

# **Zeus DataControls Ver. 1.1**

## **Manual**

## Πίνακας περιεχομένων

ZEUS DATACONTROLS VERSION 1.1 .....	6
ΟΙ ΟΡΙΣΜΟΙ ΤΩΝ ΚΛΑΣΕΩΝ .....	8
ΔΙΑΝΟΜΗ .....	12
ΕΠΙΚΟΙΝΩΝΙΑ.....	13
<b>SORTCONTROL.....</b>	<b>14</b>
ΑΠΑΡΙΘΜΗΣΕΙΣ.....	20
<i>OrderEnum</i> .....	21
ΙΔΙΟΤΗΤΕΣ.....	22
<i>AscendingText</i> .....	25
<i>DescendingText</i> .....	59
<i>FieldsColumnHeaderText</i> .....	61
<i>FieldsNames</i> .....	63
<i>FieldsNamesOriginal</i> .....	65
<i>FirstRowHeaderText</i> .....	67
<i>GridSortRowsContainerBackground</i> .....	69
<i>OrderColumnHeaderText</i> .....	71
<i>RowHeaderText</i> .....	73
<i>RowHeaderTextStyle</i> .....	75
<i>SelectedRowHeaderTextStyle</i> .....	78
<i>SortListInput</i> .....	81
<i>SortListOutput</i> .....	83
<i>SortStringOutput</i> .....	85
ΣΥΜΒΑΝΤΑ .....	87
<i>OKClicked</i> .....	88
<i>CancelClicked</i> .....	89
ΜΕΘΟΔΟΙ.....	90
<i>CreateSorting</i> .....	91
<i>LoadFromFile</i> .....	92
<i>SaveToFile</i> .....	94
STYLES ΚΑΙ TEMPLATES.....	96
<i>Parts και States</i> .....	97
<i>To default Style και ControlTemplate</i> .....	98
ΚΛΑΣΕΙΣ .....	102
<i>SortRowInfo</i> .....	103
<b>CRITERIACONTROL.....</b>	<b>104</b>
ΑΠΑΡΙΘΜΗΣΕΙΣ.....	111
<i>BooleanValuesEnum</i> .....	112
<i>ComparisonBooleanOperatorEnum</i> .....	113
<i>ComparisonNumAndDateOperatorsEnum</i> .....	114
<i>ComparisonStringOperatorsEnum</i> .....	115
<i>ConnectorEnum</i> .....	116
<i>DatabaseFormatEnum</i> .....	117
<i>FieldGeneralDataTypeEnum</i> .....	118
ΙΔΙΟΤΗΤΕΣ.....	119
<i>BooleanValueTrueText</i> .....	125

---

<i>BooleanValueFalseText</i> .....	127
<i>ButtonAddLevelText</i> .....	129
<i>ButtonAddLevelStyle</i> .....	131
<i>ButtonCancelText</i> .....	134
<i>ButtonCancelStyle</i> .....	136
<i>ButtonOKText</i> .....	139
<i>ButtonOKStyle</i> .....	141
<i>ButtonRemoveLevelText</i> .....	144
<i>ButtonRemoveLevelStyle</i> .....	146
<i>ButtonsLevelContainerBackground</i> .....	149
<i>ButtonsOKCancelContainerBackground</i> .....	151
<i>ButtonsOKCancelWidth</i> .....	153
<i>ComboBoxBooleanValueStyle</i> .....	155
<i>ComboBoxComparisonStyle</i> .....	158
<i>ComboBoxConnectorStyle</i> .....	161
<i>ComboBoxConnectorWidth</i> .....	164
<i>ComboBoxFieldsNamesStyle</i> .....	166
<i>ComparisonColumnHeaderText</i> .....	169
<i>ComparisonOperatorBoolEqualText</i> .....	171
<i>ComparisonOperatorBeginsWithText</i> .....	173
<i>ComparisonOperatorContainsText</i> .....	175
<i>ComparisonOperatorEndsWithText</i> .....	177
<i>ComparisonOperatorEqualText</i> .....	179
<i>ComparisonOperatorNotEqualText</i> .....	181
<i>ComparisonOperatorNumEqualText</i> .....	183
<i>ComparisonOperatorNumGreaterThanOrEqualText</i> .....	185
<i>ComparisonOperatorNumGreaterThanText</i> .....	187
<i>ComparisonOperatorNumLessThanOrEqualText</i> .....	189
<i>ComparisonOperatorNumLessThanText</i> .....	191
<i>ComparisonOperatorNumNotEqualText</i> .....	193
<i>ConnectorANDText</i> .....	195
<i>ConnectorORText</i> .....	197
<i>CriteriaListInput</i> .....	199
<i>CriteriaListOutput</i> .....	202
<i>CriteriaStringOutput</i> .....	204
<i>DatabaseFormat</i> .....	207
<i>DatePickerCalendarStyle</i> .....	210
<i>DatePickerStyle</i> .....	213
<i>FieldsColumnHeaderText</i> .....	216
<i>FieldsDataTypes</i> .....	218
<i>FieldsNames</i> .....	220
<i>FieldsNamesOriginal</i> .....	222
<i>GridCriteriaRowsContainerBackground</i> .....	224
<i>RowHeaderStyle</i> .....	226
<i>RowHeaderWidth</i> .....	229
<i>SelectedRowHeaderStyle</i> .....	230
<i>UseCommaAsDecimalSeparator</i> .....	233
<i>ValueColumnHeaderText</i> .....	234
<i>ValueTextBoxErrorStyle</i> .....	236
<i>ValueTextBoxStyle</i> .....	239
<i>ΣΥΜΒΑΝΤΑ</i> .....	242
<i>OKClicked</i> .....	243

---

<i>CancelClicked</i> .....	244
ΜΕΘΟΔΟΙ.....	245
<i>CreateCriteria</i> .....	246
<i>LoadFromFile</i> .....	247
<i>SaveToFile</i> .....	249
<i>ValidateCriteria</i> .....	252
STYLES ΚΑΙ TEMPLATES.....	253
<i>Parts και States</i> .....	254
<i>To default Style και ControlTemplate</i> .....	255
ΚΛΑΣΕΙΣ .....	260
<i>CriteriaRowInfo</i> .....	261
DATANAVIGATOR .....	263
ΙΔΙΟΤΗΤΕΣ.....	274
<i>ButtonsWidth</i> .....	277
<i>ButtonsStyle</i> .....	278
<i>CView</i> .....	281
<i>IsAddingNew</i> .....	283
<i>IsAddingNewOrIsEditing</i> .....	285
<i>IsEditing</i> .....	287
<i>InfoText</i> .....	289
<i>InfoTextStyle</i> .....	291
<i>ShowAddNew</i> .....	294
<i>ShowCommit</i> .....	295
<i>ShowEdit</i> .....	296
<i>ShowRemove</i> .....	297
<i>ShowToolTipsForButtons</i> .....	298
<i>ShowToolTipForInfo</i> .....	299
<i>ToolTipTextForAddNew</i> .....	300
<i>ToolTipTextForCancelEdit</i> .....	301
<i>ToolTipTextForCancelNew</i> .....	302
<i>ToolTipTextForCommit</i> .....	303
<i>ToolTipTextForEdit</i> .....	304
<i>ToolTipTextForInfo</i> .....	305
<i>ToolTipTextForMoveToFirst</i> .....	306
<i>ToolTipTextForMoveToLast</i> .....	307
<i>ToolTipTextForMoveToNext</i> .....	308
<i>ToolTipTextForMoveToPrevious</i> .....	309
<i>ToolTipTextForRemove</i> .....	310
ΣΥΜΒΑΝΤΑ .....	311
<i>AddNewStarted</i> .....	312
<i>CommitCompleted</i> .....	314
<i>CommitStarted</i> .....	316
<i>EditItemStarted</i> .....	318
<i>MoveCompleted</i> .....	321
<i>MoveStarted</i> .....	323
<i>RemoveCompleted</i> .....	325
<i>RemoveStarted</i> .....	327
ΜΕΘΟΔΟΙ.....	331
<i>AddNew</i> .....	333
<i>AddNewItem</i> .....	334
<i>Commit</i> .....	335

<i>CommitStart</i> .....	337
<i>Count</i> .....	339
<i>CurrentAddItem</i> .....	340
<i>CurrentEditItem</i> .....	341
<i>CurrentItem</i> .....	343
<i>CurrentPosition</i> .....	344
<i>EditItem</i> .....	345
<i>HasItems</i> .....	346
<i>MoveToFirst</i> .....	347
<i>MoveToLast</i> .....	348
<i>MoveToNext</i> .....	349
<i>MoveToPosition</i> .....	350
<i>MoveToPrevious</i> .....	351
STYLES ΚΑΙ TEMPLATES .....	352
<i>Parts και States</i> .....	353
<i>To default Style και ControlTemplate</i> .....	355
<b>FINDXPRESS</b> .....	<b>363</b>
ΑΠΑΡΙΘΜΗΣΕΙΣ.....	370
<i>FindOrientationEnum</i> .....	371
<i>LanguageForUIEnum</i> .....	372
ΙΔΙΟΤΗΤΕΣ.....	373
<i>ButtonCancelStyle</i> .....	375
<i>ButtonCancelText</i> .....	377
<i>ButtonOKStyle</i> .....	379
<i>ButtonOKText</i> .....	381
<i>ButtonsOKCancelWidth</i> .....	383
<i>CheckBoxFindMatchCaseStyle</i> .....	385
<i>ComboBoxFindOrientationStyle</i> .....	387
<i>CView</i> .....	389
<i>FindMatchCase</i> .....	391
<i>FindMatchCaseCaption</i> .....	395
<i>FindOptionsCaption</i> .....	397
<i>FindOrientation</i> .....	399
<i>FindOrientationTextContains</i> .....	401
<i>FindOrientationTextEndsWith</i> .....	403
<i>FindOrientationTextEqualTo</i> .....	405
<i>FindOrientationTextStartsWith</i> .....	407
<i>LanguageForUI</i> .....	409
<i>ListBoxResultStyle</i> .....	412
<i>ReturnValue</i> .....	415
<i>TextBoxValueStyle</i> .....	416
<i>ToolTipTextForValue</i> .....	418
<i>ShowToolTipForValue</i> .....	419
ΣΥΜΒΑΝΤΑ .....	420
<i>OKClicked</i> .....	421
<i>CancelClicked</i> .....	422
STYLES ΚΑΙ TEMPLATES .....	423
<i>Parts και States</i> .....	424
<i>To default Style και ControlTemplate</i> .....	425
<i>Custom Style and ControlTemplate</i> .....	428



## Zeus DataControls version 1.1



**Κλάσεις:** SortControl CriteriaControl, DataNavigator, FindXPress (νέο στην έκδοση 1.1)

**Inherits:** System.Windows.Controls

**Namespace:** Zeus.WPF.Controls.DataControls

**Assembly:** ZeusDataControls (in ZeusDataControls.dll)



**Dependencies:** -



### Περιγραφή

Η βιβλιοθήκη **Zeus DataControls** περιλαμβάνει τρία **Controls**, για την **διαχείριση δεδομένων από έναν χρήστη**. Ένα **SortControl** για την διαμόρφωση ταξινόμησης, ένα **CriteriaControl** για την διαμόρφωση κριτηρίων επιλογής (φιλτραρίσματος) δεδομένων κι ένα **DataNavigator** για την πλοήγηση, προσθήκη, διόρθωση, αποθήκευση και διαγραφή δεδομένων. Το τελευταίο μοιάζει (αλλά και επεκτείνει) με το παλιό καλό Data control της VB6 (το θυμάται κανείς;).

Στην **έκδοση 1.1**, προστέθηκε το **FindXPress** για τη γρήγορη αναζήτηση κι επιλογή κάποιου δεδομένου (κειμένου) από μία λίστα η οποία περιέχει μοναδικές (unique) τιμές.

Παρακάτω, παρατίθενται τα **DataControls**.

	<a href="#">SortControl</a>	<p>Επιτρέπει στον χρήστη <b>να επιλέξει ένα ή περισσότερα επίπεδα ταξινόμησης</b>, με βάση κάποια πεδία που του εμφανίζονται. Σε κάθε επίπεδο, επιλέγει το πεδίο και τη διάταξη ταξινόμησης (αύξουσα/φθίνουσα). Επιστρέφει ένα string που μπορεί να χρησιμοποιεί στο τμήμα Order by μίας SQL select query.</p> <p>Είναι παρόμοιο με το παράθυρο ταξινόμησης δεδομένων του Excel.</p>
	<a href="#">CriteriaControl</a>	<p>Επιτρέπει στον χρήστη <b>να επιλέξει ένα ή περισσότερα επίπεδα κριτηρίων, που μπορεί να συνδέονται με και/ή</b>. Πάλι, σε κάθε επίπεδο επιλέγει το πεδίο, τον τελεστή (π.χ. ισούται με, μεγαλύτερο από κλπ) και την τιμή. Επιστρέφει ένα string που μπορεί να χρησιμοποιεί στο τμήμα Where μίας SQL select query.</p>

	<p><a href="#">DataNavigator</a></p>	<p>Παρέχει στον χρήστη <b>μία σειρά κουμπιών (σαν toolbar) που του επιτρέπει να πλοηγηθεί σε μία συλλογή εγγραφών/αντικειμένων</b>. Η πλοήγηση περιλαμβάνει πρώτη, προηγούμενη, επόμενη και τελευταία εγγραφή σε συνδυασμό με ένα πληροφοριακό label (πχ. τρέχουσα/σύνολο).</p> <p>Επιπλέον, <b>μπορούν να εμφανίζονται κουμπιά για υποστήριξη της προσθήκης, διόρθωσης, αποθήκευσης και διαγραφής μίας εγγραφής/αντικειμένου</b>.</p> <p>Είναι ιδανικό, για να κατασκευάσουμε γύρω από αυτό μία φόρμα εν είδει καρτέλας, όπου θέλουμε να εμφανίζονται οι εγγραφές/αντικείμενα ένα προς ένα.</p>
	<p><a href="#">FindXPress</a></p> <p>(νέο στην έκδοση 1.1)</p>	<p>Παρέχει στο χρήστη μία μικρή φόρμα που περιέχει ένα <b>κουτί αναζήτησης, επιλογές αναζήτησης (π.χ. να ξεκινάει με, να περιέχει κλπ καθώς και διάκριση κεφαλαίων-πεζών), μία λίστα με μοναδικές (unique) τιμές που προσαρμόζεται (φιλτράρεται) ανάλογα με το τί γράφει ο χρήστης και τις επιλογές αναζήτησης που έχει κάνει και δύο κουμπιά OK και Ακύρωση</b>.</p> <p>Όταν ο χρήστης επιλέξει από τη λίστα μία τιμή και πατήσει OK τότε το control επιστρέφει την επιλεγείσα τιμή.</p> <p>Μπορούμε να το συνδυάσουμε με ένα ComboBox που περιέχει μία μεγάλη λίστα μοναδικών τιμών (π.χ. επαγγέλματα) ώστε ο χρήστης να βρει γρήγορα αυτό που ζητάει.</p>

## Παρατηρήσεις

Να σημειώσουμε ότι το SortControl και CriteriaControl έχουν τη δυνατότητα να αποθηκεύσουν τις επιλογές ταξινόμησης και κριτηρίων, αντίστοιχα, του χρήστη. Με αυτόν τον τρόπο, μπορεί να επανέρχεται στις αποθηκευμένες επιλογές χωρίς να χρειάζεται να τις διαμορφώνει ξανά και ξανά.



## Οι ορισμοί των κλάσεων

Οι κλάσεις έχουν οριστεί ως εξής:

---

- Για την κλάση **SortControl** :

**Σύνταξη:**

**VB:**

```
<TemplateVisualState(Name="Normal", GroupName="CommonStates"),  
TemplateVisualState(Name="MouseOver", GroupName="CommonStates"),  
TemplateVisualState(Name="Disabled", GroupName="CommonStates"),  
TemplateVisualState(Name="Focused", GroupName="FocusStates"),  
TemplateVisualState(Name="Unfocused", GroupName="FocusStates"),  
TemplatePart(Name="PART_AddLevel", Type:=GetType(Button)),  
TemplatePart(Name="PART_RemoveLevel", Type:=GetType(Button)),  
TemplatePart(Name="PART_OK", Type:=GetType(Button)),  
TemplatePart(Name="PART_Cancel", Type:=GetType(Button)),  
TemplatePart(Name="PART_SortRowsContainer", Type:=GetType(Grid))>  
Public Class SortControl  
    Inherits Control
```

**XAML Object Element Usage:**

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:SortControl ... />
```

---

- Για την κλάση **CriteriaControl**:

**Σύνταξη:**

**VB:**

```
<TemplateVisualState(Name="Normal", GroupName="CommonStates"),  
TemplateVisualState(Name="MouseOver", GroupName="CommonStates"),  
TemplateVisualState(Name="Disabled", GroupName="CommonStates"),  
TemplateVisualState(Name="Focused", GroupName="FocusStates"),  
TemplateVisualState(Name="Unfocused", GroupName="FocusStates"),  
TemplatePart(Name="PART_AddLevel", Type:=GetType(Button)),  
TemplatePart(Name="PART_RemoveLevel", Type:=GetType(Button)),  
TemplatePart(Name="PART_OK", Type:=GetType(Button)),
```

```
TemplatePart(Name:="PART_Cancel", Type:=GetType(Button)),  
TemplatePart(Name:="PART_CriteriaRowsContainer", Type:=GetType(Grid))>
```

```
Public Class CriteriaControl  
    Inherits Control
```

### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:CriteriaControl ... />
```

---

- Για την κλάση **DataNavigator**:

**Σύνταξη:**

**VB:**

```
<TemplateVisualState(Name:="Normal", GroupName:="CommonStates"),  
TemplateVisualState(Name:="MouseOver", GroupName:="CommonStates"),  
TemplateVisualState(Name:="Disabled", GroupName:="CommonStates"),  
TemplateVisualState(Name:="Focused", GroupName:="FocusStates"),  
TemplateVisualState(Name:="Unfocused", GroupName:="FocusStates"),  
TemplatePart(Name:="PART_First", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Previous", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Next", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Last", Type:=GetType(Button)),  
TemplatePart(Name:="PART_InfoText", Type:=GetType(Label)),  
TemplatePart(Name:="PART_AddNew", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Edit", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Commit", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Remove", Type:=GetType(Button))>  
Public Class DataNavigator  
    Inherits Control
```

### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:DataNavigator ... />
```

---

- Για την κλάση **FindXPress::**

### Σύνταξη:

#### VB:

```
<TemplateVisualState(Name:="Normal", GroupName:="CommonStates"),  
TemplateVisualState(Name:="MouseOver", GroupName:="CommonStates"),  
TemplateVisualState(Name:="Disabled", GroupName:="CommonStates"),  
TemplateVisualState(Name:="Focused", GroupName:="FocusStates"),  
TemplateVisualState(Name:="Unfocused", GroupName:="FocusStates"),  
TemplatePart(Name:="PART_Value", Type:=GetType(TextBox)),  
TemplatePart(Name:="PART_FindOrientation", Type:=GetType(ComboBox)),  
TemplatePart(Name:="PART_FindMatchCase", Type:=GetType(CheckBox)),  
TemplatePart(Name:="PART_ListResult", Type:=GetType(ListBox)),  
TemplatePart(Name:="PART_OK", Type:=GetType(Button)),  
TemplatePart(Name:="PART_Cancel", Type:=GetType(Button))>  
Public Class FindXPress  
    Inherits Control
```

#### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:FindXPress ... />
```

## Τί νέο υπάρχει στην έκδοση 1.1

Στην έκδοση 1.1 έχει προστεθεί το **FindXPress** control.

