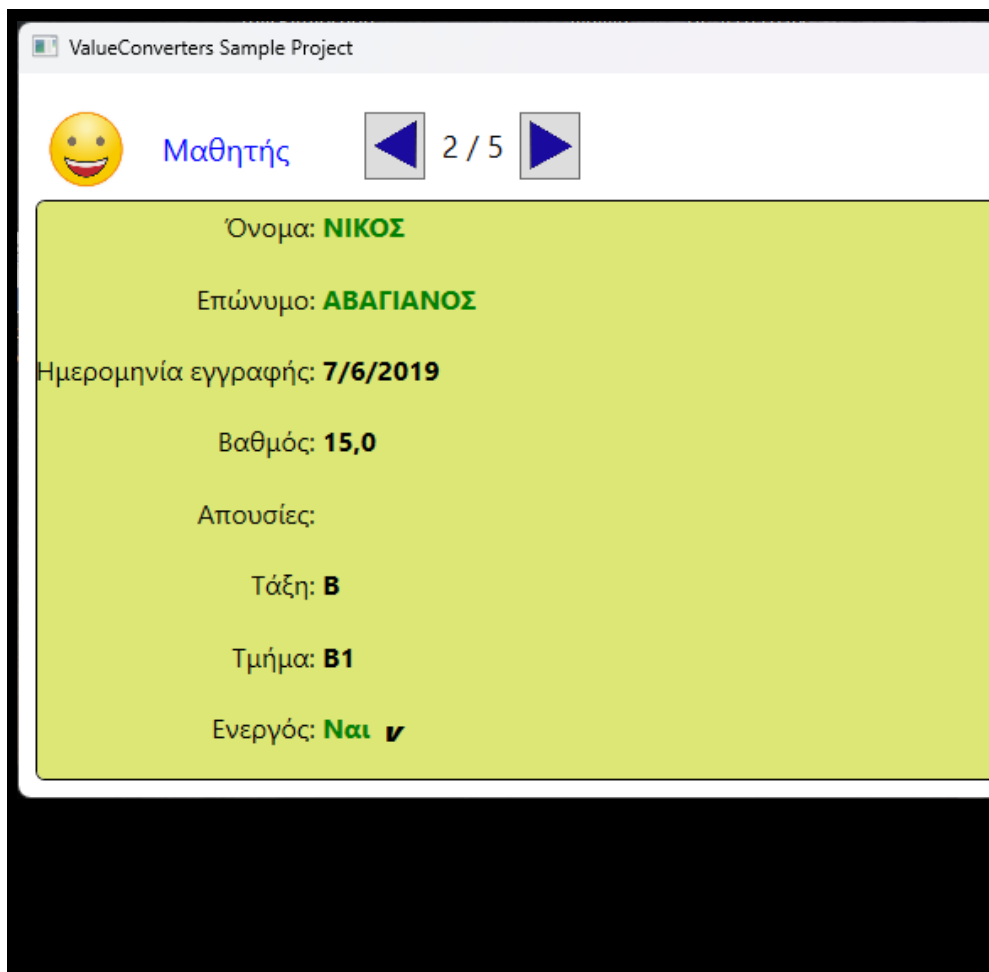
Απουσίες: **10**

Τάξη: **A**

Τμήμα: **A1**

Ενεργός: **Ναι** ✓

Όταν η ιδιότητα Absences δεν έχει τιμή Nothing τότε το αντίστοιχο TextBlock εμφανίζει την τιμή (εδώ 10).



The screenshot shows a web application window titled "ValueConverters Sample Project". At the top, there is a smiley face icon, the word "Μαθητής" (Student), and navigation arrows with "2 / 5" in between. Below this is a light green box containing the following student information:

- Όνομα: **ΝΙΚΟΣ**
- Επώνυμο: **ΑΒΑΓΙΑΝΟΣ**
- Ημερομηνία εγγραφής: **7/6/2019**
- Βαθμός: **15,0**
- Απουσίες:
- Τάξη: **B**
- Τμήμα: **B1**
- Ενεργός: **Ναι** ✓

On the right side of the screenshot, there is a text block explaining a UI behavior: "Όταν η ιδιότητα Absences έχει τιμή Nothing τότε το αντίστοιχο TextBlock δεν εμφανίζει τίποτα."