## Διανομή

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

## Επικοινωνία

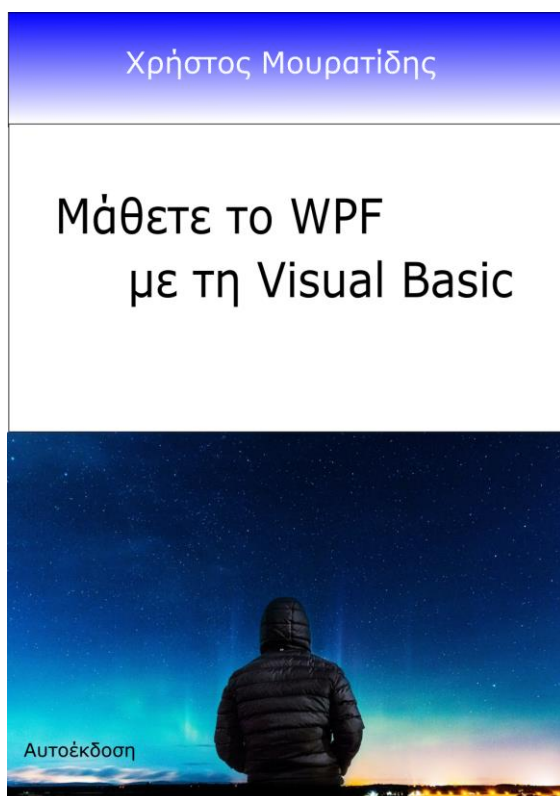
Για οποιαδήποτε πληροφορία ή διευκρίνιση παρακαλώ επικοινωνήστε στο :

[mouratx@yahoo.com](mailto:mouratx@yahoo.com) ή [mouratx@hotmail.com](mailto:mouratx@hotmail.com)



Χρήστος Μουρατίδης,  
Πειραιάς Αύγουστος 2019-2022

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).





## SortControl

Ένα **Control** που παρέχει μία φόρμα (παρόμοια με του Excel) για τη διαμόρφωση, από τον χρήστη, ενός ή περισσότερων επιπέδων ταξινόμησης.

### Σύνταξη:

#### VB:

```
<TemplateVisualState(Name="Normal", GroupName="CommonStates"),
TemplateVisualState(Name="MouseOver", GroupName="CommonStates"),
TemplateVisualState(Name="Disabled", GroupName="CommonStates"),
TemplateVisualState(Name="Focused", GroupName="FocusStates"),
TemplateVisualState(Name="Unfocused", GroupName="FocusStates"),
TemplatePart(Name="PART_AddLevel", Type:=GetType(Button)),
TemplatePart(Name="PART_RemoveLevel", Type:=GetType(Button)),
TemplatePart(Name="PART_OK", Type:=GetType(Button)),
TemplatePart(Name="PART_Cancel", Type:=GetType(Button)),
TemplatePart(Name="PART_SortRowsContainer", Type:=GetType(Grid))>
Public Class SortControl
    Inherits Control
```

#### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:SortControl ... />
```

## Παρατηρήσεις:

Πρόκειται για ένα πολύ ευέλικτο control που παρέχει στον χρήστη μία φόρμα για να διαμορφώσει ένα ή περισσότερα επίπεδα ταξινόμησης. Η φόρμα αυτή, μοιάζει με αυτή του Excel. Κάθε επίπεδο ταξινόμησης είναι μία γραμμή με στοιχεία Πεδίο-Διάταξη. Για παράδειγμα, Επώνυμο-Αύξουσα ή Μισθός-Φθίνουσα.

Επιπλέον, για να τον διευκολύνουμε να μην ξαναδιαμορφώνει εκ νέου τα επίπεδα ταξινόμησης, το control έχει μεθόδους για αποθήκευση και ανάκτηση (απεριόριστες διαμορφώσεις).

Σχεδόν όλα τα γραφικά στοιχεία του control καθώς και τα λεκτικά μπορούν να παραμετροποιηθούν μέσω των αντίστοιχων ιδιοτήτων. Για παράδειγμα, τα λεκτικά των buttons Προσθήκη επιπέδου και Διαγραφή επιπέδου να αλλάξουν σε Add Level και Remove Level. επίσης, μπορούμε να αλλάξουμε και τα styles των buttons αυτών.

## Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε μέσα σε ένα Grid ενός παραθύρου διάφορα controls μεταξύ των οποίων ένα SortControl και δύο oval buttons για αποθήκευση και φόρτωση της διαμόρφωσης. Για τα oval buttons έχουμε κατασκευάσει ένα ειδικό OvalButtonStyle, όπου όλη η δουλειά έχει γίνει μέσα στο ControlTemplate. Παρόμοια, στο τμήμα Window.Resources, θα δούμε κι άλλα custom styles που θα εφαρμοστούν σε αντίστοιχες ιδιότητες του SortControl, ώστε να το παραμετροποιήσουμε σε εμφάνιση.

Μπορούμε να δούμε μερικές από τις ιδιότητες που έχουμε θέσει στο SortControl, βασικότερες των οποίων είναι η FieldsNames (τα ονόματα των πεδίων προς επιλογή, για εμφάνιση), FieldsNamesOriginal (τα ονόματα των πεδίων όπως είναι στη Βάση).

```
<Window x:Class="winSorting"
  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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  xmlns:local="clr-namespace:DataControls_Sample_Project"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

<Window.Resources >

  <!-- Ένα απλό custom Oval Button. -->
  <Style x:Key="OvalButtonsStyle" TargetType="Button">

    <Setter Property="Background" Value="LightGray" />
    <Setter Property="Template" >
```



```
<Setter.Value>
  <ControlTemplate TargetType="{x:Type Button}">
    <Border Name="border" BorderBrush="Black" BorderThickness="1"
      CornerRadius="8"
      Background="{TemplateBinding Background}" >
      <!-- Visual States -->
      <VisualStateManager.VisualStateGroups >
        <VisualStateGroup Name="CommonStates">
          <VisualStateGroup.Transitions >
            <VisualTransition GeneratedDuration="0:0:0.2">
            </VisualTransition>
          </VisualStateGroup.Transitions>
          <VisualState Name="Normal"/>
          <VisualState Name="MouseOver" />
          <VisualState Name="Disabled">
            <Storyboard >
              <ColorAnimationUsingKeyFrames
                Storyboard.TargetName="border"
                Storyboard.TargetProperty="Background.Color" >
                <DiscreteColorKeyFrame KeyTime="0"
Value="{StaticResource {x:Static SystemColors.InactiveBorderColorKey }}" />
              </ColorAnimationUsingKeyFrames>
            </Storyboard>
          </VisualState>
        </VisualStateGroup>
        <VisualStateGroup Name="FocusStates">
          <VisualStateGroup.Transitions >
            <VisualTransition GeneratedDuration="0:0:0.2">
            </VisualTransition>
          </VisualStateGroup.Transitions>
          <VisualState Name="Focused" />
          <VisualState Name="Unfocused"/>
        </VisualStateGroup>
      </VisualStateManager.VisualStateGroups>
      <ContentPresenter Name="presenter"
        Margin="{TemplateBinding Padding}" />
    </Border>
  </ControlTemplate>
</Setter.Value>
</Setter>
<Style.Triggers >
  <Trigger Property="IsMouseOver" Value="True">
    <Setter Property="Background" Value="Yellow"/>
  </Trigger>
</Style.Triggers>
```

```
        <Setter Property="FontWeight" Value="Bold"/>
    </Trigger>
    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Background" Value="Cyan"/>
        <Setter Property="FontWeight" Value="Bold"/>
    </Trigger>

</Style.Triggers>

</Style>

<!-- Ένα style για το OK button-->
<Style x:Key="SortOKButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Green"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

<!-- Ένα style για το Cancel button-->
<Style x:Key="SortCancelButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

<!-- Ένα style για το ComboBoxItem στο FieldNames ComboBox-->
<Style x:Key="FieldsNamesComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

    <Style.Triggers >
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Green"/>
        </Trigger>
    </Style.Triggers>
</Style>

<Style x:Key="FieldsNamesComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource FieldsNamesComboBoxItemStyle }"/>

</Style>

<!-- Ένα style για τον Row Header.-->
<Style x:Key="RowHeaderStyle" TargetType="{x:Type TextBlock }">

    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="FontWeight" Value="Normal"/>

</Style>

</Window.Resources>

<Grid>
```

```

<Grid.RowDefinitions >
  <RowDefinition />
  <RowDefinition Height="Auto"/>
</Grid.RowDefinitions>

<z:SortControl Name="sortControl" Width="Auto" Margin="10"

    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,IS
ACTIVE"

    ButtonOKStyle="{StaticResource SortOKButtonStyle}"
    ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"

    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
    OKClicked="SortControl_OKClicked"
    CancelClicked="SortControl_CancelClicked"/>

<!-- Buttons για Αποθήκευση, και Φόρτωση ταξινόμησης. -->
<StackPanel Grid.Row="1" Margin="10" Orientation="Horizontal" >

  <Button Name="btnSaveSorting"
    Content="Αποθήκευση ταξινόμησης"
    Padding="5" Click="btnSaveSorting_Click"
    ToolTip="Αποθηκεύει την ταξινόμηση στο αρχείο sorting.xaml στον φάκελο της
εφαρμογής."
    Style="{StaticResource OvalButtonsStyle}"/>

  <Button Name="btnLoadSorting"
    Content="Φόρτωση ταξινόμησης"
    Padding="5" Margin="7,0,0,0" Click="btnLoadSorting_Click"
    ToolTip="Φορτώνει την ταξινόμηση από το αρχείο sorting.xaml, που βρίσκεται
στον φάκελο της εφαρμογής."
    Style="{StaticResource OvalButtonsStyle}"/>

</StackPanel>

</Grid>

</Window>

```

Παρακάτω, ο χρήστης έχει κάνει κάποιες επιλογές στο **SortControl**:

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

Το σημαντικό εδώ, είναι ότι όταν ο χρήστης πατήσει το **OK** τότε το control θα επιστρέψει μέσω της ιδιότητας **SortStringOutput** το σχετικό sorting string, που στην προκειμένη περίπτωση είναι "**LASTNAME ASC, FIRSTNAME ASC**". Έτσι, μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Order by μίας SQL select query.

Για διευκόλυνση του προγραμματιστή, ώστε να κρατάει κάπου την διαμόρφωση αυτή για να την θέσει εκ νέου αργότερα στο πρόγραμμα, παρέχεται και η ιδιότητα **SortListOutput** που είναι τύπου **List(of SortRowInfo)**. Το αντικείμενο [SortRowInfo](#), κρατάει το πεδίο εμφάνισης (FieldName), το πεδίο όπως είναι στη Βάση (FieldNameOriginal) και τη διάταξη (OrderValue) κάθε επιπέδου (γραμμής) ταξινόμησης.

## Απαριθμήσεις

Όνομα	Περιγραφή
OrderEnum	Περιλαμβάνει τις <b>τιμές Διάταξης</b> .

## OrderEnum

Περιλαμβάνει διατάξεις ταξινόμησης.

**Σύνταξη:**

**VB:**

```
Public Enum OrderEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Ascending	Αύξουσα
Descending	Φθίνουσα

## Ιδιότητες

Όνομα	Περιγραφή
AscendingText	Καθορίζει το κείμενο (λεκτικό) για την τιμή <b>Ascending</b> όπως θα εμφανίζεται στο σχετικό <b>ComboBox</b> , τύπου <b>String</b> .
ButtonAddLevelText	Καθορίζει το κείμενο (λεκτικό) για το <b>button</b> " <b>Προσθήκη επιπέδου</b> ", τύπου <b>String</b> .
ButtonAddLevelStyle	Καθορίζει το <b>στυλ</b> για το <b>button</b> " <b>Προσθήκη επιπέδου</b> ", τύπου <b>Style</b> .
ButtonCancelText	Καθορίζει το κείμενο (λεκτικό) για το <b>button</b> " <b>Ακύρωση</b> ", τύπου <b>String</b> .
ButtonCancelStyle	Καθορίζει το <b>στυλ</b> για το <b>button</b> " <b>Ακύρωση</b> ", τύπου <b>Style</b> .
ButtonOKText	Καθορίζει το κείμενο (λεκτικό) για το <b>button</b> " <b>OK</b> ", τύπου <b>String</b> .
ButtonOKStyle	Καθορίζει το <b>στυλ</b> για το <b>button</b> " <b>OK</b> ", τύπου <b>Style</b> .
ButtonRemoveLevelText	Καθορίζει το κείμενο (λεκτικό) για το <b>button</b> " <b>Διαγραφή επιπέδου</b> ", τύπου <b>String</b> .
ButtonRemoveLevelStyle	Καθορίζει το <b>στυλ</b> για το <b>button</b> " <b>Διαγραφή επιπέδου</b> ", τύπου <b>Style</b> .
ButtonsLevelContainerBackground	Καθορίζει το <b>χρώμα φόντου</b> για το <b>container</b> των <b>buttons</b> " <b>Προσθήκη επιπέδου</b> " και " <b>Διαγραφή επιπέδου</b> ", τύπου <b>Brush</b> .
ButtonsOKCancelContainerBackground	Καθορίζει το <b>χρώμα φόντου</b> για το <b>container</b> των <b>buttons</b> " <b>OK</b> " και " <b>Ακύρωση</b> ", τύπου <b>Brush</b> .

ButtonsOKCancelWidth	Καθορίζει το πλάτος, σε pixels, των buttons "OK" και "Ακύρωση". Είναι τύπου <b>Double</b> .
ComboBoxFieldsNamesStyle	Καθορίζει το <b>στυλ</b> για το <b>ComboBox</b> των πεδίων ταξινόμησης, τύπου <b>Style</b> .
ComboBoxOrderStyle	Καθορίζει το <b>στυλ</b> για το <b>ComboBox</b> των τιμών διάταξης, τύπου <b>Style</b> .
DescendingText	Καθορίζει το κείμενο (λεκτικό) για την τιμή <b>Descending</b> όπως θα εμφανίζεται στο σχετικό <b>ComboBox</b> , τύπου <b>String</b> .
FieldsColumnHeaderText	Καθορίζει το κείμενο (λεκτικό) για τη στήλη των πεδίων ταξινόμησης, τύπου <b>String</b> .
FieldsNames	Καθορίζει τα ονόματα των πεδίων ταξινόμησης, όπως θα εμφανίζονται στο αντίστοιχο <b>ComboBox</b> . Είναι τύπου <b>String</b> .  Πρέπει να διαχωρίζονται με κόμμα, χωρίς κενά. Για παράδειγμα, "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ". Ο αριθμός των πεδίων πρέπει να είναι ίδιος με αυτόν της ιδιότητας <b>FieldsNamesOriginal</b> .
FieldsNamesOriginal	Καθορίζει τα πραγματικά ονόματα των πεδίων ταξινόμησης, όπως υπάρχουν στη Βάση. Είναι τύπου <b>String</b> .  Πρέπει να διαχωρίζονται με κόμμα, χωρίς κενά. Για παράδειγμα, "LASTNAME,FIRSTNAME,CLASS,DIVISION". Ο αριθμός των πεδίων πρέπει να είναι ίδιος με αυτόν της ιδιότητας <b>FieldsNames</b> . Με βάση αυτά τα πεδία θα σχηματιστεί το <b>SortStringOutput</b> .
FirstRowHeaderText	Καθορίζει το κείμενο (λεκτικό) του <b>Row Header</b> που εμφανίζεται μόνο στο πρώτο επίπεδο (γραμμή) ταξινόμησης στο αριστερό μέρος (με το default γαλάζιο χρώμα). Είναι τύπου <b>String</b> .
GridSortRowsContainerBackground	Καθορίζει το χρώμα φόντου για το <b>container</b> των επιπέδων ταξινόμησης, τύπου <b>Brush</b> .



OrderColumnHeaderText	Καθορίζει το κείμενο (λεκτικό) για τη στήλη των τιμών διάταξης, τύπου <b>String</b> .
RowHeaderText	Καθορίζει το κείμενο (λεκτικό) του Row Header που εμφανίζεται σε κάθε επίπεδο (γραμμή) ταξινόμησης (εκτός του πρώτου) στο αριστερό μέρος. Είναι τύπου <b>String</b> .
RowHeaderTextStyle	Καθορίζει το στυλ του Row Header που εμφανίζεται σε κάθε επίπεδο (γραμμή) ταξινόμησης (εκτός του επιλεγμένου) στο αριστερό μέρος. Είναι τύπου <b>Style</b> .
SelectedRowHeaderTextStyle	Καθορίζει το στυλ του Row Header που εμφανίζεται για το επιλεγμένο επίπεδο (γραμμή) ταξινόμησης στο αριστερό μέρος (με το default γαλάζιο χρώμα). Είναι τύπου <b>Style</b> .
SortListInput	Καθορίζει μία λίστα αντικειμένων <b>SortRowInfo</b> , καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) ταξινόμησης. Είναι τύπου <b>List(of SortRowInfo)</b> .  Θέτοντας αυτήν την ιδιότητα, διαμορφώνονται αυτόματα τα επίπεδα ταξινόμησης.
SortListOutput	Επιστρέφει μία λίστα αντικειμένων <b>SortRowInfo</b> , καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) ταξινόμησης. Είναι τύπου <b>List(of SortRowInfo)</b> .  Αυτή η λίστα διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί εσωτερικά η public μέθοδος <b>CreateSorting</b> .
SortStringOutput	Επιστρέφει ένα <b>string</b> με τα πεδία ταξινόμησης (πραγματικά ονόματα, όπως είναι στη Βάση) και την τιμή διατάξής τους, διαχωρισμένα με κόμμα. Για παράδειγμα, "LASTNAME ASC, FIRSTNAME ASC". Είναι τύπου <b>String</b> .  Αυτή το output string διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί εσωτερικά η public μέθοδος <b>CreateSorting</b> . Μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Order by μίας SQL select query.