ValueConverters Sample Project

Μαθητής

◀

2 / 5

▶

Όνομα: **ΝΙΚΟΣ**

Επώνυμο: **ΑΒΑΓΙΑΝΟΣ**

Ημερομηνία εγγραφής: **7/6/2019**

Βαθμός: **15,0**

Απουσίες: **Δεν έχει απουσίες**

Τάξη: **B**

Τμήμα: **B1**

Ενεργός: **Ναι**

Εδώ έχουμε καθορίσει την ιδιότητα **TextWhenHasNothing** σε "**Δεν έχει απουσίες**".

Όταν η ιδιότητα **Absences** έχει τιμή **Nothing** τότε το αντίστοιχο **TextBlock** εμφανίζει το κείμενο αυτό.

Ιδιότητες

Όνομα	Περιγραφή
TextWhenHasNothing	Καθορίζει το String όταν η source τιμή είναι Nothing .

TextWhenHasNothing

Καθορίζει το **String** όταν η **source** τιμή είναι **Nothing**.

Σύνταξη:

VB:

```
Public Property TextWhenHasNothing As String = String.Empty
```

Τύπος: **System.String**

Προσδιορίζουμε το **String** που θα αποτελεί την μετατρεπόμενη τιμή όταν η **source** τιμή είναι **Nothing**. Η **default** τιμή είναι **String.Empty**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως **resource**, ένα **NumberNullableToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του **source** αντικείμενου **Student** είναι η ιδιότητα **Absences**, τύπου **Nullable(of Integer)**. Αυτή η ιδιότητα καθορίζει τις απουσίες του μαθητή. Θέλουμε όταν η **ιδιότητα Absences** είναι **Nothing**, τότε η **ιδιότητα Text** του **TextBlock** να εμφανίζει το κείμενο **"Δεν έχει απουσίες"** (φυσικά, αν δεν είναι **Nothing** τότε επιστρέφει το **number string**, σύμφωνα με την **StringFormat**). Στην σύνδεση με την **binding** έκφραση, την δουλειά αυτή κάνει το **NumberNullableToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample
Project" Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:NumberNullableToStringConverter
            x:Key="numberNullableToStringConverter"
            TextWhenHasNothing="Δεν έχει απουσίες"/>

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />
```

```

...

<!-- Source ιδιότητα Absences. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Απουσίες: " />
    <TextBlock Text="{Binding Absences, StringFormat=N0,
        ConverterCulture=el,
        Converter={StaticResource numberNullableToStringConverter}}"
        Margin="15,0,0,0" />

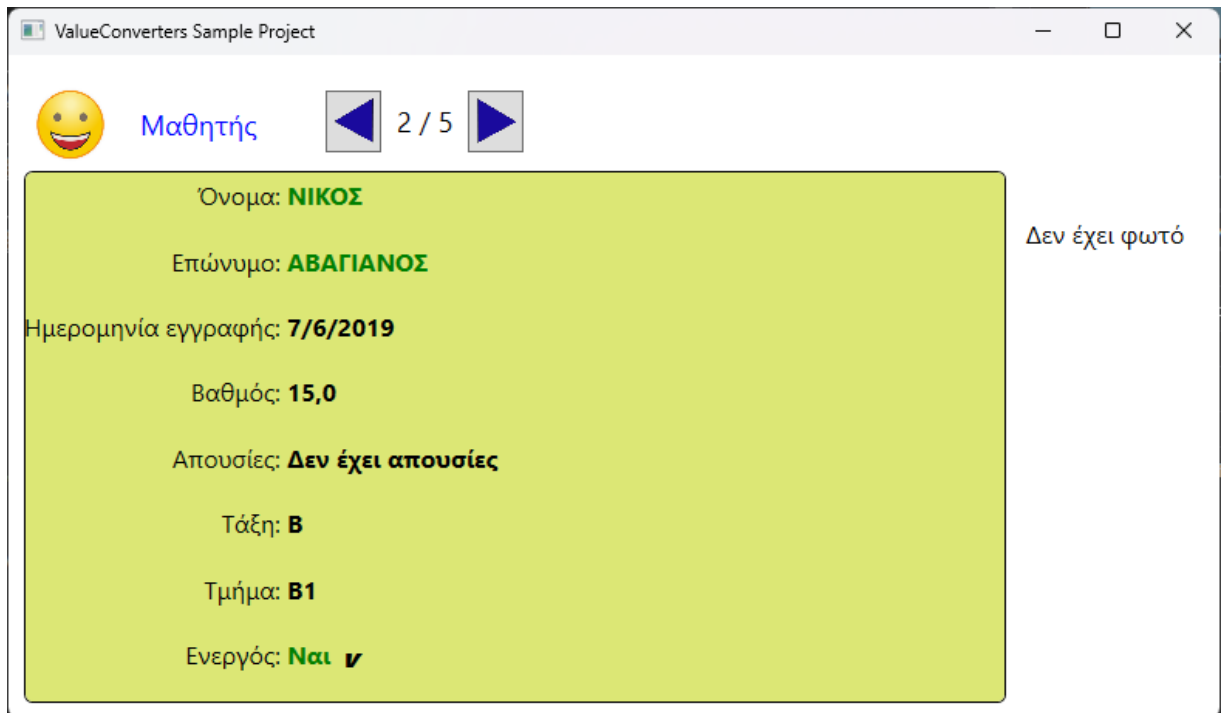
</StackPanel>

</Grid>

</Window>

```

Στην παρακάτω εικόνα, ο μαθητής Αβαγιανός εμφανίζει, στο πεδίο Απουσίες, το κείμενο "Δεν έχει απουσίες". Το αντικείμενο Student έχει στην ιδιότητα Absences την τιμή Nothing.





NumberAsNullableToStringConverter

Μετατρέπει μία **αριθμητική τιμή** (π.χ. **Byte** ή **Short** ή **Integer** ή **Single** κλπ) σε **String**. Καθορίζουμε μία **τιμή-φρουρό που λειτουργεί ως τιμή Nothing**. Όταν η source τιμή είναι ο φρουρός τότε επιστρέφει κενό ή συγκεκριμένο κείμενο. Υποστηρίζει **two-way** binding mode.

Σύνταξη:

VB:

```
Imports System.Windows.Data

Public Class NumberAsNullableToStringConverter
    Implements IValueConverter
```

XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueConverters"
```

Χρήση:

```
<container.Resources >

    <zeus:NumberAsNullableToStringConverter
        x:Key="numberAsNullableToStringConverter"/>

</container.Resources>

<Element property="{Binding sourcePropertyNumberType,
    Converter={StaticResource numberAsNullableToStringConverter}}" />
```

Παρατηρήσεις:

Η κλάση **NumberAsNullableToStringConverter**, εξ' ορισμού μετατρέπει την τιμή -1 (φρουρός) σε **String.Empty**. Για να προσδιορίσουμε ποιό θα είναι η συγκεκριμένη **τιμή-φρουρός** θα θέσουμε κατάλληλα την **ιδιότητα NumberAsNothing** ενώ για να προσδιορίσουμε ποιό **String** θα επιστραφεί για την τιμή **Nothing** θα θέσουμε κατάλληλα την **ιδιότητα TextWhenNumberNull**.


```

<container.Resources >

    <zeus:NumberAsNullableToStringConverter
        x:Key="numberAsNullableToStringConverter"
        NumberAsNothing="0"
        TextWhenNumberNull="Δεν υπάρχει τιμή" />

</container.Resources>