## AscendingText

Καθορίζει το κείμενο (λεκτικό) για την τιμή `Ascending` όπως θα εμφανίζεται στο σχετικό `ComboBox`, τύπου `String`.

### Σύνταξη:

#### VB:

```
Public Property AscendingText As String
```

#### Τύπος: `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό για την τιμή `"Ascending"` στο `ComboBox`. Η default τιμή είναι `"Αύξουσα"`.

### Dependency Property Information:

**Identifier field:** `AscendingTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε την τιμή `"Ascending"` στην ιδιότητα `AscendingText`: και `"Descending"` στην ιδιότητα `DescendingText`.

#### XAML:

```
<zeus:SortControl Name="sortControl"  
    AscendingText="Ascending"  
    DescendingText="Descending" ... />
```

#### VB:

```
sortControl.AscendingText = "Ascending"  
sortControl.DescendingText = "Descending"
```

Ταξινόμηση κατά:	Πεδίο	Διάταξη
	ΕΠΩΝΥΜΟ	Ascending

## ButtonAddLevelText

Καθορίζει το κείμενο (λεκτικό) για το button "Προσθήκη επιπέδου", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonAddLevelText As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το button "Προσθήκη επιπέδου". Η default τιμή είναι "Προσθήκη επιπέδου".

### Dependency Property Information:

**Identifier field:** ButtonAddLevelTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "Add Level" για το button "Προσθήκη επιπέδου" και "Remove Level" για το button "Διαγραφή επιπέδου" :

#### XAML:

```
<zeus:SortControl Name="sortControl"
    ButtonAddLevelText="Add Level"
    ButtonRemoveLevelText="Remove Level"... />
```

#### VB:

```
sortControl.ButtonAddLevelText = "Add Level"
sortControl.ButtonRemoveLevelText = "Remove Level"
```

<input type="button" value="+ Add Level"/> <input type="button" value="✗ Remove Level"/>		
	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΕΠΩΝΥΜΟ ▾	Αύξουσα ▾
Έπειτα κατά:	ΟΝΟΜΑ ▾	Αύξουσα ▾
		<input type="button" value="OK"/> <input type="button" value="Ακύρωση"/>

## ButtonAddLevelStyle

Καθορίζει το **στυλ για το button "Προσθήκη επιπέδου"**, τύπου Style.

### Σύνταξη:

#### VB:

```
Public Property ButtonAddLevelStyle As Style
```

#### Τύπος: System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Προσθήκη επιπέδου". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonAddLevelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα "SortLevelButtonsStyle " και το εφαρμόζουμε στα **buttons "Προσθήκη επιπέδου"** και **"Διαγραφή επιπέδου"** :

#### XAML:

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- Ένα style για τα Sort Level buttons-->
    <Style x:Key="SortLevelButtonsStyle" TargetType="{x:Type Button }">

      <Setter Property="FontSize" Value="12"/>
      <Setter Property="FontWeight" Value="Bold"/>
      <Setter Property="FontFamily" Value="Times New Roman"/>

    </Style>

    ...
```

```
</Window.Resources>

<Grid>
...
    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        ButtonAddLevelStyle="{StaticResource SortLevelButtonsStyle}"
        ButtonRemoveLevelStyle="{StaticResource SortLevelButtonsStyle}"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />
</Grid>
</Window>
```

**VB:**

```
Dim sortLevelButtonStyle As New Style(GetType(Button))

With sortLevelButtonStyle
    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 12.0})
    .Setters.Add(New Setter With {.Property = FontWeightProperty,
        .Value = FontWeights.Bold})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty,
        .Value = New FontFamily("Times New Roman")})
End With

With sortControl
    .ButtonAddLevelStyle = sortLevelButtonStyle
    .ButtonRemoveLevelStyle = sortLevelButtonStyle
End With
```

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΜΙΣΘΟΣ	Φθίνουσα
Έπειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Έπειτα κατά:	ΟΝΟΜΑ	Αύξουσα

OK Ακύρωση



## ButtonCancelText

Καθορίζει το κείμενο (λεκτικό) για το **button "Ακύρωση"**, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonCancelText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button "Ακύρωση"**. Η default τιμή είναι "Ακύρωση".

### Dependency Property Information:

**Identifier field:** ButtonCancelTextProperty

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το **button "Ακύρωση"**, το λεκτικό **"Done"** για το **button "OK"** και θέτουμε το πλάτος των **buttons** αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση **Button**.

#### XAML:

```
<zeus:SortControl Name="sortControl"
    ButtonOKText="Done"
    ButtonCancelText="Cancel"
    ButtonsOKCancelWidth="80"
    ... />
```

#### VB:

```
sortControl.ButtonOKText = " Done"
sortControl.ButtonCancelText = "Cancel"
sortControl.ButtonsOKCancelWidth = 80
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Ταξινόμηση κατά:	Πεδίο	Διάταξη	
	ΕΠΩΝΥΜΟ	Αύξουσα	
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα	
<input type="button" value="Done"/> <input type="button" value="Cancel"/>			

## ButtonCancelStyle

Καθορίζει το **στυλ για το button "Ακύρωση"**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ButtonCancelStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Ακύρωση".  
Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonCancelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"SortOKButtonStyle"** και το εφαρμόζουμε στο **button "OK"** και ένα δεύτερο **resource style** με όνομα **"SortCancelButtonStyle"** και το εφαρμόζουμε στο **button "Ακύρωση"**.

**XAML:**

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- Ένα style για το OK button-->
    <Style x:Key="SortOKButtonStyle" TargetType="{x:Type Button}">

      <Setter Property="Foreground" Value="White"/>
      <Setter Property="Background" Value="Green"/>
      <Setter Property="FontSize" Value="16"/>
      <Setter Property="FontFamily" Value="Times New Roman"/>

    </Style>
```

```

<!-- Ένα style για το Cancel button-->
<Style x:Key="SortCancelButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />

</Grid>

</Window>

```

**VB:**

```

Dim sortOKButtonStyle As New Style(GetType(Button))
Dim sortCancelButtonStyle As New Style(GetType(Button))

With sortOKButtonStyle

    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.Property = ForegroundProperty,
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.Property = BackgroundProperty,
        .Value = Brushes.Green})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty,
        .Value = New FontFamily("Times New Roman")})

End With

With sortCancelButtonStyle

    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.Property = ForegroundProperty,
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.Property = BackgroundProperty,
        .Value = Brushes.Red})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty,

```

```
.Value = New FontFamily("Times New Roman"))}
```

```
End With
```

```
With sortControl
```

```
    .ButtonOKStyle = sortOKButtonStyle  
+    .ButtonCancelStyle = sortCancelButtonStyle
```

```
End With
```

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	Φθίνουσα
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

OK Ακύρωση

## ButtonOKText

Καθορίζει το κείμενο (λεκτικό) για το button "OK", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonOKText As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για το button "OK".  
Η default τιμή είναι "OK".

### Dependency Property Information:

**Identifier field:** ButtonOKProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "Cancel" για το button "Ακύρωση", το λεκτικό "Done" για το button "OK" και θέτουμε το πλάτος των buttons αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση Button.

#### XAML:

```
<zeus:SortControl Name="sortControl"  
    ButtonOKText="Done"  
    ButtonCancelText="Cancel"  
    ButtonsOKCancelWidth="80"  
    ... />
```

#### VB:

```
sortControl.ButtonOKText = " Done"  
sortControl.ButtonCancelText = "Cancel"  
sortControl.ButtonsOKCancelWidth = 80
```

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Έπειτα κατά:	ΟΝΟΜΑ	Αύξουσα

## ButtonOKStyle

Καθορίζει το **στυλ για το button "OK"**, τύπου Style.

### Σύνταξη:

#### VB:

```
Public Property ButtonOKStyle As Style
```

#### Τύπος: System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "OK".  
Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonOKStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"SortOKButtonStyle"** και το εφαρμόζουμε στο **button "OK"** και ένα δεύτερο **resource style** με όνομα **"SortCancelButtonStyle"** και το εφαρμόζουμε στο **button "Ακύρωση"**.

#### XAML:

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- Ένα style για το OK button-->
    <Style x:Key="SortOKButtonStyle" TargetType="{x:Type Button}">

      <Setter Property="Foreground" Value="White"/>
      <Setter Property="Background" Value="Green"/>
      <Setter Property="FontSize" Value="16"/>
      <Setter Property="FontFamily" Value="Times New Roman"/>

    </Style>
```



```

<!-- Ένα style για το Cancel button-->
<Style x:Key="SortCancelButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />

</Grid>

</Window>

```

**VB:**

```

Dim sortOKButtonStyle As New Style(GetType(Button))
Dim sortCancelButtonStyle As New Style(GetType(Button))

With sortOKButtonStyle

    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.Property = ForegroundProperty,
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.Property = BackgroundProperty,
        .Value = Brushes.Green})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty,
        .Value = New FontFamily("Times New Roman")})

End With

With sortCancelButtonStyle

    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.Property = ForegroundProperty,
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.Property = BackgroundProperty,
        .Value = Brushes.Red})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty,

```

```
.Value = New FontFamily("Times New Roman"))}
```

```
End With
```

```
With sortControl
```

```
    .ButtonOKStyle = sortOKButtonStyle
```

```
    .ButtonCancelStyle = sortCancelButtonStyle
```

```
End With
```

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	Φθίνουσα
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

OK Ακύρωση

## ButtonRemoveLevelText

Καθορίζει το κείμενο (λεκτικό) για το button "Διαγραφή επιπέδου", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonRemoveLevelText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το button "Διαγραφή επιπέδου". Η default τιμή είναι "Διαγραφή επιπέδου".

### Dependency Property Information:

**Identifier field:** `ButtonRemoveLevelTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "Add Level" για το button "Προσθήκη επιπέδου" και "Remove Level" για το button "Διαγραφή επιπέδου" :

#### XAML:

```
<zeus:SortControl Name="sortControl"  
    ButtonAddLevelText="Add Level"  
    ButtonRemoveLevelText="Remove Level"... />
```

#### VB:

```
sortControl.ButtonAddLevelText = "Add Level"  
sortControl.ButtonRemoveLevelText = "Remove Level"
```

<input type="button" value="+ Add Level"/> <input type="button" value="✗ Remove Level"/>		
Ταξινόμηση κατά:	Πεδίο	Διάταξη
	ΕΠΩΝΥΜΟ	Αύξουσα
Έπειτα κατά:	ΟΝΟΜΑ	Αύξουσα
<input type="button" value="OK"/> <input type="button" value="Ακύρωση"/>		

## ButtonRemoveLevelStyle

Καθορίζει το **στυλ για το button "Διαγραφή επιπέδου"**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ButtonRemoveLevelStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Διαγραφή επιπέδου". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonRemoveLevelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"SortLevelButtonsStyle "** και το εφαρμόζουμε στα **buttons "Προσθήκη επιπέδου"** και **"Διαγραφή επιπέδου"** :

**XAML:**

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- Ένα style για τα Sort Level buttons-->
    <Style x:Key="SortLevelButtonsStyle" TargetType="{x:Type Button }">

      <Setter Property="FontSize" Value="12"/>
      <Setter Property="FontWeight" Value="Bold"/>
      <Setter Property="FontFamily" Value="Times New Roman"/>

    </Style>

    ...
```

```
</Window.Resources>

<Grid>
    ...
    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        ButtonAddLevelStyle="{StaticResource SortLevelButtonsStyle}"
        ButtonRemoveLevelStyle="{StaticResource SortLevelButtonsStyle}"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />
</Grid>
</Window>
```

**VB:**

```
Dim sortLevelButtonStyle As New Style(GetType(Button))

With sortLevelButtonStyle
    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 12.0})
    .Setters.Add(New Setter With {.Property = FontWeightProperty,
        .Value = FontWeights.Bold})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty,
        .Value = New FontFamily("Times New Roman")})
End With

With sortControl
    .ButtonAddLevelStyle = sortLevelButtonStyle
    .ButtonRemoveLevelStyle = sortLevelButtonStyle
End With
```

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΜΙΣΘΟΣ	Φθίνουσα
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

OK Ακύρωση

## ButtonsLevelContainerBackground

Καθορίζει το **χρώμα φόντου** για το **container** των **buttons** "Προσθήκη επιπέδου" και "Διαγραφή επιπέδου", τύπου **Brush**.

### Σύνταξη:

#### VB:

```
Public Property ButtonsLevelContainerBackground As Brush
```

#### Τύπος: System.Windows.Media.Brush

Προσδιορίζουμε ένα αντικείμενο **Brush** για τον χρωματισμό του φόντου του **container** των **buttons** "Προσθήκη επιπέδου" και "Διαγραφή επιπέδου".  
Η default τιμή είναι `Brushes.LightGray`.

### Dependency Property Information:

**Identifier field:** ButtonsLevelContainerBackgroundProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το **χρώμα φόντου του container** των **buttons** επιπέδου ταξινόμησης σε μία **απόχρωση του πορτοκαλί (Amber)**.



#### XAML:

```
<zeus:SortControl Name="sortControl" Width="Auto"  
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"  
    FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"  
    ButtonsLevelContainerBackground="#FFD54F"  
    ... />
```

#### VB:

```
sortControl.ButtonsLevelContainerBackground = _  
    New SolidColorBrush(ColorConverter.ConvertFromString("#FFD54F"))
```



 Προσθήκη επιπέδου Διαγραφή επιπέδου

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	<input type="text" value=""/>	<input type="text" value="Αύξουσα"/>

OKΑκύρωση

**Σημείωση:** Ένα καλό site για τιμές χρωμάτων είναι το <https://www.materialui.co/colors>

## ButtonsOKCancelContainerBackground

Καθορίζει το **χρώμα φόντου για το container των buttons "OK" και "Ακύρωση"**, τύπου **Brush**.

### Σύνταξη:

#### VB:

```
Public Property ButtonsOKCancelContainerBackground As Brush
```

**Τύπος:** **System.Windows.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο Brush για τον χρωματισμό του φόντου του container των buttons "OK" και "Ακύρωση". Η default τιμή είναι Brushes.LightGray.

### Dependency Property Information:

**Identifier field:** ButtonsOKCancelContainerBackgroundProperty

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το **χρώμα φόντου του container των buttons "OK" και "Ακύρωση"** σε μία απόχρωση του μωβ (Purple).

#### XAML:

```
<zeus:SortControl Name="sortControl" Width="Auto"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
    FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
    ButtonsOKCancelContainerBackground= "#CE93D8"
    ... />
```

#### VB:

```
sortControl.ButtonsOKCancelContainerBackground = _
    New SolidColorBrush(ColorConverter.ConvertFromString("#CE93D8"))
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Ταξινόμηση κατά:	Πεδίο	Διάταξη	
	ΕΠΩΝΥΜΟ	Αύξουσα	
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα	

OK Ακύρωση

**Σημείωση:** Ένα καλό site για τιμές χρωμάτων είναι το <https://www.materialui.co/colors>

## ButtonsOKCancelWidth

Καθορίζει το πλάτος, σε pixels, των **buttons "OK" και "Ακύρωση"**. Είναι τύπου **Double**.

### Σύνταξη:

#### VB:

```
Public Property ButtonsOKCancelWidth As Double
```

#### Τύπος: System.Double

Προσδιορίζουμε μία τιμή Double που καθορίζει το πλάτος για τα buttons "OK" και "Ακύρωση". Η default τιμή είναι 100.

### Dependency Property Information:

**Identifier field:** ButtonsOKCancelWidthProperty

### Παρατηρήσεις:

Ο λόγος που υπάρχει αυτή η ιδιότητα είναι για να καθορίσουμε γρήγορα ένα ομοιόμορφο πλάτος για τα buttons "OK" και "Ακύρωση".

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το **button "Ακύρωση"**, το λεκτικό **"Done"** για το **button "OK"** και θέτουμε το **πλάτος αυτών των buttons** σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση Button.

#### XAML:

```
<zeus:SortControl Name="sortControl"
  ButtonOKText="Done"
  ButtonCancelText="Cancel"
  ButtonsOKCancelWidth="80"
  ... />
```

#### VB:

```
sortControl.ButtonOKText = " Done"
sortControl.ButtonCancelText = "Cancel"
sortControl.ButtonsOKCancelWidth = 80
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Ταξινόμηση κατά:	Πεδίο	Διάταξη	
	ΕΠΩΝΥΜΟ	Αύξουσα	
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα	

Done Cancel

## ComboBoxFieldsNamesStyle

Καθορίζει το **στυλ για το ComboBox των πεδίων ταξινόμησης**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ComboBoxFieldsNamesStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο ComboBox των πεδίων ταξινόμησης. Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxFieldsNamesStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"FieldsNamesComboBoxItemStyle"** που αφορά το **style** για ένα **ComboBoxItem** και ένα δεύτερο **resource style** με όνομα **"FieldsNamesComboBoxStyle"** που αφορά το **style** ενός **ComboBox**. Το πρώτο, κάνει ένα **ComboBoxItem** να είναι **έντονο πράσινο και μεγαλύτερο μέγεθος όταν ο δείκτης του ποντικιού είναι πάνω του**. Το δεύτερο, χρησιμοποιεί το πρώτο και στο τέλος εφαρμόζεται στην ιδιότητα **ComboBoxFieldsNamesStyle** με μία **StaticResource** έκφραση.

#### XAML:

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- To style για το ComboBoxItem-->
    <Style x:Key="FieldsNamesComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

      <Style.Triggers >

        <Trigger Property="IsMouseOver" Value="True">
```

```

        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Foreground" Value="Green"/>
    </Trigger>

</Style.Triggers>

</Style>

<!-- To style του FieldsNames ComboBox-->
<Style x:Key="FieldsNamesComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource FieldsNamesComboBoxItemStyle }"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />

</Grid>

</Window>

```

**VB:**

```

Dim FieldsNamesComboBoxItemStyle As New Style(GetType(ComboBoxItem))
Dim FieldsNamesComboBoxStyle As New Style(GetType(ComboBox))

Dim trigger As New Trigger With {.[Property] = IsMouseOverProperty, .Value = True}

With trigger.Setters

    .Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Add(New Setter With {.[Property] = ForegroundProperty, _
        .Value = Brushes.Green})
    .Add(New Setter With {.[Property] = FontWeightProperty, _
        .Value = FontWeights.Bold})

End With

FieldsNamesComboBoxItemStyle.Triggers.Add(trigger)

```

```
FieldsNamesComboBoxStyle.Setters.Add(New Setter  
    With {.Property = ItemsControl.ItemContainerStyleProperty, _  
        .Value = FieldsNamesComboBoxItemStyle})
```

```
sortControl.ComboBoxFieldsNamesStyle = FieldsNamesComboBoxStyle
```

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:		Αύξουσα
	ΕΠΩΝΥΜΟ	
	<b>ΟΝΟΜΑ</b>	
	ΠΑΤΡΩΝΥΜΟ	
	ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	
	ΜΙΣΘΟΣ	

OK Ακύρωση



## ComboBoxOrderStyle

Καθορίζει το **στυλ για το ComboBox των τιμών διάταξης**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ComboBoxOrderStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο ComboBox των τιμών διάταξης (order). Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxOrderStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"OrderComboBoxItemStyle"** που αφορά το **style** για ένα **ComboBoxItem** και ένα δεύτερο **resource style** με όνομα **"OrderComboBoxStyle"** που αφορά το **style** ενός **ComboBox**. Το πρώτο, κάνει ένα **ComboBoxItem** να είναι **έντονο μωβ και μεγαλύτερο μέγεθος όταν ο δείκτης του ποντικιού είναι πάνω του**. Το δεύτερο, χρησιμοποιεί το πρώτο και στο τέλος εφαρμόζεται στην ιδιότητα **ComboBoxOrderStyle** με μία **StaticResource** έκφραση.