```

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **NumberAsNullableToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα Grade, τύπου Single. Αυτή η ιδιότητα καθορίζει τον βαθμό του μαθητή. Θέλουμε όταν η **ιδιότητα Grade** είναι **-1**, τότε η **ιδιότητα Text** του **TextBlock** να έχει κενό (φυσικά, αν δεν είναι **-1** τότε επιστρέφει το *number string*, σύμφωνα με την *StringFormat*). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **NumberAsNullableToStringConverter** αντικείμενο.

```

<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
    mc:Ignorable="d"
    Title="ValueConverters Sample Project"
    Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

    <Window.Resources >

        <!-- ValueConverter -->
        <zeus:NumberAsNullableToStringConverter
            x:Key="numberAsNullableToStringConverter" />

    </Window.Resources>

    <Grid Margin="10" Name="grdStudent">

        <!-- Title-->
        <TextBlock Text="Μαθητής" />

        ...

        <!-- Source ιδιότητα Grade. -->
        <StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

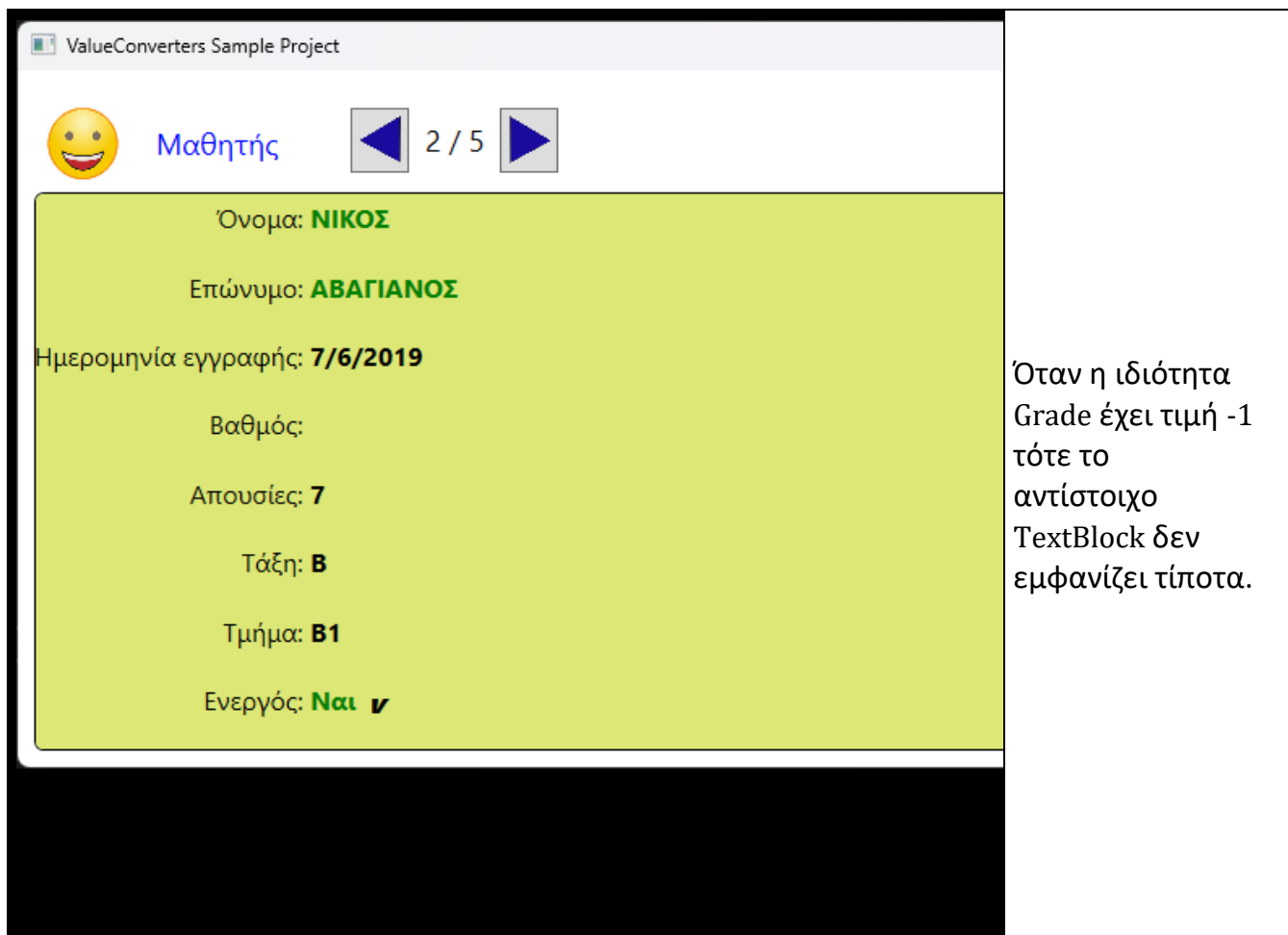
            <TextBlock Text="Βαθμός: " />
            <TextBlock Text="{Binding Grade, StringFormat=N1,
                ConverterCulture=el,
                Converter={StaticResource numberAsNullableToStringConverter}}"
                Margin="15,0,0,0" />