#### XAML:

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- To style για το ComboBoxItem-->
    <Style x:Key="OrderComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

      <Style.Triggers >

        <Trigger Property="IsMouseOver" Value="True">
```

```

        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Foreground" Value="Magenta"/>
    </Trigger>

</Style.Triggers>

</Style>

<!-- To style του Order ComboBox-->
<Style x:Key="OrderComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource OrderComboBoxItemStyle }"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ComboBoxOrderStyle="{StaticResource OrderComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />

</Grid>

</Window>

```

**VB:**

```

Dim OrderComboBoxItemStyle As New Style(GetType(ComboBoxItem))
Dim OrderComboBoxStyle As New Style(GetType(ComboBox))

Dim trigger As New Trigger With {.[Property] = IsMouseOverProperty, .Value = True}

With trigger.Setters

    .Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Add(New Setter With {.[Property] = ForegroundProperty, _
        .Value = Brushes.Magenta})
    .Add(New Setter With {.[Property] = FontWeightProperty, _
        .Value = FontWeights.Bold})

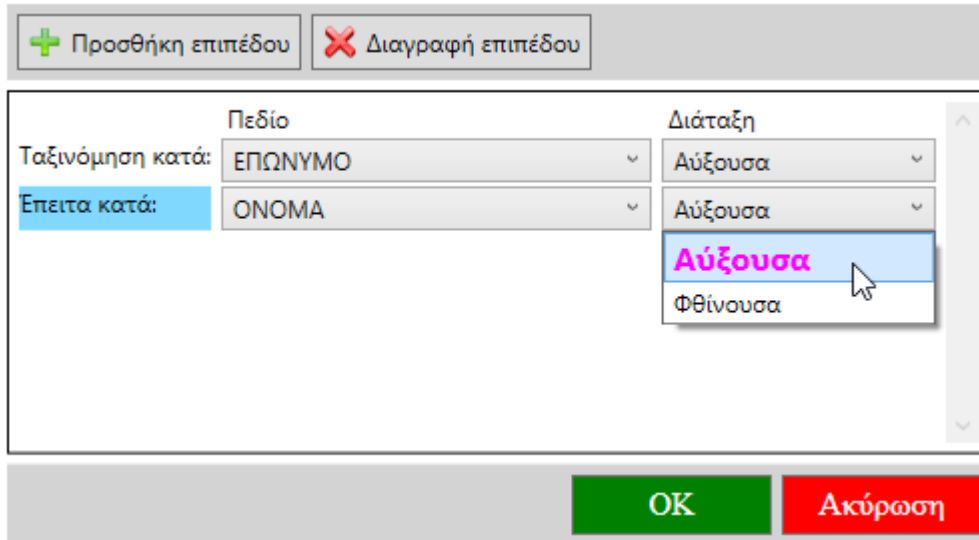
End With

OrderComboBoxItemStyle.Triggers.Add(trigger)

```

```
OrderComboBoxStyle.Setters.Add(New Setter  
    With {.Property = ItemsControl.ItemContainerStyleProperty, _  
        .Value = FieldsNamesComboBoxItemStyle})
```

```
sortControl.ComboBoxOrderStyle = OrderComboBoxStyle
```



## DescendingText

Καθορίζει το κείμενο (λεκτικό) για την τιμή **Descending** όπως θα εμφανίζεται στο σχετικό **ComboBox**, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property DescendingText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για την τιμή "Descending" στο **ComboBox**. Η default τιμή είναι "Φθίνουσα".

### Dependency Property Information:

**Identifier field:** `DescendingTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε την τιμή "Ascending" στην ιδιότητα **AscendingText**: και "Descending" στην ιδιότητα **DescendingText**.

#### XAML:

```
<zeus:SortControl Name="sortControl"  
    AscendingText="Ascending"  
    DescendingText="Descending" ... />
```

#### VB:

```
sortControl.AscendingText = "Ascending"  
sortControl.DescendingText = "Descending"
```

Ταξινόμηση κατά:	Πεδίο	Διάταξη
	ΕΠΩΝΥΜΟ	Ascending

## FieldsColumnHeaderText

Καθορίζει το κείμενο (λεκτικό) για τη στήλη των πεδίων ταξινόμησης, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property FieldsColumnHeaderText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό της επικεφαλίδας της στήλης των πεδίων ταξινόμησης. Η default τιμή είναι "Πεδίο".

### Dependency Property Information:

**Identifier field:** `FieldsColumnHeaderTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Field"** για την επικεφαλίδα της στήλης των πεδίων ταξινόμησης και το λεκτικό **"Order"** για την επικεφαλίδα της στήλης της διάταξης.

#### XAML:



```
<zeus:SortControl Name="sortControl"  
    FieldsColumnHeaderText="Field"  
    OrderColumnHeaderText="Order" ... />
```

#### VB:


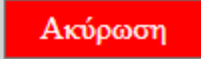
```
With sortControl
```

```
    .FieldsColumnHeaderText = "Field"  
    .OrderColumnHeaderText = "Order"
```

```
End With
```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Field	Order
Ταξινόμηση κατά:	ΜΙΣΘΟΣ	Φθίνουσα
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

## FieldsNames

Καθορίζει τα ονόματα των πεδίων ταξινόμησης, όπως θα εμφανίζονται στο αντίστοιχο **ComboBox**. Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FieldsNames As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που περιέχει, διαχωρισμένα με κόμμα, τα ονόματα των πεδίων ταξινόμησης. Η default τιμή είναι **String.Empty**.

### Dependency Property Information:

**Identifier field:** `FieldsNamesProperty`

### Παρατηρήσεις:

Πρόκειται για τα **display names** των πεδίων ταξινόμησης, δηλαδή όπως θα εμφανίζονται στο **ComboBox**. Τα πραγματικά ονόματα, δηλαδή όπως έχουν οριστεί στην πηγή (π.χ. στη Βάση) καθορίζονται στην άλλη ιδιότητα **FieldsNamesOriginal**.

Να σημειώσουμε ότι, εκατέρωθεν του κόμματος δεν πρέπει να υπάρχουν κενά. Για παράδειγμα, η τιμή "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ" είναι έγκυρη ενώ η "ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΠΑΤΡΩΝΥΜΟ" όχι. Συνεπώς, θα πρέπει να είμαστε προσεκτικοί στον καθορισμό της ιδιότητας. Φυσικά, το όνομα ενός πεδίου μπορεί να έχει κενά. Για παράδειγμα, "ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ, ΜΙΣΘΟΣ". Εσωτερικά το **SortControl** τοποθετεί αγκύλες [ ], που ως γνωστόν χρησιμοποιούν οι **Access**, **SQLite** και **Sql Server**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα πεδία ταξινόμησης "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;" για επιλογή από τον χρήστη. Τα πραγματικά ονόματα, όμως, των πεδίων ταξινόμησης είναι "LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,IS ACTIVE" που ορίζονται στην ιδιότητα **FieldsNamesOriginal**:



**XAML:**

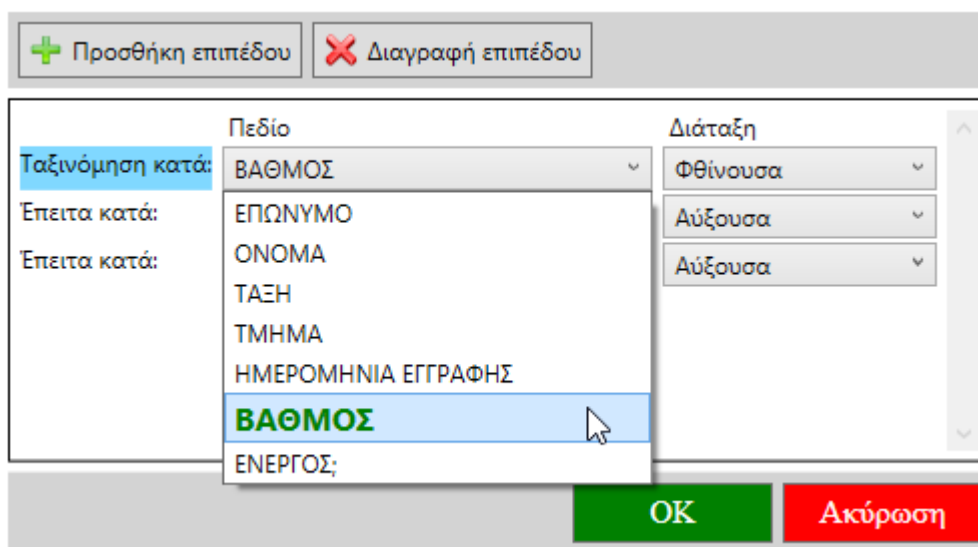
```
<zeus:SortControl Name="sortControl" Name="sortControl" Width="Auto" Margin="10"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    ButtonOKStyle="{StaticResource SortOKButtonStyle}"
    ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
    OKClicked="SortControl_OKClicked"
    CancelClicked="SortControl_CancelClicked"/>
```

**VB:**

With sortControl

```
.FieldsNames = "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
.FieldsNamesOriginal = "LASTNAME,FIRSTNAME,CLASS,DIVISION,
    REGISTRATION DATE,GRADE,IS ACTIVE"
```

End With



## FieldsNamesOriginal

Καθορίζει τα πραγματικά ονόματα των πεδίων ταξινόμησης, όπως υπάρχουν στη Βάση. Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FieldsNamesOriginal As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που περιέχει, διαχωρισμένα με κόμμα, τα πραγματικά ονόματα των πεδίων ταξινόμησης. Η default τιμή είναι **String.Empty**.

### Dependency Property Information:

**Identifier field:** `FieldsNamesOriginalProperty`

### Παρατηρήσεις:

Πρόκειται για τα πραγματικά ονόματα των πεδίων ταξινόμησης, δηλαδή όπως υπάρχουν στη Βάση. Για τα display ονόματα, δηλαδή όπως αυτά θα εμφανιστούν προς επιλογή στο σχετικό **ComboBox** καθορίζονται στην άλλη ιδιότητα **FieldsNames**.

Να σημειώσουμε ότι, εκατέρωθεν του κόμματος δεν πρέπει να υπάρχουν κενά. Για παράδειγμα, η τιμή `"FIRSTNAME, LASTNAME, FATHER NAME"` είναι έγκυρη ενώ η `"FIRSTNAME, LASTNAME, FATHER NAME"` όχι. Συνεπώς, θα πρέπει να είμαστε προσεκτικοί στον καθορισμό της ιδιότητας. Φυσικά, το όνομα ενός πεδίου μπορεί να έχει κενά., όπως το `"FATHER NAME"`. Εσωτερικά το **SortControl** τοποθετεί αγκύλες [ ], που ως γνωστόν χρησιμοποιούν οι **Access**, **SQLite** και **Sql Server**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα πεδία ταξινόμησης `"ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΤΑΞΗ, ΤΜΗΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ, ΒΑΘΜΟΣ, ΕΝΕΡΓΟΣ;"` για επιλογή από τον χρήστη. Τα πραγματικά ονόματα, όμως, των πεδίων ταξινόμησης είναι `"LASTNAME, FIRSTNAME, CLASS, DIVISION, REGISTRATION DATE, GRADE, IS ACTIVE"` που ορίζονται στην ιδιότητα **FieldsNamesOriginal**:

**XAML:**

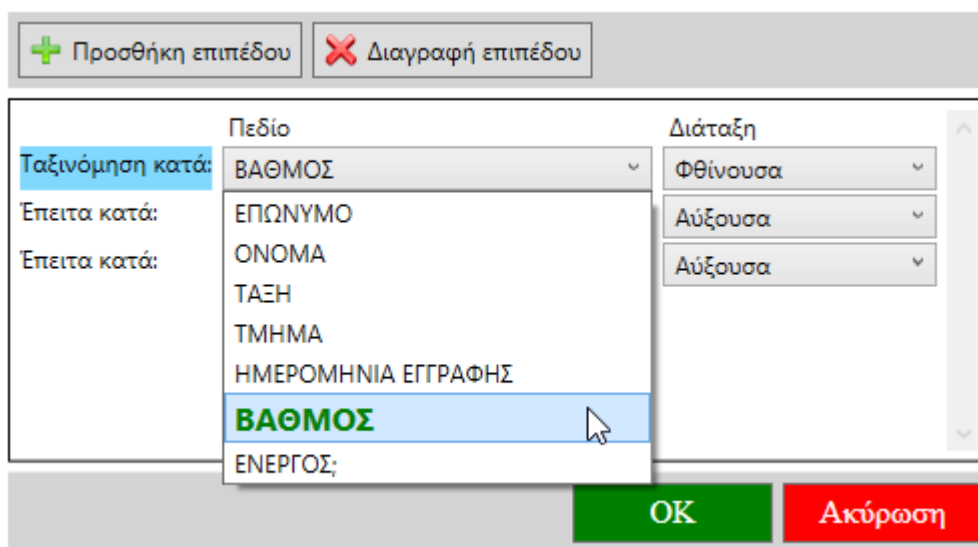
```
<zeus:SortControl Name="sortControl" Name="sortControl" Width="Auto" Margin="10"
  FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
  FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
  IS ACTIVE"
  ButtonOKStyle="{StaticResource SortOKButtonStyle}"
  ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
  ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
  RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
  OKClicked="SortControl_OKClicked"
  CancelClicked="SortControl_CancelClicked"/>
```

**VB:**

With sortControl

```
.FieldsNames = "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
.FieldsNamesOriginal = "LASTNAME,FIRSTNAME,CLASS,DIVISION,
  REGISTRATION DATE,GRADE,IS ACTIVE"
```

End With



## FirstRowHeaderText

Καθορίζει το κείμενο (λεκτικό) του Row Header που εμφανίζεται μόνο στο πρώτο επίπεδο (γραμμή) ταξινόμησης στο αριστερό μέρος (με το default γαλάζιο χρώμα). Είναι τύπου `String`.

### Σύνταξη:

#### VB:

```
Public Property FirstRowHeaderText As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό για την επικεφαλίδα του πρώτου επιπέδου (γραμμής) ταξινόμησης. Η default τιμή είναι "Ταξινόμηση κατά:".

### Dependency Property Information:

**Identifier field:** `FirstRowHeaderTextProperty`

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε την τιμή "Sort by:" στην ιδιότητα `FirstRowHeaderText` και "Then by:" στην ιδιότητα `RowHeaderText`. Έτσι, το πρώτο επίπεδο (γραμμή) ταξινόμησης θα έχει επικεφαλίδα "Sort by:" ενώ τα επόμενα "Then by:"

#### XAML:

```
<zeus:SortControl Name="sortControl"
    FirstRowHeaderText="Sort by:"
    RowHeaderText="Then by:" ... />
```

#### VB:

```
With sortControl
    .FirstRowHeaderText = "Sort by:"
    .RowHeaderText = "Then by:"
End With
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Sort by:	Πεδίο ΕΠΩΝΥΜΟ	Διάταξη Αύξουσα	
Then by:	ΟΝΟΜΑ	Αύξουσα	
^			
v			
		<b>OK</b>	<b>Ακύρωση</b>

## GridSortRowsContainerBackground

Καθορίζει το **χρώμα φόντου** για το **container των επιπέδων ταξινόμησης**,, τύπου **Brush**.

### Σύνταξη:

#### VB:

```
Public Property GridSortRowsContainerBackground As Brush
```

**Τύπος:** **System.Windows.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο Brush για τον χρωματισμό του φόντου του container των επιπέδων ταξινόμησης. Η default τιμή είναι Brushes.Transparent.

### Dependency Property Information:

**Identifier field:** GridSortRowsContainerBackgroundProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το **χρώμα φόντου του container των επιπέδων ταξινόμησης** σε μία **απόχρωση του ανοιχτού πράσινου (Light Green)**.

#### XAML:

```
<zeus:SortControl Name="sortControl" Width="Auto"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
    FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
    GridSortRowsContainerBackground="#8BC34A"
    ... />
```

#### VB:

```
sortControl.GridSortRowsContainerBackground = _
    New SolidColorBrush(ColorConverter.ConvertFromString("#8BC34A"))
```

+ Προσθήκη επιπέδου ✖ Διαγραφή επιπέδου

	Πεδίο	Διάταξη
Sort by:	ΕΠΩΝΥΜΟ	Αύξουσα
Then by:	ΟΝΟΜΑ	Αύξουσα

OK Ακύρωση

**Σημείωση:** Ένα καλό site για τιμές χρωμάτων είναι το <https://www.materialui.co/colors>

## OrderColumnHeaderText

Καθορίζει το κείμενο (λεκτικό) για τη στήλη των τιμών διάταξης, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property OrderColumnHeaderText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό της επικεφαλίδας της στήλης της διάταξης (**order**). Η default τιμή είναι "Διάταξη".

### Dependency Property Information:

**Identifier field:** OrderColumnHeaderTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Field"** για την επικεφαλίδα της στήλης των πεδίων ταξινόμησης και το λεκτικό **"Order"** για την επικεφαλίδα της στήλης της διάταξης.

#### XAML:

```
<zeus:SortControl Name="sortControl"  
    FieldsColumnHeaderText="Field"  
    OrderColumnHeaderText="Order" ... />
```



#### VB:

```
With sortControl
```

```
    .FieldsColumnHeaderText = "Field"  
    .OrderColumnHeaderText = "Order"
```

```
End With
```



 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
	Field	Order	
Ταξινόμηση κατά:	ΜΙΣΘΟΣ	Φθίνουσα	
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα	
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα	

OK Ακύρωση

## RowHeaderText

Καθορίζει το κείμενο (λεκτικό) του Row Header που εμφανίζεται σε κάθε επίπεδο (γραμμή) ταξινόμησης (εκτός του πρώτου) στο αριστερό μέρος. Είναι τύπου `String`.

### Σύνταξη:

#### VB:

```
Public Property RowHeaderText As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό για την επικεφαλίδα των επιπέδων (γραμμών) ταξινόμησης, εκτός του πρώτου. Η default τιμή είναι "Έπειτα κατά:".