```

```

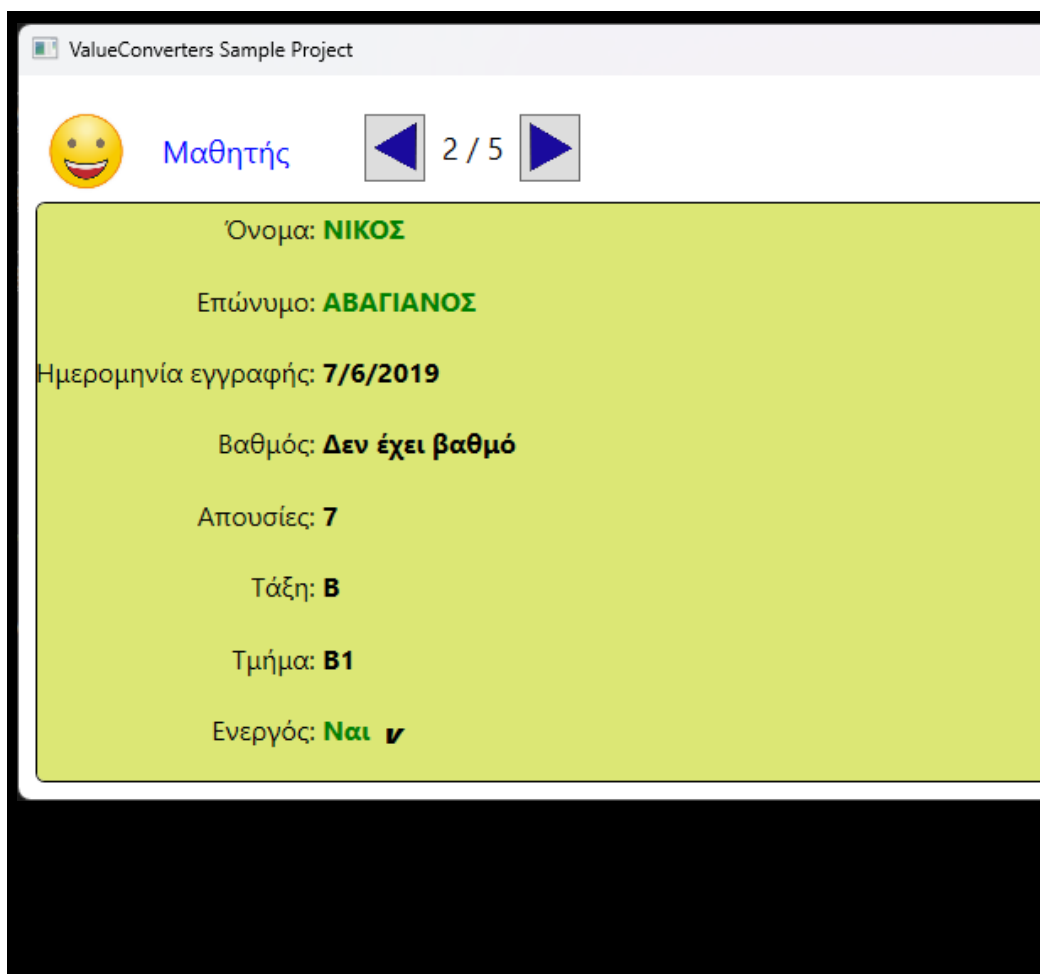
        </StackPanel>
    </Grid>
</Window>
    
```

Όταν η ιδιότητα Grade δεν έχει τιμή -1 τότε το αντίστοιχο TextBlock εμφανίζει την τιμή (εδώ 15,5).