### Dependency Property Information:

**Identifier field:** `RowHeaderTextProperty`

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε την τιμή "Sort by:" στην ιδιότητα `FirstRowHeaderText` και "Then by:" στην ιδιότητα `RowHeaderText`. Έτσι, το πρώτο επίπεδο (γραμμή) ταξινόμησης θα έχει επικεφαλίδα "Sort by:" ενώ τα επόμενα "Then by:"

#### XAML:

```
<zeus:SortControl Name="sortControl"
    FirstRowHeaderText="Sort by:"
    RowHeaderText="Then by:" ... />
```

#### VB:

```
With sortControl
    .FirstRowHeaderText = "Sort by:"
    .RowHeaderText = "Then by:"
End With
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Sort by:	Πεδίο ΕΠΩΝΥΜΟ	Διάταξη	Αύξουσα
Then by:	ΟΝΟΜΑ	Αύξουσα	

OK    Ακύρωση

## RowHeaderTextStyle

Καθορίζει το **στυλ του Row Header** που εμφανίζεται σε κάθε επίπεδο (γραμμή) ταξινόμησης (εκτός του επιλεγμένου) στο αριστερό μέρος, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property RowHeaderTextStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα **Style** που θα εφαρμοστεί στο **Row Header** σε κάθε επίπεδο ταξινόμησης (εκτός του επιλεγμένου, για το οποίο υπάρχει αντίστοιχη ιδιότητα). Το default style είναι αυτό ενός **TextBlock** element.

### Dependency Property Information:

**Identifier field:** `RowHeaderTextStyleProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"RowHeaderStyle"** που αφορά το **style** για για τα **row headers** των επιπέδων (γραμμών) ταξινόμησης πλην του επιλεγμένου. Για το **επιλεγμένο row header** καθορίζουμε ένα **ξεχωριστό resource style** με όνομα **"SelectedRowHeaderStyle"**.

### XAML:

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

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

      <Setter Property="Foreground" Value="Blue"/>
      <Setter Property="FontWeight" Value="Normal"/>

    </Style>

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

```
<Setter Property="Background" Value="Green"/>
<Setter Property="Foreground" Value="White" />
<Setter Property="FontWeight" Value="Normal" />

</Style>

...

</Window.Resources>

<Grid>

...

<zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
    FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
    RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
    SelectedRowHeaderTextStyle="{StaticResource SelectedRowHeaderStyle}"
    ButtonOKStyle="{StaticResource SortOKButtonStyle}"
    ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
    OKClicked="SortControl_OKClicked"
    CancelClicked="SortControl_CancelClicked" />

</Grid>

</Window>
```

**VB:**

```
Dim rowHeaderStyle As New Style(GetType(TextBlock))
Dim selectedRowHeaderStyle As New Style(GetType(TextBlock))

With rowHeaderStyle

    .Setters.Add(New Setter With {.Property = ForegroundProperty, _
        .Value = Brushes.Blue})
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _
        .Value = FontWeights.Normal})

End With

With selectedRowHeaderStyle

    .Setters.Add(New Setter With {.Property = ForegroundProperty, _
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.Property = TextBlock.BackgroundProperty, _
        .Value = Brushes.Green})
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _
        .Value = FontWeights.Normal})

End With

With sortControl
```

```
.RowHeaderTextStyle = rowHeaderStyle  
.SelectedRowHeaderTextStyle = selectedRowHeaderStyle
```

End With

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Έπειτα κατά:	ΟΝΟΜΑ	Αύξουσα
Έπειτα κατά:	ΕΝΕΡΓΟΣ;	Αύξουσα

OK Ακύρωση

## SelectedRowHeaderTextStyle

Καθορίζει το **στυλ του Row Header** που εμφανίζεται στο επιλεγμένο επίπεδο (γραμμή) ταξινόμησης, τύπου **Style**.

**Σύνταξη:**

**VB:**

```
Public Property SelectedRowHeaderTextStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα **Style** που θα εφαρμοστεί στο **Row Header** στο επιλεγμένο ταξινόμησης. Το default style είναι αυτό ενός **TextBlock element** με γαλάζιο φόντο.

### Dependency Property Information:

**Identifier field:** SelectedRowHeaderTextStyleProperty

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"RowHeaderStyle"** που αφορά το **style** για για τα row headers των επιπέδων (γραμμών) ταξινόμησης πλην του επιλεγμένου. Για το **επιλεγμένο row header** καθορίζουμε ένα **ξεχωριστό resource style** με όνομα **"SelectedRowHeaderStyle"**.

**XAML:**

```
<Window x:Class="winSorting"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Ταξινόμηση" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <Style x:Key="RowHeaderStyle" TargetType="{x:Type TextBlock }">
      <Setter Property="Foreground" Value="Blue"/>
      <Setter Property="FontWeight" Value="Normal"/>
    </Style>

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

```

    <Setter Property="Background" Value="Green"/>
    <Setter Property="Foreground" Value="White" />
    <Setter Property="FontWeight" Value="Normal" />

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:SortControl Name="sortControl" Width="Auto" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ,ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ,ΜΙΣΘΟΣ"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,FATHER NAME,DATE BIRTH,SALARY"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        SelectedRowHeaderTextStyle="{StaticResource SelectedRowHeaderStyle}"
        ButtonOKStyle="{StaticResource SortOKButtonStyle}"
        ButtonCancelStyle="{StaticResource SortCancelButtonStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        RowHeaderTextStyle="{StaticResource RowHeaderStyle}"
        OKClicked="SortControl_OKClicked"
        CancelClicked="SortControl_CancelClicked" />

</Grid>

</Window>

```

**VB:**

```

Dim rowHeaderStyle As New Style(GetType(TextBlock))
Dim selectedRowHeaderStyle As New Style(GetType(TextBlock))

With rowHeaderStyle

    .Setters.Add(New Setter With {.Property = ForegroundProperty, _
        .Value = Brushes.Blue})
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _
        .Value = FontWeights.Normal})

End With

With selectedRowHeaderStyle

    .Setters.Add(New Setter With {.Property = ForegroundProperty, _
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.Property = TextBlock.BackgroundProperty, _
        .Value = Brushes.Green})
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _
        .Value = FontWeights.Normal})

End With

With sortControl

    .RowHeaderTextStyle = rowHeaderStyle

```



`.SelectedRowHeaderTextStyle = selectedRowHeaderStyle`

End With

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Έπειτα κατά:	ΟΝΟΜΑ	Αύξουσα
Έπειτα κατά:	ΕΝΕΡΓΟΣ;	Αύξουσα

OK Ακύρωση

## SortListInput

Καθορίζει μία λίστα αντικειμένων `SortRowInfo`, καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) ταξινόμησης. Είναι τύπου `List(of SortRowInfo)`.

### Σύνταξη:

#### VB:

```
Public Property SortListInput As List(Of SortRowInfo)
```

#### Τύπος: `System.Collections.Generic.List(Of T)`

Προσδιορίζουμε μία λίστα αντικειμένων `SortRowInfo`, που το καθένα περιέχει πληροφορίες για την κατασκευή ενός επιπέδου ταξινόμησης, Η default τιμή είναι μία κενή λίστα.

### Dependency Property Information:

**Identifier field:** `SortListInputProperty`

### Παρατηρήσεις:

Αν επιθυμούμε να κατασκευαστούν αυτόματα τα επίπεδα ταξινόμησης στο `SortControl`, ώστε ο χρήστης να τα βρει έτοιμα μπροστά του, θα χρησιμοποιήσουμε αυτήν την ιδιότητα. Θέτοντας σε αυτήν ένα αντικείμενο `List(Of SortRowInfo)`, σε run-time mode, τα επίπεδα κατασκευάζονται αυτόματα.

Θα πρέπει να είμαστε προσεκτικοί στον καθορισμό των πεδίων στις ιδιότητες `FieldsNames` και `FieldsNamesOriginal`, ώστε να συμπίπτουν στην ονοματοδοσία όπως αυτή έχει οριστεί πρωτύτερα στον XAML κώδικα κατά τον ορισμό του `SortControl`.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε **τρία έτοιμα επίπεδα ταξινόμησης, με πεδία ταξινόμησης ΒΑΘΜΟΣ, ΕΠΩΝΥΜΟ, ΟΝΟΜΑ**, πραγματικά ονόματα τα `GRADE`, `LASTNAME`, `FIRSTNAME` και σχετικές τιμές διάταξης: Να σημειώσουμε ότι, οι τιμές διάταξης προέρχονται από την αντίστοιχη απαρίθμηση `OrderEnum` του `SortControl`.

#### XAML:

```
<zeus:SortControl Name="sortControl" Width="Auto" Margin="10"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    ... />
```

**VB:**

```
sortControl.SortListInput = New List(Of SortRowInfo) From
{
    New SortRowInfo With {.FieldName = "ΒΑΘΜΟΣ", _
                        .FieldNameOriginal = "GRADE", _
                        .OrderValue = SortControl.OrderEnum.Descending},

    New SortRowInfo With {.FieldName = "ΕΠΩΝΥΜΟ", _
                        .FieldNameOriginal = "LASTNAME", _
                        .OrderValue = SortControl.OrderEnum.Ascending},

    New SortRowInfo With {.FieldName = "ΟΝΟΜΑ", _
                        .FieldNameOriginal = "FIRSTNAME", _
                        .OrderValue = SortControl.OrderEnum.Ascending}
}
```

Η παρακάτω εικόνα δείχνει το αποτέλεσμα.

The screenshot shows a user interface for sorting data. At the top, there are two buttons: a green button with a plus sign labeled "Προσθήκη επιπέδου" (Add level) and a red button with a minus sign labeled "Διαγραφή επιπέδου" (Delete level). Below these is a table with two columns: "Πεδίο" (Field) and "Διάταξη" (Sort). The table has three rows, each with a label on the left and a dropdown menu in each column. The first row is labeled "Ταξινόμηση κατά:" (Sort by:) and has "ΒΑΘΜΟΣ" in the field column and "Φθίνουσα" (Descending) in the sort column. The second row is labeled "Επειτα κατά:" (Then by:) and has "ΕΠΩΝΥΜΟ" in the field column and "Αύξουσα" (Ascending) in the sort column. The third row is labeled "Επειτα κατά:" (Then by:) and has "ΟΝΟΜΑ" in the field column and "Αύξουσα" (Ascending) in the sort column. The third row is highlighted in blue. At the bottom of the interface are two buttons: a green "OK" button and a red "Ακύρωση" (Cancel) button.

	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΒΑΘΜΟΣ	Φθίνουσα
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

## SortListOutput

Επιστρέφει μία λίστα αντικειμένων **SortRowInfo**, καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) ταξινόμησης. Είναι τύπου **List(of SortRowInfo)**.

### Σύνταξη:

#### VB:

```
Public ReadOnly Property SortListOutput As List(Of SortRowInfo)
```

#### Τύπος: **System.Collections.Generic.List(Of T)**

Επιστρέφει μία λίστα αντικειμένων **SortRowInfo**, που το καθένα περιέχει πληροφορίες για κάθε επίπεδο ταξινόμησης που έχει διαμορφώσει ο χρήστης, Η default τιμή είναι μία κενή λίστα.

### Dependency Property Information:

**Identifier field:** SortListOutputProperty

### Παρατηρήσεις:

Οι πληροφορίες του κάθε **SortRowInfo** αντικειμένου αφορούν: το display όνομα του πεδίου ταξινόμησης (**FieldName**), το πραγματικό όνομα του πεδίου ταξινόμησης, όπως είναι στη βάση (**FieldNameOriginal**) και η τιμή διάταξης (**OrderValue**, τύπου **OrderEnum**). Αυτή η λίστα διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί στο πρόγραμμα η public μέθοδος **CreateSorting**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, στον **SortControl OKClicked** event handler, διαβάζουμε ένα-ένα τα αντικείμενα **SortRowInfo** της ιδιότητας **SortListOutput** και τυπώνουμε συνολικά τα display names των πεδίων ταξινόμησης και τις τιμές διάταξης (όπως επιστρέφονται default με τη μέθοδο **ToString**).

#### VB:

```
Private Sub SortControl_OKClicked(sender As Object, e As RoutedEventArgs)

    Dim sList As String = ""

    For Each s As SortRowInfo In sortControl.SortListOutput

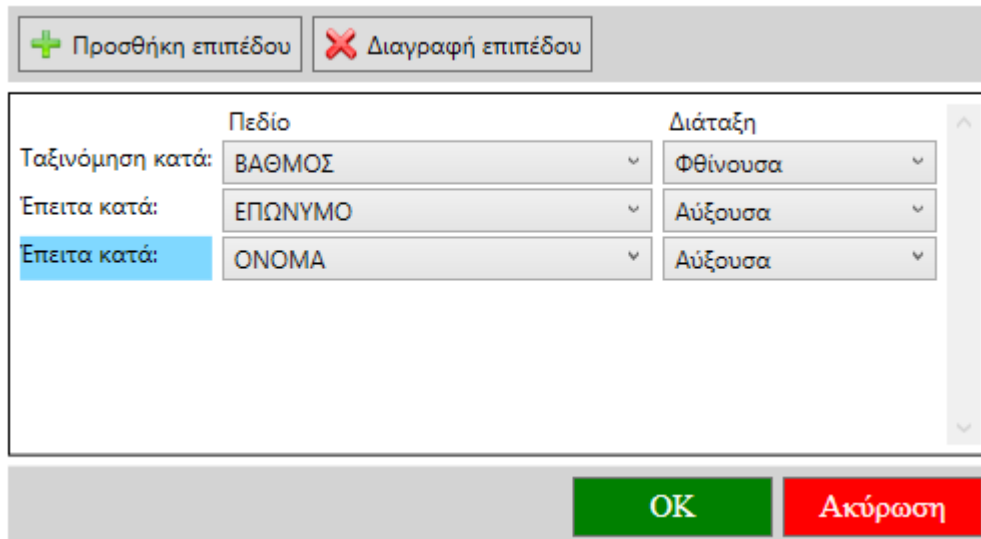
        sList += s.ToString & vbCrLf
    
```

Next

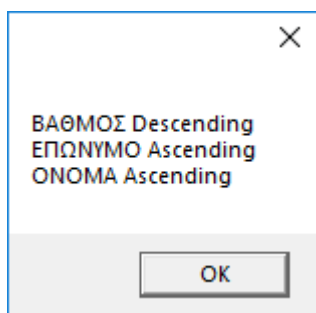
```
MessageBox.Show(sList)
```

End Sub

Η παρακάτω εικόνα δείχνει πώς έχει διαμορφωθεί το SortControl με βάση τις επιλογές του χρήστη...



... και όταν πατήσει το button OK εμφανίζεται η λίστα των πεδίων με τις τιμές διάταξης:



Φυσικά, σε μία πραγματική εφαρμογή, η ιδιότητα `SortListOutput` θα μπορούσε να αξιοποιηθεί διαφορετικά.

## SortStringOutput

Επιστρέφει ένα string με τα πεδία ταξινόμησης (πραγματικά ονόματα, όπως είναι στη Βάση) και την τιμή διάταξής τους, διαχωρισμένα με κόμμα.

### Σύνταξη:

#### VB:

```
Public ReadOnly Property SortStringOutput As String
```

**Τύπος:** System.String

Επιστρέφει ένα String, που περιέχει τα πεδία ταξινόμησης (τα πραγματικά ονόματα, όπως δηλώνονται στην ιδιότητα FieldsNamesOriginal) με τις τιμές διάταξης, με βάση τις επιλογές του χρήστη. Η default τιμή είναι String.Empty.

### Dependency Property Information:

**Identifier field:** SortStringOutputProperty

### Παρατηρήσεις:

Ένα παράδειγμα είναι "LASTNAME ASC, FIRSTNAME ASC". Αυτό το string μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Order by μίας SQL select query. Η SortStringOutput διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί στο πρόγραμμα η public μέθοδος CreateSorting.

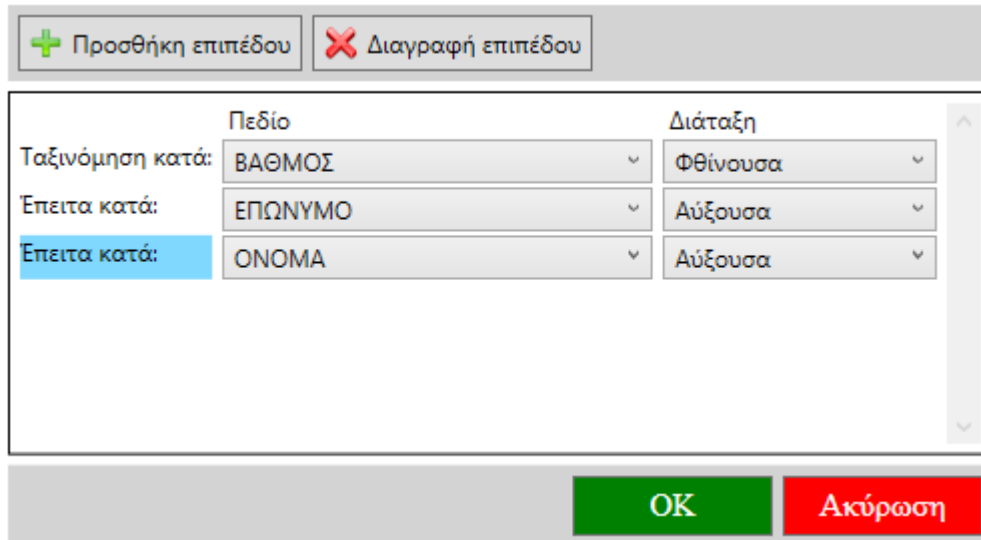
### Παράδειγμα:

Στο επόμενο παράδειγμα, στον SortControl OKClicked event handler, τυπώνουμε την τιμή της ιδιότητας SortStringOutput.

#### VB:

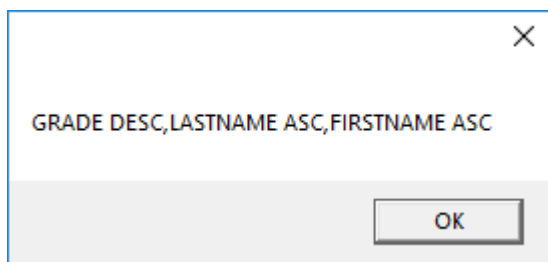
```
Private Sub SortControl_OKClicked(sender As Object, e As RoutedEventArgs)  
    MessageBox.Show(sortControl.SortStringOutput)  
End Sub
```

Η παρακάτω εικόνα δείχνει πώς έχει διαμορφωθεί το SortControl με βάση τις επιλογές του χρήστη...



	Πεδίο	Διάταξη
Ταξινόμηση κατά:	ΒΑΘΜΟΣ	Φθίνουσα
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα

... και όταν πατήσει το button OK εμφανίζεται το περιεχόμενο της SortStringOutput.



GRADE DESC, LASTNAME ASC, FIRSTNAME ASC

OK

## Συμβάντα

Όνομα	Περιγραφή
OKClicked	Ενεργοποιείται όταν ο χρήστης πατήσει το button "OK".
CancelClicked	Ενεργοποιείται όταν ο χρήστης πατήσει το button "Ακύρωση".



## OKClicked

Ενεργοποιείται όταν ο χρήστης πατήσει το button "OK".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event OKClicked As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:SortControl OKClicked="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε τοποθετήσει αυτόνομα το SortControl σε ένα παράθυρο (Window). Όταν ο χρήστης πατήσει το button "OK", το περιεχόμενο της ιδιότητας SortStringOutput τοποθετείται σε μία public ιδιότητα ReturnSortString και το παράθυρο κλείνει.

Προφανώς, η ιδιότητα ReturnSortString θα αξιοποιηθεί σε μία SQL select query για να φέρει κάποια δεδομένα από τη Βάση με την επιλεγμένη ταξινόμηση.

#### XAML:

```
<zeus:SortControl Name="sortControl" ...  
    OKClicked="SortControl_OKClicked"  
    CancelClicked="SortControl_CancelClicked"/>
```

#### VB:

```
Private Sub SortControl_OKClicked(sender As Object, e As RoutedEventArgs)  
  
    ReturnSortString = sortControl.SortStringOutput  
    Me.Close()  
  
End Sub
```

## CancelClicked

Ενεργοποιείται όταν ο χρήστης πατήσει το button "Ακύρωση".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event CancelClicked As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:SortControl CancelClicked="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε τοποθετήσει αυτόνομα το SortControl σε ένα παράθυρο (Window). Όταν ο χρήστης πατήσει το button "Ακύρωση", το παράθυρο κλείνει.

#### XAML:

```
<zeus:SortControl Name="sortControl" ...  
    OKClicked="SortControl_OKClicked"  
    CancelClicked="SortControl_CancelClicked"/>
```

#### VB:

```
Private Sub SortControl_CancelClicked(sender As Object, e As RoutedEventArgs)  
    Me.Close()  
End Sub
```

## Μέθοδοι

Όνομα	Περιγραφή
CreateSorting()	<b>Δημιουργεί το περιεχόμενο των ιδιοτήτων SortStringOutput και SortListOutput.</b>
LoadFromFile(filePath As String)	<b>Φορτώνει το περιεχόμενο του SortControl (τα επίπεδα ταξινόμησης) από xml αρχείο.</b>
SaveToFile(filePath As String)	<b>Αποθηκεύει το περιεχόμενο του SortControl (τα επίπεδα ταξινόμησης) σε xml αρχείο.</b>

## CreateSorting

Δημιουργεί το περιεχόμενο των ιδιοτήτων `SortStringOutput` και `SortListOutput`.

Σύνταξη:

**VB :**

```
Public Sub CreateSorting()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η δημιουργία ήταν επιτυχής.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καλούμε την `CreateSorting` πριν αποθηκεύσουμε το `SortControl` στο αρχείο `Sorting.xml`:

**VB:**

```
If sortControl.CreateSorting Then
```

```
    sortControl.SaveToFile("Sorting.xml")
```

```
    MessageBox.Show("Η ταξινόμηση αποθηκεύτηκε με επιτυχία")
```

```
End If
```

## LoadFromFile

Φορτώνει το περιεχόμενο του SortControl (τα επίπεδα ταξινόμησης) από xml αρχείο.

Σύνταξη:

**VB :**

```
Public Sub LoadFromFile(  
    filePath As String  
)
```

Παράμετροι:

*filePath*

Τύπος: String

Το όνομα του xml αρχείου που περιέχει το SortControl .

Επιστρεφόμενη τιμή:

Τύπος: Boolean.

True, αν η φόρτωση ήταν επιτυχής.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, φορτώνουμε τα επίπεδα ταξινόμησης, που είναι αποθηκευμένα στο αρχείο Sorting.xml, στο SortControl:



**VB:**

```
sortControl.LoadFromFile("Sorting.xml")
```

Αν το περιεχόμενο του αρχείου **Sorting.xml** είναι το παρακάτω:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>  
<SortControl>  
  <SortLevels>  
    <SortLevel FileName="ΒΑΘΜΟΣ" FileNameOriginal="GRADE"  
              Order="Descending" />  
    <SortLevel FileName="ΕΠΩΝΥΜΟ" FileNameOriginal="LASTNAME"  
              Order="Ascending" />  
    <SortLevel FileName="ΟΝΟΜΑ" FileNameOriginal="FIRSTNAME"  
              Order="Ascending" />  
  </SortLevels>  
</SortControl>
```

τότε η **φόρτωσή του στο SortControl** θα το διαμορφώσει όπως φαίνεται στην παρακάτω εικόνα:

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
	Πεδίο	Διάταξη	
Ταξινόμηση κατά:	ΒΑΘΜΟΣ	Φθίνουσα	
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα	
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα	

OK Ακύρωση

## SaveToFile

Αποθηκεύει το περιεχόμενο του SortControl (τα επίπεδα ταξινόμησης) σε xml αρχείο.

### Σύνταξη:

#### VB :

```
Public Sub SaveToFile(  
    filePath As String  
)
```

Παράμετροι:

*filePath*

Τύπος: String

Το όνομα του xml αρχείου στο οποίο θα γίνει η αποθήκευση .

*Επιστρεφόμενη τιμή:*

Τύπος: Boolean.

True, αν η αποθήκευση ήταν επιτυχής.



### Παράδειγμα:

Στο επόμενο παράδειγμα, αποθηκεύουμε τα επίπεδα ταξινόμησης του SortControl, στο αρχείο Sorting.xml:

#### VB:

```
sortControl.SaveToFile("Sorting.xml")
```

Αν το **SortControl** έχει διαμορφωθεί όπως στην παρακάτω εικόνα:

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Ταξινόμηση κατά:	Πεδίο	Διάταξη	
	ΒΑΘΜΟΣ	Φθίνουσα	
Επειτα κατά:	ΕΠΩΝΥΜΟ	Αύξουσα	
Επειτα κατά:	ΟΝΟΜΑ	Αύξουσα	
		OK	Ακύρωση

τότε η **αποθήκευση στο αρχείο Sorting.xml** θα δημιουργήσει το xml περιεχόμενο των επιπέδων ταξινόμησης:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<SortControl>
  <SortLevels>
    <SortLevel FileName="ΒΑΘΜΟΣ" FileNameOriginal="GRADE"
              Order="Descending" />
    <SortLevel FileName="ΕΠΩΝΥΜΟ" FileNameOriginal="LASTNAME"
              Order="Ascending" />
    <SortLevel FileName="ΟΝΟΜΑ" FileNameOriginal="FIRSTNAME"
              Order="Ascending" />
  </SortLevels>
</SortControl>
```



## Styles και Templates

- Parts και States
- To default Style και ControlTemplate

## Parts και States

Το default ControlTemplate περιλαμβάνει κάποια **part names** και **visual states**. Μπορείτε να τροποποιήσετε το default ControlTemplate ώστε να δώσετε στο control μία μοναδική εμφάνιση.

Ο παρακάτω πίνακας εμφανίζει τα **part names** του **SortControl**:

Part	Τύπος	Περιγραφή
PART_AddLevel	Button	Το button που προσθέτει ένα επίπεδο ταξινόμησης ("Προσθήκη επιπέδου").
PART_RemoveLevel	Button	Το button που αφαιρεί ένα επίπεδο ταξινόμησης ("Διαγραφή επιπέδου").
PART_OK	Button	Το button επιβεβαίωσης ("OK").
PART_Cancel	Button	Το button ακύρωσης ("Ακύρωση").
PART_SortRowsContainer	Grid	Το container των επιπέδων (γραμμών) ταξινόμησης.

Ο παρακάτω πίνακας εμφανίζει τα **visual states** του **SortControl**:

VisualState	VisualStateGroup	Περιγραφή
Normal	CommonStates	Το default state.
MouseOver	CommonStates	Το state όταν ο δείκτης του ποντικιού είναι πάνω από το control.
Disabled	CommonStates	Το state όταν το control είναι disabled.
Focused	FocusStates	Το state όταν το control έχει το focus.
Unfocused	FocusStates	Το state όταν το control δεν έχει το focus.

Το default ControlTemplate έχει καθορισμένο μόνο το Disabled state. Μπορείτε να δημιουργήσετε ένα custom ControlTemplate για να το αλλάξετε ή/και για να καθορίσετε τα υπόλοιπα.

## To default Style και ControlTemplate

Ο XAML κώδικας για το **default Style** και **ControlTemplate** φαίνεται παρακάτω. Μπορείτε να βασιστείτε σε αυτόν για να δημιουργήσετε μία μικρή ή μεγάλη παραλλαγή του δικού σας custom Style και ControlTemplate:

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.DataControls">

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

    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="White"/>
    <Setter Property="FontWeight" Value="Normal"/>

</Style>

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

    <Setter Property="Background" Value="#80D8FF"/>
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="FontWeight" Value="Normal"/>

</Style>

<Style TargetType="{x:Type z:SortControl}">

    <Setter Property="RowHeaderTextStyle"
        Value="{StaticResource SortHeaderRowDefaultStyle }" />
    <Setter Property="SelectedRowHeaderTextStyle"
        Value="{StaticResource SelectedSortHeaderRowDefaultStyle }" />

    <Setter Property="Template">
        <Setter.Value>

            <ControlTemplate TargetType="{x:Type z:SortControl}">

                <!-- Root element -->
                <Border Background="{TemplateBinding Background}"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}" >

                    <!-- Visual States -->
                    <VisualStateManager.VisualStateGroups >

                        <VisualStateGroup Name="CommonStates">

                            <VisualStateGroup.Transitions >
                                <VisualTransition GeneratedDuration="0:0:0.2">
                                </VisualTransition>
                            </VisualStateGroup.Transitions>

                            <VisualState Name="Normal"/>
                            <VisualState Name="MouseOver" />
                            <VisualState Name="Disabled">
                                <Storyboard >
                                    <ColorAnimationUsingKeyFrames
                                        Storyboard.TargetName="borderButtonsLevelContainerBackground"
                                        Storyboard.TargetProperty="Background" />
                                </Storyboard>
                            </VisualState>
                        </VisualStateGroup>
                    </VisualStateManager.VisualStateGroups >
                </Border>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

```

        Storyboard.TargetProperty ="Background.Color" >
            <DiscreteColorKeyFrame KeyTime="0"
Value="{StaticResource {x:Static SystemColors.InactiveBorderColorKey }}" />
            </ColorAnimationUsingKeyFrames>
            <ColorAnimationUsingKeyFrames
Storyboard.TargetName="borderButtonsOKCancelContainerBackground"
Storyboard.TargetProperty ="Background.Color" >
            <DiscreteColorKeyFrame KeyTime="0"
Value="{StaticResource {x:Static SystemColors.InactiveBorderColorKey }}" />
            </ColorAnimationUsingKeyFrames>
        </Storyboard>
    </VisualState>
</VisualStateGroup>

<VisualStateGroup Name="FocusStates">

    <VisualStateGroup.Transitions >
        <VisualTransition GeneratedDuration="0:0:0.2">
        </VisualTransition>
    </VisualStateGroup.Transitions>

    <VisualState Name="Focused" />
    <VisualState Name="Unfocused"/>

</VisualStateGroup>

</VisualStateManager.VisualStateGroups>

<!-- Content -->
<DockPanel IsEnabled="{TemplateBinding IsEnabled}"
Background="Transparent"
Margin="{TemplateBinding Padding}"
LastChildFill="True" >

    <!-- StackPanel container για τα AddLevel/Remove level buttons-->
    <Border Name="borderButtonsLevelContainerBackground"
DockPanel.Dock="Top"
Background="{TemplateBinding ButtonsLevelContainerBackground}" >

        <StackPanel Name="stkLevelButtons"
Orientation="Horizontal" Margin="5" >

            <Button Name="PART_AddLevel" Padding="5"
Style="{TemplateBinding ButtonAddLevelStyle}" >
                <Button.Content>
                    <StackPanel Orientation="Horizontal" >
                        <Image
Source="/ZeusDataControls;component/Images/Add_32x32.png"
Width="16" Height="16" />
                        <TextBlock
Text="{TemplateBinding ButtonAddLevelText}"
VerticalAlignment="Center"
Margin="5,0,0,0"/>
                    </StackPanel>
                </Button.Content>
            </Button>

            <Button Name="PART_RemoveLevel" Margin="5,0,0,0"
Padding="5"
Style="{TemplateBinding ButtonRemoveLevelStyle}">
                <Button.Content>

```

```

        <StackPanel Orientation="Horizontal" >
            <Image
Source="/ZeusDataControls;component/Images/Delete_32x32.png"
Width="16" Height="16" />
            <TextBlock
Text="{TemplateBinding ButtonRemoveLevelText}"
VerticalAlignment="Center"
Margin="5,0,0,0"/>
        </StackPanel>
    </Button.Content>
</Button>

</StackPanel>
</Border>

<!-- StackPanel container για τα OK/Cancel buttons-->
<Border Name="borderButtonsOKCancelContainerBackground"
DockPanel.Dock="Bottom"
Background="{TemplateBinding ButtonsOKCancelContainerBackground }" >

    <StackPanel Name="stkConfirmationButtons"
Orientation="Horizontal" Margin="5"
HorizontalAlignment="Right" >

        <Button Name="PART_OK"
Content="{TemplateBinding ButtonOKText}"
Padding="5"
Width="{TemplateBinding ButtonsOKCancelWidth }"
Style="{TemplateBinding ButtonOKStyle}"/>

        <Button Name="PART_Cancel"
Content="{TemplateBinding ButtonCancelText}"
Padding="5" Margin="5,0,0,0"
Width="{TemplateBinding ButtonsOKCancelWidth }"
Style="{TemplateBinding ButtonCancelStyle}"/>

    </StackPanel>

</Border>

<Border BorderBrush="Black" BorderThickness="1"
Margin="0,5,0,5">

    <ScrollViewer Margin="5" >

        <Grid Name="PART_SortRowsContainer"
Background="{TemplateBinding GridSortRowsContainerBackground }">

            <Grid.RowDefinitions >
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>

            <Grid.ColumnDefinitions >

                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="1.5*"/>
                <ColumnDefinition Width="*/>

            </Grid.ColumnDefinitions>

```

```
<!--Headers-->
<TextBlock
  Text="{TemplateBinding FieldsColumnHeaderText}"
  Grid.Column="1" Margin="8,0,0,2"/>
<TextBlock
  Text="{TemplateBinding OrderColumnHeaderText}"
  Grid.Column="2" Margin="8,0,0,2"/>

<!-- 1st Sort row, ready for user selection-->
<TextBlock
  Text="{TemplateBinding FirstRowHeaderText}"
  Grid.Row = "1" Grid.Column="0" />

<ComboBox  Grid.Row="1" Grid.Column="1"
  Margin="5,0,0,0"
Style="{TemplateBinding ComboBoxFieldsNamesStyle }"/>

<ComboBox  Grid.Row="1" Grid.Column="2"
  Margin="5,0,5,0"
  Style="{TemplateBinding ComboBoxOrderStyle }"/>

</Grid>

</ScrollViewer>

</Border>

</DockPanel>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>
```

## Κλάσεις

Το SortControl χρησιμοποιεί τις παρακάτω **κλάσεις** για τη λειτουργία του:

Όνομα	Περιγραφή
<b>SortRowInfo</b>	Περιλαμβάνει <b>πληροφορίες για ένα επίπεδο (γραμμή) ταξινόμησης.</b>

## SortRowInfo

Περιλαμβάνει πληροφορίες για ένα επίπεδο (γραμμή) ταξινόμησης.

### Σύνταξη:

#### VB:

```
Public Class SortRowInfo
```

#### Μέλη:

Ιδιότητα	Τύπος	Περιγραφή
FieldName	String	Το display όνομα του πεδίου ταξινόμησης.
FieldNameOriginal	String	Το πραγματικό όνομα του πεδίου ταξινόμησης, όπως έχει οριστεί στην Βάση.
OrderValue	SortControl.OrderEnum	Η τιμή διάταξης.

- Να σημειώσουμε ότι, οι πληροφορίες αυτές χρησιμοποιούνται για την αποθήκευση/φόρτωση ενός *SortControl* σε/από ένα xml αρχείο.





## CriteriaControl

Ένα **Control** που παρέχει μία φόρμα για τη διαμόρφωση, από τον χρήστη, ενός ή περισσότερων επιπέδων κριτηρίων για αναζήτηση/φιλτράρισμα.

### Σύνταξη:

#### VB:

```
<TemplateVisualState(Name:="Normal", GroupName:="CommonStates"),
TemplateVisualState(Name:="MouseOver", GroupName:="CommonStates"),
TemplateVisualState(Name:="Disabled", GroupName:="CommonStates"),
TemplateVisualState(Name:="Focused", GroupName:="FocusStates"),
TemplateVisualState(Name:="Unfocused", GroupName:="FocusStates"),
TemplatePart(Name:="PART_AddLevel", Type:=GetType(Button)),
TemplatePart(Name:="PART_RemoveLevel", Type:=GetType(Button)),
TemplatePart(Name:="PART_OK", Type:=GetType(Button)),
TemplatePart(Name:="PART_Cancel", Type:=GetType(Button)),
TemplatePart(Name:="PART_CriteriaRowsContainer", Type:=GetType(Grid))>
Public Class CriteriaControl
    Inherits Control
```

#### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:CriteriaControl ... />
```

## Παρατηρήσεις:

Πρόκειται για ένα πολύ ευέλικτο control που παρέχει στον χρήστη μία φόρμα για να διαμορφώσει ένα ή περισσότερα επίπεδα κριτηρίων αναζήτησης/φιλτραρίσματος. Η φόρμα αυτή, μοιάζει με αυτή του SortControl. Κάθε επίπεδο κριτηρίων είναι μία γραμμή με στοιχεία Πεδίο-Σύγκριση-Τιμή. Για περισσότερα του ενός κριτήρια, προστίθεται και σύνδεσμος και/ή (and/or). Για παράδειγμα, Επώνυμο-αρχίζει από-ΠΑΠ ή Βαθμός-μεγαλύτερο από-15,5. Επιπλέον, για να τον διευκολύνουμε να μην ξαναδιαμορφώνει εκ νέου τα επίπεδα κριτηρίων, το control έχει μεθόδους για αποθήκευση και ανάκτηση (απεριόριστες διαμορφώσεις).

Σχεδόν όλα τα γραφικά στοιχεία του control καθώς και τα λεκτικά μπορούν να παραμετροποιηθούν μέσω των αντίστοιχων ιδιοτήτων. Για παράδειγμα, τα λεκτικά των buttons Προσθήκη επιπέδου και Διαγραφή επιπέδου να αλλάξουν σε Add Level και Remove Level. επίσης, μπορούμε να αλλάξουμε και τα styles των buttons αυτών.

## Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε μέσα σε ένα Grid ενός παραθύρου διάφορα controls μεταξύ των οποίων ένα CriteriaControl και δύο oval buttons για αποθήκευση και φόρτωση της διαμόρφωσης καθώς και έλεγχος κριτηρίων. Για τα oval buttons έχουμε κατασκευάσει ένα ειδικό OvalButtonStyle, όπου όλη η δουλειά έχει γίνει μέσα στο ControlTemplate. Παρόμοια, στο τμήμα Window.Resources, θα δούμε κι άλλα custom styles που θα εφαρμοστούν σε αντίστοιχες ιδιότητες του CriteriaControl, ώστε να το παραμετροποιήσουμε σε εμφάνιση.

Μπορούμε να δούμε μερικές από τις ιδιότητες που έχουμε θέσει στο CriteriaControl, βασικότερες των οποίων είναι η FieldsNames (τα ονόματα των πεδίων προς επιλογή, για εμφάνιση), FieldsNamesOriginal (τα ονόματα των πεδίων όπως είναι στη Βάση) και FieldTypes (οι τύποι των πεδίων).