The screenshot shows a mobile application window titled "ValueConverters Sample Project". At the top, there is a yellow smiley face icon, the word "Μαθητής" (Student) in blue, and a navigation bar with left and right arrows and "2 / 5". Below this is a light green card with the following text: "Όνομα: **ΝΙΚΟΣ**", "Επώνυμο: **ΑΒΑΓΙΑΝΟΣ**", "Ημερομηνία εγγραφής: **7/6/2019**", "Βαθμός:", "Απουσίες: **7**", "Τάξη: **B**", "Τμήμα: **B1**", and "Ενεργός: **Ναι** ✓".

Όταν η ιδιότητα Grade έχει τιμή -1 τότε το αντίστοιχο TextBlock δεν εμφανίζει τίποτα.



ValueConverters Sample Project

😊 Μαθητής 2 / 5

Όνομα: **ΝΙΚΟΣ**

Επώνυμο: **ΑΒΑΓΙΑΝΟΣ**

Ημερομηνία εγγραφής: **7/6/2019**

Βαθμός: **Δεν έχει βαθμό**

Απουσίες: **7**

Τάξη: **B**

Τμήμα: **B1**

Ενεργός: **Ναι** ✓

Εδώ έχουμε καθορίσει την ιδιότητα **TextWhenNumber Null** σε "Δεν έχει βαθμό".

Όταν η ιδιότητα **Grade** έχει τιμή -1 τότε το αντίστοιχο **TextBlock** εμφανίζει το κείμενο αυτό.

Ιδιότητες

Όνομα	Περιγραφή
NumberAsNothing	Καθορίζει τον αριθμό-φρουρό που θα λειτουργεί ως Nothing . Είναι τύπου Integer . Η default τιμή είναι -1.
TextWhenNumberNull	Καθορίζει το String όταν η source τιμή έχει την τιμή-φρουρό, όπως αυτή προσδιορίζεται από την ιδιότητα NumberAsNothing . Η default τιμή είναι String.Empty .

NumberAsNothing

Καθορίζει τον αριθμό-φρουρό που θα λειτουργεί ως **Nothing**. Είναι τύπου **Integer**.

Σύνταξη:

VB:

```
Public Property NumberAsNothing As Integer = -1
```

Τύπος: **System.Integer**

Προσδιορίζουμε τον αριθμό που θα λειτουργεί ως **Nothing**. Η default τιμή είναι -1.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **NumberAsNullableToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου Student είναι η ιδιότητα Grade, τύπου Single. Αυτή η ιδιότητα καθορίζει τον βαθμό του μαθητή. Θέλουμε όταν η **ιδιότητα Grade** είναι **0**, τότε η **ιδιότητα Text** του **TextBlock** να έχει κενό (φυσικά, αν δεν είναι 0 τότε επιστρέφει το number string, σύμφωνα με την *StringFormat*). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **NumberAsNullableToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project"
  Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:NumberAsNullableToStringConverter
      x:Key="numberAsNullableToStringConverter"
      NumberAsNothing="0"/>

  </Window.Resources>

  <Grid Margin="10" Name="grdStudent">

    <!-- Title-->
    <TextBlock Text="Μαθητής" />

    ...
```

```

<!-- Source ιδιότητα Grade. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Βαθμός: " />
    <TextBlock Text="{Binding Grade, StringFormat=N1,
        ConverterCulture=en,
        Converter={StaticResource numberAsNullableToStringConverter}}"
        Margin="15,0,0,0" />