```
<Window x:Class="WinCriteria"
  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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded" >

<Window.Resources >

  <!-- Ένα απλό custom Oval Button. -->
  <Style x:Key="OvalButtonsStyle" TargetType="Button">

    <Setter Property="Background" Value="LightGray" />
```

```
<Setter Property="Template" >
  <Setter.Value>

    <ControlTemplate TargetType="{x:Type Button}">

      <Border Name="border" BorderBrush="Black" BorderThickness="1"
        CornerRadius="8"
        Background="{TemplateBinding Background}" >

        <!-- Visual States -->
        <VisualStateManager.VisualStateGroups >

          <VisualStateGroup Name="CommonStates">

            <VisualStateGroup.Transitions >
              <VisualTransition GeneratedDuration="0:0:0.2">
                </VisualTransition>
            </VisualStateGroup.Transitions>

            <VisualState Name="Normal"/>
            <VisualState Name="MouseOver" />
            <VisualState Name="Disabled">
              <Storyboard >
                <ColorAnimationUsingKeyFrames
                  Storyboard.TargetName="border"
                  Storyboard.TargetProperty="Background.Color" >
                  <DiscreteColorKeyFrame KeyTime="0"
                    Value="{StaticResource {x:Static SystemColors.InactiveBorderColorKey }}" />
                </ColorAnimationUsingKeyFrames>
              </Storyboard>
            </VisualState>

          </VisualStateGroup>

          <VisualStateGroup Name="FocusStates">

            <VisualStateGroup.Transitions >
              <VisualTransition GeneratedDuration="0:0:0.2">
                </VisualTransition>
            </VisualStateGroup.Transitions>

            <VisualState Name="Focused" />
            <VisualState Name="Unfocused"/>

          </VisualStateGroup>

        </VisualStateManager.VisualStateGroups>

        <ContentPresenter Name="presenter"
          Margin="{TemplateBinding Padding}" />

      </Border>

    </ControlTemplate>

  </Setter.Value>

</Setter>

<Style.Triggers >

  <Trigger Property="IsMouseOver" Value="True">
```

```
        <Setter Property="Background" Value="Yellow"/>
        <Setter Property="FontWeight" Value="Bold"/>
    </Trigger>
    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Background" Value="Cyan"/>
        <Setter Property="FontWeight" Value="Bold"/>
    </Trigger>

</Style.Triggers>

</Style>

<!-- Ένα style για το ComboBoxItem στο FieldNames ComboBox-->
<Style x:Key="FieldsNamesComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

    <Style.Triggers >
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Green"/>
        </Trigger>
    </Style.Triggers>
</Style>

<Style x:Key="FieldsNamesComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource FieldsNamesComboBoxItemStyle }"/>

</Style>

<Style x:Key="calendarStyle" TargetType="Calendar" >
    <Setter Property="Background" Value="LightBlue"/>
    <Setter Property="BorderBrush" Value="DarkBlue"/>
    <Setter Property="BorderThickness" Value="5"/>
</Style>

<!-- Ένα style για το OK button-->
<Style x:Key="CriteriaOKButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Green"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Verdana"/>

</Style>

<!-- Ένα style για το Cancel button-->
<Style x:Key="CriteriaCancelButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Verdana"/>

</Style>

</Window.Resources>

<Grid >

    <Grid.Resources >
```

```

<!-- Το style για τις λανθασμένες αριθμητικές τιμές εισόδου. -->
<Style x:Key="valueTextBoxErrorStyle" TargetType="TextBox">
    <Setter Property="Foreground" Value="Red"/>
</Style>

</Grid.Resources>

<Grid.RowDefinitions >
    <RowDefinition />
    <RowDefinition Height="Auto"/>
</Grid.RowDefinitions>

<z:CriteriaControl Name="criteriaControl" Width="Auto" Margin="10"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    DatePickerCalendarStyle="{StaticResource calendarStyle}"
    ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
    ButtonOKStyle="{StaticResource CriteriaOKButtonStyle}"
    ButtonCancelStyle="{StaticResource CriteriaCancelButtonStyle}"
    OKClicked="CriteriaControl_OKClicked"
    CancelClicked="CriteriaControl_CancelClicked" />

<!-- Buttons για Αποθήκευση, Φόρτωση και Έλεγχο κριτηρίων. -->
<StackPanel Grid.Row="1" Margin="10" Orientation="Horizontal" >

    <Button Name="btnSaveCriteria" Content="Αποθήκευση κριτηρίων"
        Padding="5" Click="btnSaveCriteria_Click"
        ToolTip="Αποθηκεύει τα κριτήρια στο αρχείο criteria.xaml στον φάκελο της
            εφαρμογής."
        Style="{StaticResource OvalButtonsStyle}"/>

    <Button Name="btnLoadCriteria" Content="Φόρτωση κριτηρίων"
        Padding="5" Margin="7,0,0,0" Click="btnLoadCriteria_Click"
        ToolTip="Φορτώνει τα κριτήρια από το αρχείο criteria.xaml, που βρίσκεται
            στον φάκελο της εφαρμογής."
        Style="{StaticResource OvalButtonsStyle}"/>

    <Button Name="btnValidateCriteria" Content="Έλεγχος κριτηρίων"
        Padding="5" Margin="7,0,0,0" Click="btCheckCriteria_Click"
        ToolTip="Ελέγχει τις αριθμητικές τιμές στα TextBoxes, με βάση τον τύπο τους
            (π.χ. Integer, Long, Single κλπ). Αν βρεί σφάλμα, εφαρμόζει το
            ValueTextBoxErrorStyle."
        Style="{StaticResource OvalButtonsStyle}"/>

</StackPanel>

</Grid>

</Window>

```

Παρακάτω, ο χρήστης έχει κάνει κάποιες επιλογές στο **CriteriaControl**:

Κριτήρια αναζήτησης (Φιλτράρισμα)

+ Προσθήκη επιπέδου ✖ Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τιμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑ
και	ΤΑΞΗ	είναι ίσο με	A
και	ΒΑΘΜΟΣ	μεγαλύτερο ή ίσο του	15,5
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019

OK Ακύρωση

Αποθήκευση κριτηρίων Φόρτωση κριτηρίων Έλεγχος κριτηρίων

Ο ΒΑΘΜΟΣ είναι τύπου Single. Αν δώσει λανθασμένη τιμή στο TextBox θα σημανθεί ένα λάθος, διότι στην ιδιότητα **ValueTextBoxErrorStyle** έχουμε θέσει ένα error style.

Κριτήρια αναζήτησης (Φιλτράρισμα)

+ Προσθήκη επιπέδου ✖ Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τιμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑ
και	ΤΑΞΗ	είναι ίσο με	A
και	ΒΑΘΜΟΣ	μεγαλύτερο ή ίσο του	15,5φ
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019

OK Ακύρωση

Αποθήκευση κριτηρίων Φόρτωση κριτηρίων Έλεγχος κριτηρίων

Το σημαντικό εδώ, είναι ότι όταν ο χρήστης πατήσει το **OK** τότε το control θα επιστρέψει μέσω της ιδιότητας **CriteriaStringOutput** το σχετικό criteria string, που στην προκειμένη περίπτωση είναι **"LASTNAME LIKE ΠΑ% AND CLASS='A' AND GRADE<=15.5 AND [REGISTRATION DATE] >=#6/1/2019#** (σε Access. Για SQLite και SQL Server το τμήμα της ημερομηνίας διαμορφώνεται διαφορετικά). Έτσι, μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Where μίας SQL select query.

Για διευκόλυνση του προγραμματιστή, ώστε να κρατάει κάπου την διαμόρφωση αυτή για να την θέσει εκ νέου αργότερα στο πρόγραμμα, παρέχεται και η ιδιότητα **CriteriaListOutput** που είναι τύπου **List(of CriteriaRowInfo)**. Το αντικείμενο **CriteriaRowInfo**, κρατάει τις πληροφορίες για το επίπεδο κριτηρίου (μίας γραμμής δηλαδή), όπου μεταξύ άλλων είναι που το πεδίο εμφάνισης

(FieldName), το πεδίο όπως είναι στη Βάση (FieldNameOriginal) και τον τύπο του πεδίου (FieldType).

## Απαριθμήσεις

Όνομα	Περιγραφή
<b>BooleanValuesEnum</b>	Περιλαμβάνει τις τιμές <b>Boolean</b> .
<b>ComparisonBooleanOperatorEnum</b>	Περιλαμβάνει τις τιμές Σύγκρισης για τύπο <b>Boolean</b> .
<b>ComparisonNumAndDateOperatorsEnum</b>	Περιλαμβάνει τις τιμές Σύγκρισης για γενικό τύπο <b>Numeric</b> και <b>DateTime</b> .
<b>ComparisonStringOperatorsEnum</b>	Περιλαμβάνει τις τιμές Σύγκρισης για τύπο <b>String</b> .
<b>ConnectorEnum</b>	Περιλαμβάνει τις τιμές Σύνδεσης των επιπέδων κριτηρίων.
<b>DatabaseFormatEnum</b>	Περιλαμβάνει τις τιμές για τον τύπο της Βάσης.
<b>FieldGeneralDataTypeEnum</b>	Περιλαμβάνει τις τιμές για τον γενικό τύπο ενός πεδίου.



## BooleanValuesEnum

Περιλαμβάνει τιμές **Boolean**.

**Σύνταξη:**

**VB:**

```
Public Enum BooleanValuesEnum
```

**Μέλη:**

Τιμή	Περιγραφή
TrueValue	Η τιμή για True.
FalseValue	Η τιμή για False.

## ComparisonBooleanOperatorEnum

Περιλαμβάνει τις τιμές Σύγκρισης για τύπο Boolean.

**Σύνταξη:**

**VB:**

```
Public Enum ComparisonBooleanOperatorEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Equal	Η τιμή για το "είναι ίσο με" (=)

## ComparisonNumAndDateOperatorsEnum

Περιλαμβάνει τις τιμές Σύγκρισης για γενικό τύπο Numeric και Date.

### Σύνταξη:

**VB:**

```
Public Enum ComparisonNumAndDateOperatorsEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Equal	Η τιμή για το "είναι ίσο με" (=)
GreaterThan	Η τιμή για το "μεγαλύτερο από" (>)
GreaterThanOrEqual	Η τιμή για το "μεγαλύτερο ή ίσο του" (>=)
LessThan	Η τιμή για το "μικρότερο από" (<)
LessThanOrEqual	Η τιμή για το "μικρότερο ή ίσο του" (<=)
NotEqual	Η τιμή για το "δεν είναι ίσο με" (<>)

## ComparisonStringOperatorsEnum

Περιλαμβάνει τις τιμές Σύγκρισης για τύπο **String**.

**Σύνταξη:**

**VB:**

```
Public Enum ComparisonStringOperatorsEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Equal	Η τιμή για το "είναι ίσο με"
BeginsWith	Η τιμή για το "αρχίζει με"
EndsWith	Η τιμή για το "τελειώνει σε"
Contains	Η τιμή για το "περιέχει"
NotEqual	Η τιμή για το "δεν είναι ίσο με"

## ConnectorEnum

Περιλαμβάνει τις τιμές Σύνδεσης των επιπέδων κριτηρίων.

**Σύνταξη:**

**VB:**

```
Public Enum ConnectorEnum
```

**Μέλη:**

Τιμή	Περιγραφή
AndConnector	Η τιμή για το "And"
OrConnector	Η τιμή για "Or".

## DatabaseFormatEnum

Περιλαμβάνει τις τιμές για τον τύπο της Βάσης.

**Σύνταξη:**

**VB:**

```
Public Enum DatabaseFormatEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Access2007	Η τιμή για Βάση Δεδομένων Access 2007 και άνω.
Sqlite3	Η τιμή για Βάση Δεδομένων SQLite 3.
SqlServer	Η τιμή για Βάση Δεδομένων SQL Server.

## FieldGeneralDataTypeEnum

Περιλαμβάνει τις τιμές για τον γενικό τύπο ενός πεδίου.

### Σύνταξη:

**VB:**

```
Public Enum FieldGeneralDataTypeEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Undefined	Η τιμή για το "μη καθορισμένο".
Text	Η τιμή για το τύπο κειμένου.
Numeric	Η τιμή για αριθμητικό τύπο.
DateTime	Η τιμή για ημερομηνιακό τύπο.
Bool	Η τιμή για λογικό τύπο.

## Ιδιότητες

Όνομα	Περιγραφή
BooleanValueTrueText	Καθορίζει το κείμενο (λεκτικό) για την τιμή "True", όπως θα εμφανίζεται στο ComboBox των τιμών Boolean. Είναι τύπου String.
BooleanValueFalseText	Καθορίζει το κείμενο (λεκτικό) για την τιμή "False", όπως θα εμφανίζεται στο ComboBox των τιμών Boolean. Είναι τύπου String.
ButtonAddLevelText	Καθορίζει το κείμενο (λεκτικό) για το button "Προσθήκη επιπέδου", τύπου String.
ButtonAddLevelStyle	Καθορίζει το στυλ για το button "Προσθήκη επιπέδου", τύπου Style.
ButtonCancelText	Καθορίζει το κείμενο (λεκτικό) για το button "Ακύρωση", τύπου String.
ButtonCancelStyle	Καθορίζει το στυλ για το button "Ακύρωση", τύπου Style.
ButtonOKText	Καθορίζει το κείμενο (λεκτικό) για το button "OK", τύπου String.
ButtonOKStyle	Καθορίζει το στυλ για το button "OK", τύπου Style.
ButtonRemoveLevelText	Καθορίζει το κείμενο (λεκτικό) για το button "Διαγραφή επιπέδου", τύπου String.
ButtonRemoveLevelStyle	Καθορίζει το στυλ για το button "Διαγραφή επιπέδου", τύπου Style.
ButtonsLevelContainerBackground	Καθορίζει το χρώμα φόντου για το container των buttons "Προσθήκη επιπέδου" και "Διαγραφή επιπέδου", τύπου Brush.



ButtonsOKCancelContainerBackground	Καθορίζει το χρώμα φόντου για το container των buttons "OK" και "Ακύρωση", τύπου Brush.
ButtonsOKCancelWidth	Καθορίζει το πλάτος, σε pixels, των buttons "OK" και "Ακύρωση". Είναι τύπου Double.
ComboBoxBooleanValueStyle	Καθορίζει το στυλ για το ComboBox των τιμών Boolean, τύπου Style.
ComboBoxComparisonStyle	Καθορίζει το στυλ για το ComboBox των τελεστών Σύγκρισης, τύπου Style.
ComboBoxConnectorStyle	Καθορίζει το στυλ για το ComboBox των τιμών Σύνδεσης, τύπου Style.
ComboBoxConnectorWidth	Καθορίζει το πλάτος, σε pixels, του ComboBox των τιμών Σύνδεσης, τύπου Double.
ComboBoxFieldsNamesStyle	Καθορίζει το στυλ για το ComboBox των πεδίων ταξινόμησης, τύπου Style.
ComparisonColumnHeaderText	Καθορίζει το κείμενο (λεκτικό) για την Επικεφαλίδα της στήλης Σύγκρισης, τύπου String.
ComparisonOperatorBoolEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "είναι ίσο με", για Boolean τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorBeginsWithText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "αρχίζει με", για String τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorContainsText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "περιέχει", για String τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorEndsWithText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "τελειώνει σε", για String τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "είναι ίσο με", για String τύπο δεδομένων. Είναι τύπου String.

ComparisonOperatorNotEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "δεν είναι ίσο με", για String τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorNumEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "είναι ίσο με", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorNumGreaterThanOrEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μεγαλύτερο ή ίσο του", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorNumGreaterThanOrEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μεγαλύτερο από", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorNumLessThanOrEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μικρότερο ή ίσο του", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorNumLessThanOrEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μικρότερο από", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.
ComparisonOperatorNumNotEqualText	Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "δεν είναι ίσο με", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.
ConnectorANDText	Καθορίζει το κείμενο (λεκτικό) για τον συνδέτη "AND" των επιπέδων κριτηρίων. Είναι τύπου String.
ConnectorORText	Καθορίζει το κείμενο (λεκτικό) για τον συνδέτη "OR" των επιπέδων κριτηρίων. Είναι τύπου String.