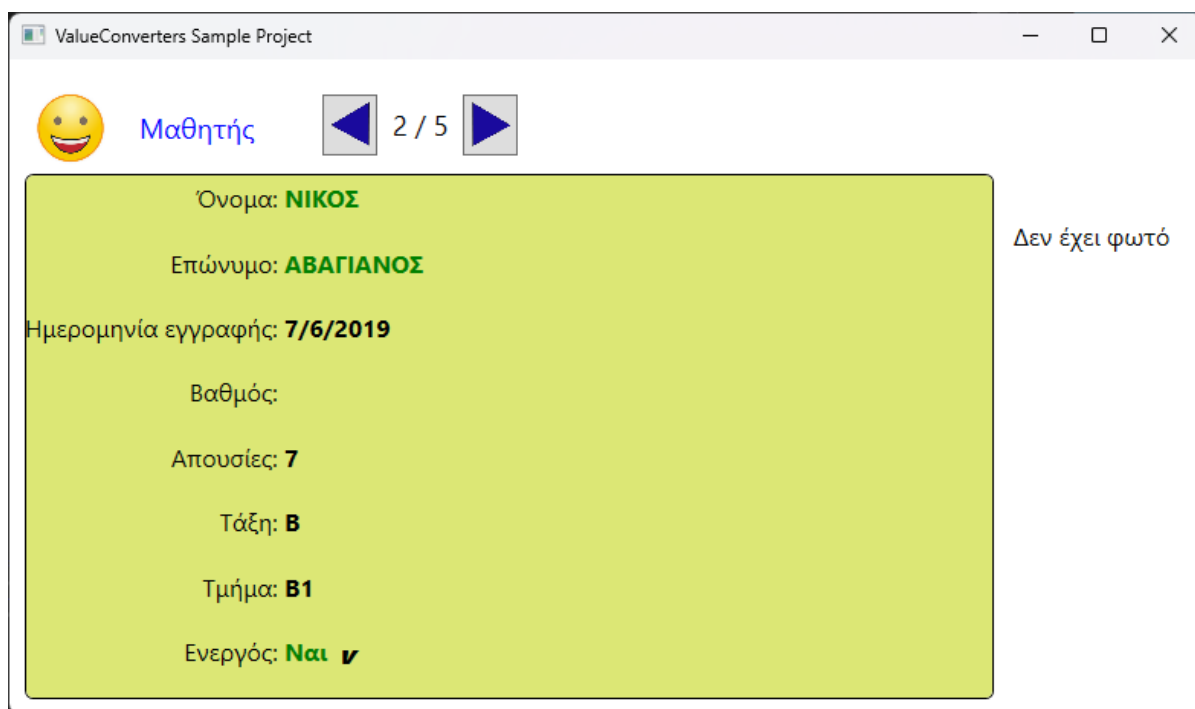
</StackPanel>

</Grid>

</Window>

```

Στην παρακάτω εικόνα, ο μαθητής Αβαγιανός δεν εμφανίζει κάποιον βαθμό (κενό). Το αντικείμενο Student έχει στην ιδιότητα Grade την τιμή 0.



TextWhenNumberNull

Καθορίζει το **String** όταν η **source** τιμή έχει την τιμή που προσδιορίζεται από την ιδιότητα **NumberAsNothing** (τιμή-φρουρός)

Σύνταξη:

VB:

```
Public Property TextWhenNumberNull As String = String.Empty
```

Τύπος: **System.String**

Προσδιορίζουμε το string που θα εμφανίζεται όταν η source τιμή είναι ίση με την τιμή-φρουρό, όπως προσδιορίζεται στην ιδιότητα **NumberAsNothing**. Η default τιμή είναι **String.Empty**.

Παράδειγμα:

Στο παράδειγμα που ακολουθεί, έχουμε εισάγει, ως resource, ένα **NumberAsNullableToStringConverter** αντικείμενο. Παρακάτω, υπάρχει μία φόρμα μαθητή, όπου μεταξύ άλλων ιδιοτήτων του source αντικειμένου **Student** είναι η ιδιότητα **Grade**, τύπου **Single**. Αυτή η ιδιότητα καθορίζει τον βαθμό του μαθητή. Θέλουμε όταν η **ιδιότητα Grade** είναι **0**, τότε η **ιδιότητα Text** του **TextBlock** να έχει το κείμενο "**Δεν έχει βαθμό**" (φυσικά, αν δεν είναι 0 τότε επιστρέφει το number string, σύμφωνα με την *StringFormat*). Στην σύνδεση με την binding έκφραση, την δουλειά αυτή κάνει το **NumberAsNullableToStringConverter** αντικείμενο.

```
<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:zeus="clr-namespace:Zeus.WPF.Classes.ValueConverters;assembly=ZeusValueCo
nverters"
  mc:Ignorable="d"
  Title="ValueConverters Sample Project"
  Height="467.775" Width="799.064" WindowStartupLocation="CenterScreen" >

  <Window.Resources >

    <!-- ValueConverter -->
    <zeus:NumberAsNullableToStringConverter
      x:Key="numberAsNullableToStringConverter"
      NumberAsNothing="0"
      TextWhenNumberNull="Δεν έχει βαθμό"/>

  </Window.Resources >

  <Grid Margin="10" Name="grdStudent">
```



```

<!-- Title-->
<TextBlock Text="Μαθητής" />

...

<!-- Source ιδιότητα Grade. -->
<StackPanel Orientation="Horizontal" Grid.Row="7" Grid.Column="1" >

    <TextBlock Text="Βαθμός: " />
    <TextBlock Text="{Binding Grade, StringFormat=N1,
        ConverterCulture=el,
        Converter={StaticResource numberAsNullableToStringConverter}}"
        Margin="15,0,0,0" />

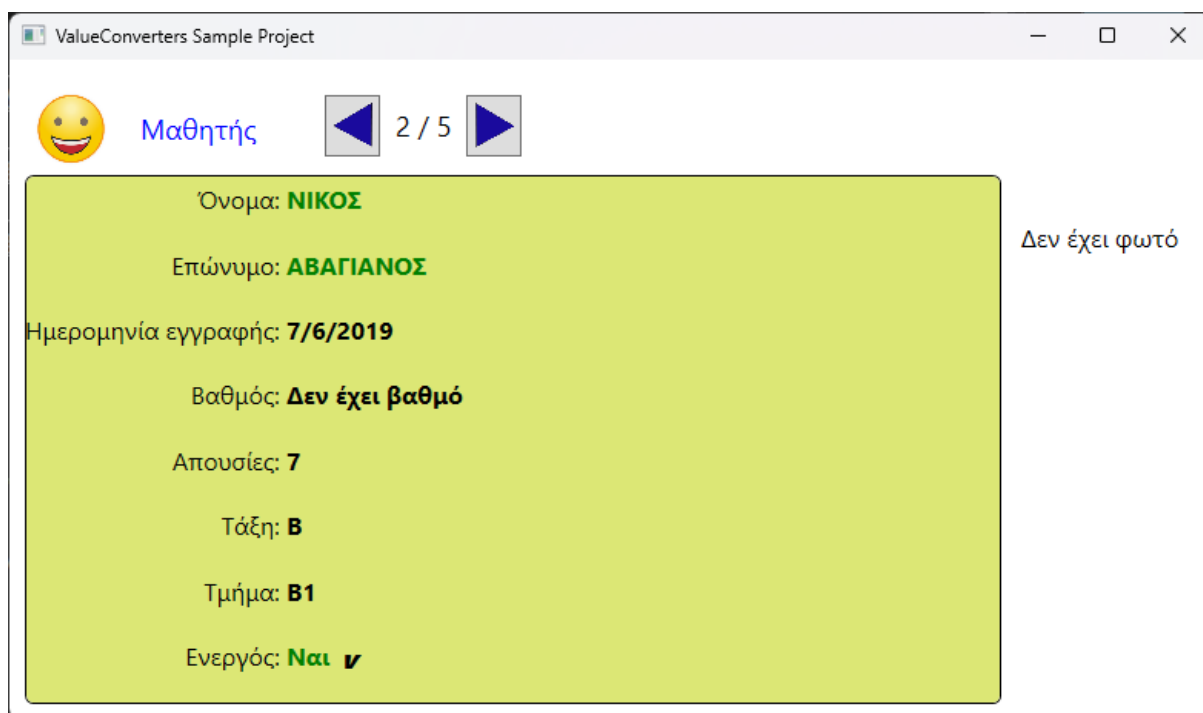
</StackPanel>

</Grid>

</Window>

```

Στην παρακάτω εικόνα, ο μαθητής Αβαγιανός εμφανίζει, στο πεδίο Βαθμός, το κείμενο "Δεν έχει βαθμό". Το αντικείμενο Student έχει στην ιδιότητα Grade την τιμή 0.



Τέλος Manual