CriteriaListInput	<p>Καθορίζει <b>μία λίστα αντικειμένων CriteriaRowInfo</b>, καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) κριτηρίων. Είναι τύπου <b>List(of CriteriaRowInfo)</b>.</p> <p>Θέτοντας αυτήν την ιδιότητα, διαμορφώνονται αυτόματα τα επίπεδα κριτηρίων.</p>
CriteriaListOutput	<p>Επιστρέφει <b>μία λίστα αντικειμένων CriteriaRowInfo</b>, καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) κριτηρίων. Είναι τύπου <b>List(of CriteriaRowInfo)</b>.</p> <p>Αυτή η λίστα διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί εσωτερικά η public μέθοδος CreateCriteria.</p>
CriteriaStringOutput	<p>Επιστρέφει <b>ένα string με τα πεδία κριτηρίων (πραγματικά ονόματα, όπως είναι στη Βάση) τον τελεστή σύγκρισης, την τιμή μαζί με την σύνδεση (για το μετά το πρώτο επίπεδο), διαχωρισμένα με κόμμα</b>. Για παράδειγμα, "LASTNAME LIKE ΠΑΠ% AND CLASS='A' AND GRADE&lt;=15.5" . Είναι τύπου <b>String</b>.</p> <p>Αυτή το output string διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί εσωτερικά η public μέθοδος CreateCriteria.</p> <p>Μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Where μίας SQL select query.</p>
DatabaseFormat	<p>Καθορίζει τον <b>τύπο της Βάσης Δεδομένων, ώστε να δημιουργήσει σωστά την τιμή της ιδιότητας CriteriaStringOutput</b>. Είναι τύπου <b>DatabaseFormatEnum</b>.</p>
DatePickerCalendarStyle	<p>Καθορίζει το <b>στυλ για το Calendar του DatePicker που χρησιμοποιείται για επιλογή ημερομηνιακής τιμής</b>. Είναι τύπου <b>Style</b>.</p>
DatePickerStyle	<p>Καθορίζει το <b>στυλ για το DatePicker που χρησιμοποιείται για επιλογή ημερομηνιακής τιμής</b>. Είναι τύπου <b>Style</b>.</p>
FieldsColumnHeaderText	<p>Καθορίζει το <b>κείμενο (λεκτικό) για τη στήλη των πεδίων κριτηρίων, τύπου String</b>.</p>

FieldsDataTypes	<p>Καθορίζει τα ονόματα των τύπων δεδομένων για τα πεδία των κριτηρίων. Είναι τύπου <b>String</b>.</p> <p><b>Πρέπει να διαχωρίζονται με κόμμα, χωρίς κενά.</b> Για παράδειγμα, "String,String,Integer,Date,Boolean". Ο αριθμός των προσδιοριζομένων τύπων πρέπει να είναι ίδιος με αυτόν της ιδιότητας FieldsNamesOriginal και να συμβαδίζει τακτικά (ίδια θέση) με τα ονόματα των πεδίων.</p>
FieldsNames	<p>Καθορίζει τα ονόματα των πεδίων κριτηρίων, όπως θα εμφανίζονται στο αντίστοιχο <b>ComboBox</b>. Είναι τύπου <b>String</b>.</p> <p><b>Πρέπει να διαχωρίζονται με κόμμα, χωρίς κενά.</b> Για παράδειγμα, "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ". Ο αριθμός των πεδίων πρέπει να είναι ίδιος με αυτόν της ιδιότητας FieldsNamesOriginal.</p>
FieldsNamesOriginal	<p>Καθορίζει τα πραγματικά ονόματα των πεδίων ταξινόμησης, όπως υπάρχουν στη Βάση. Είναι τύπου <b>String</b>.</p> <p><b>Πρέπει να διαχωρίζονται με κόμμα, χωρίς κενά.</b> Για παράδειγμα, "LASTNAME,FIRSTNAME,CLASS,DIVISION". Ο αριθμός των πεδίων πρέπει να είναι ίδιος με αυτόν της ιδιότητας FieldsNames. Με βάση αυτά τα πεδία θα σχηματιστεί το CriteriaStringOutput.</p>
GridCriteriaRowsContainerBackground	Καθορίζει το χρώμα φόντου για το container των επιπέδων κριτηρίων, τύπου <b>Brush</b> .
RowHeaderStyle	Καθορίζει το στυλ του Row Header που εμφανίζεται σε κάθε επίπεδο (γραμμή) κριτηρίων (εκτός του επιλεγμένου) στο αριστερό μέρος. Είναι τύπου <b>Style</b> .
RowHeaderWidth	Καθορίζει το πλάτος, σε pixels, του Row Header που εμφανίζεται σε κάθε επίπεδο (γραμμή) κριτηρίων. Είναι τύπου <b>Double</b> .

SelectedRowHeaderStyle	Καθορίζει το <b>στυλ του Row Header</b> που εμφανίζεται για το επιλεγμένο επίπεδο (γραμμή) κριτηρίων στο αριστερό μέρος (με το default γαλάζιο χρώμα). Είναι τύπου <b>Style</b> .
UseCommaAsDecimalSeparator	Καθορίζει αν θα χρησιμοποιείται το κόμμα ως διαχωριστή δεκαδικών τιμών. Είναι τύπου <b>Boolean</b> .
ValueColumnHeaderText	Καθορίζει το κείμενο (λεκτικό) για τη στήλη των <b>Τιμών</b> , τύπου <b>String</b> .
ValueTextBoxErrorStyle	Καθορίζει το <b>στυλ</b> για το <b>TextBox</b> σε περίπτωση που ο χρήστης εισάγει μία λανθασμένη αριθμητική τιμή. Είναι τύπου <b>Style</b> .
ValueTextBoxStyle	Καθορίζει το <b>στυλ</b> για το <b>TextBox</b> που χρησιμοποιείται για εισαγωγή από τον χρήστη μίας αριθμητικής τιμής ή κειμένου. Είναι τύπου <b>Style</b> .

## BooleanValueTrueText

Καθορίζει το κείμενο (λεκτικό) για την τιμή "True", όπως θα εμφανίζεται στο ComboBox των τιμών Boolean. Είναι τύπου String.

### Σύνταξη:

**VB:**

```
Public Property BooleanValueTrueText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για την τιμή "True" στο ComboBox των τιμών Boolean. Η default τιμή είναι "Ναι".

### Dependency Property Information:

**Identifier field:** BooleanValueTrueTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε την τιμή "Yes" στην ιδιότητα **BooleanValueTrueText** και "No" στην ιδιότητα **BooleanValueFalseText**.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    BooleanValueTrueText="Yes"
    BooleanValueFalseText="No" ... />
```

**VB:**

```
criteriaControl.BooleanValueTrueText = "Yes"
criteriaControl.BooleanValueFalseText = "No"
```

	Πεδίο	Σύγκριση	Τιμή
<input type="checkbox"/>	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
<input checked="" type="checkbox"/>	και		
<input type="checkbox"/>	ΕΝΕΡΓΟΣ;	είναι ίσο με	Yes
			Yes
			No

## BooleanValueFalseText

Καθορίζει το κείμενο (λεκτικό) για την τιμή "False", όπως θα εμφανίζεται στο `ComboBox` των τιμών `Boolean`. Είναι τύπου `String`.

### Σύνταξη:

#### VB:

```
Public Property BooleanValueFalse Text As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό για την τιμή "False" στο `ComboBox` των τιμών `Boolean`. Η default τιμή είναι "Όχι".

### Dependency Property Information:

**Identifier field:** `BooleanValueFalseTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε την τιμή "Yes" στην ιδιότητα `BooleanValueTrueText` και "No" στην ιδιότητα `BooleanValueFalseText`.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"  
    BooleanValueTrueText="Yes"  
    BooleanValueFalseText="No" ... />
```

#### VB:

```
criteriaControl.BooleanValueTrueText = "Yes"  
criteriaControl.BooleanValueFalseText = "No"
```



	Πεδίο	Σύγκριση	Τιμή
<input type="checkbox"/>	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
<input checked="" type="checkbox"/>	και		Yes
<input type="checkbox"/>	ΕΝΕΡΓΟΣ;	είναι ίσο με	Yes
			No

## ButtonAddLevelText

Καθορίζει το κείμενο (λεκτικό) για το **button** "Προσθήκη επιπέδου", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonAddLevelText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button** "Προσθήκη επιπέδου". Η default τιμή είναι "Προσθήκη επιπέδου".

### Dependency Property Information:

**Identifier field:** `ButtonAddLevelTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Add Level"** για το **button** "Προσθήκη επιπέδου" και **"Remove Level"** για το **button** "Διαγραφή επιπέδου" :

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ButtonAddLevelText="Add Level"
    ButtonRemoveLevelText="Remove Level"... />
```

#### VB:

```
criteriaControl.ButtonAddLevelText = "Add Level"
criteriaControl.ButtonRemoveLevelText = "Remove Level"
```

		Πεδίο	Σύγκριση	Τιμή
		ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
<input checked="" type="checkbox"/>	ή	ΕΠΩΝΥΜΟ	αρχίζει από	ΔΙΑ

## ButtonAddLevelStyle

Καθορίζει το **στυλ για το button "Προσθήκη επιπέδου"**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ButtonAddLevelStyle As Style
```

#### Τύπος: **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Προσθήκη επιπέδου". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonAddLevelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"CriteriaLevelButtonsStyle"** και το εφαρμόζουμε στα **buttons "Προσθήκη επιπέδου"** και **"Διαγραφή επιπέδου"** :

#### XAML:

```
<Window x:Class="WinCriteria"
  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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded" >

<Window.Resources >

  ...

  <!-- Ένα style για τα Level buttons-->
  <Style x:Key="CriteriaLevelButtonsStyle" TargetType="{x:Type Button}">

    <Setter Property="FontSize" Value="12"/>
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="Background" Value="LightBlue" />
    <Setter Property="FontFamily" Value="Times New Roman"/>

  </Style>
```

```

</Window.Resources>

<Grid >

...

<z:CriteriaControl Name="criteriaControl" Width="Auto" Margin="10"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
        ButtonAddLevelStyle="{StaticResource CriteriaLevelButtonsStyle}"
        ButtonRemoveLevelStyle="{StaticResource CriteriaLevelButtonsStyle}"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        ButtonOKStyle="{StaticResource CriteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource CriteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"
        CancelClicked="CriteriaControl_CancelClicked" />

</Grid>

</Window>

```

**VB:**

```

Dim criteriaLevelButtonStyle As New Style(GetType(Button))

With criteriaLevelButtonStyle

    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 12.0})
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _
        .Value = FontWeights.Bold})
    .Setters.Add(New Setter With {.Property = BackgroundProperty, _
        .Value = Brushes.LightBlue})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty, _
        .Value = New FontFamily("Times New Roman")})

End With

With criteriaControl

    .ButtonAddLevelStyle = criteriaLevelButtonStyle
    .ButtonRemoveLevelStyle = criteriaLevelButtonStyle

End With

```

+ Προσθήκη επιπέδου		✖ Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τιμή	
<input type="checkbox"/> και	ΕΠΩΝΥΜΟ	τελειώνει σε	ΟΥ
	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μικρότερο ή ίσο του	6/6/2019 <input type="button" value="15"/>

OK Ακύρωση

## ButtonCancelText

Καθορίζει το κείμενο (λεκτικό) για το **button "Ακύρωση"**, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonCancelText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button "Ακύρωση"**. Η default τιμή είναι **"Ακύρωση"**.

### Dependency Property Information:

**Identifier field:** `ButtonCancelTextProperty`

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το **button "Ακύρωση"**, το λεκτικό **"Done"** για το **button "OK"** και θέτουμε το **πλάτος των buttons** αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το **default style**, που προέρχεται από την κλάση **Button**.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ButtonOKText="Done"
    ButtonCancelText="Cancel"
    ButtonsOKCancelWidth="80"
    ... />
```

#### VB:

```
criteriaControl.ButtonOKText = "Done"
criteriaControl.ButtonCancelText = "Cancel"
criteriaControl.ButtonsOKCancelWidth = 80
```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
<input type="checkbox"/>	και <input type="checkbox"/> ΤΜΗΜΑ	είναι ίσο με	A1



## ButtonCancelStyle

Καθορίζει το **στυλ για το button "Ακύρωση"**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ButtonCancelStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Ακύρωση". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonCancelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"CriteriaOKButtonStyle"** και το εφαρμόζουμε στο **button "OK"** και ένα δεύτερο **resource style** με όνομα **"CriteriaCancelButtonStyle"** και το εφαρμόζουμε στο **button "Ακύρωση"**.

**XAML:**

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

  <!--Ένα style για το OK button-->
  <Style x:Key="CriteriaOKButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Green"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

  </Style>
```

```

<!-- Ένα style για το Cancel button-->
<Style x:Key="CriteriaCancelButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{DynamicResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim criteriaOKButtonStyle As New Style(GetType(Button))
Dim criteriaCancelButtonStyle As New Style(GetType(Button))

With criteriaOKButtonStyle

    .Setters.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.[Property] = ForegroundProperty, _
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.[Property] = BackgroundProperty, _
        .Value = Brushes.Green})
    .Setters.Add(New Setter With {.[Property] = FontFamilyProperty, _
        .Value = New FontFamily("Times New Roman")})

End With

With criteriaCancelButtonStyle

    .Setters.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.[Property] = ForegroundProperty, _

```

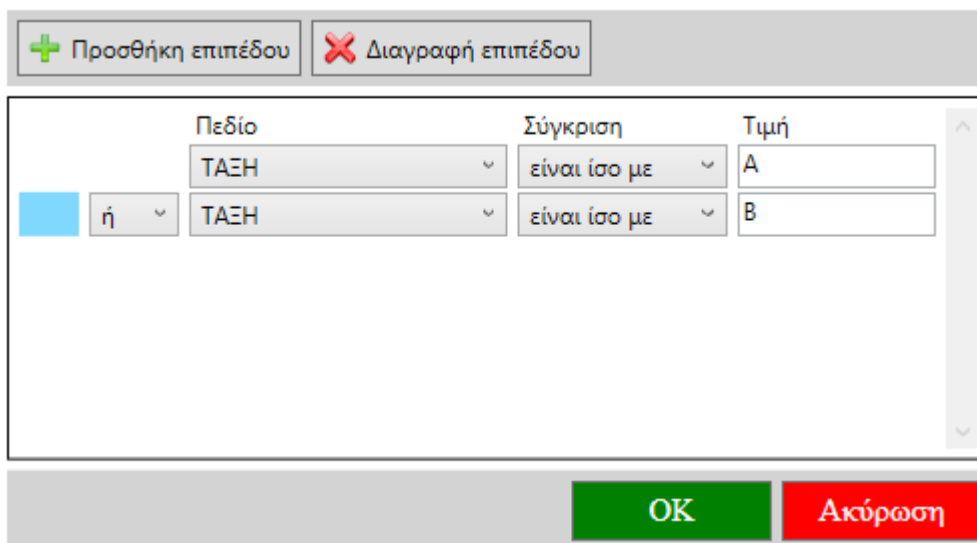
```
.Value = Brushes.White})  
.Setters.Add(New Setter With {.[Property] = BackgroundProperty, _  
.Value = Brushes.Red})  
.Setters.Add(New Setter With {.[Property] = FontFamilyProperty, _  
.Value = New FontFamily("Times New Roman")})
```

End With

With criteriaControl

```
.ButtonOKStyle = criteriaOKButtonStyle  
.ButtonCancelStyle = criteriaCancelButtonStyle
```

End With



	Πεδίο	Σύγκριση	Τιμή
	ΤΑΞΗ	είναι ίσο με	A
ή	ΤΑΞΗ	είναι ίσο με	B

## ButtonOKText

Καθορίζει το κείμενο (λεκτικό) για το **button "OK"**, τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property ButtonOKText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button "OK"**. Η default τιμή είναι **"OK"**.

### Dependency Property Information:

**Identifier field:** `ButtonOKTextProperty`

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το **button "Ακύρωση"**, το λεκτικό **"Done"** για το **button "OK"** και θέτουμε το **πλάτος των buttons** αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση **Button**.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    ButtonOKText="Done"
    ButtonCancelText="Cancel"
    ButtonsOKCancelWidth="80"
    ... />
```

**VB:**

```
criteriaControl.ButtonOKText = "Done"
criteriaControl.ButtonCancelText = "Cancel"
criteriaControl.ButtonsOKCancelWidth = 80
```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
<input type="checkbox"/>	και <input type="checkbox"/> ΤΜΗΜΑ	είναι ίσο με	A1

## ButtonOKStyle

Καθορίζει το **στυλ για το button "OK"**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ButtonOKStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "OK".  
Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonOKStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα "**CriteriaOKButtonStyle**" και το εφαρμόζουμε στο **button "OK"** και ένα δεύτερο **resource style** με όνομα "**CriteriaCancelButtonStyle**" και το εφαρμόζουμε στο **button "Ακύρωση"**.

**XAML:**

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

  <!--Ένα style για το OK button-->
  <Style x:Key="CriteriaOKButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Green"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

  </Style>
```

```

<!-- Ένα style για το Cancel button-->
<Style x:Key="CriteriaCancelButtonStyle" TargetType="{x:Type Button}">

    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{DynamicResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim criteriaOKButtonStyle As New Style(GetType(Button))
Dim criteriaCancelButtonStyle As New Style(GetType(Button))

With criteriaOKButtonStyle

    .Setters.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.[Property] = ForegroundProperty, _
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.[Property] = BackgroundProperty, _
        .Value = Brushes.Green})
    .Setters.Add(New Setter With {.[Property] = FontFamilyProperty, _
        .Value = New FontFamily("Times New Roman")})

End With

With criteriaCancelButtonStyle

    .Setters.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.[Property] = ForegroundProperty, _

```

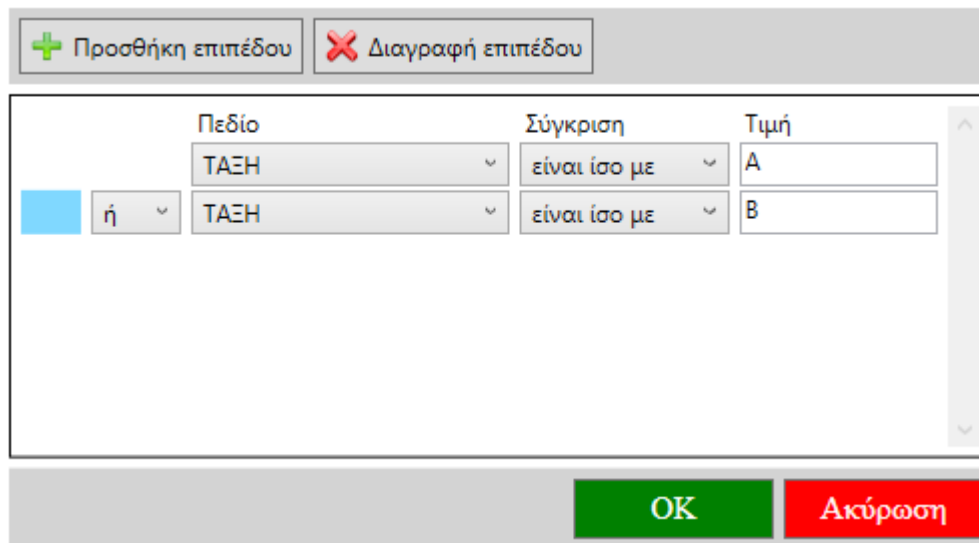
```
.Value = Brushes.White})  
.Setters.Add(New Setter With {.[Property] = BackgroundProperty, _  
.Value = Brushes.Red})  
.Setters.Add(New Setter With {.[Property] = FontFamilyProperty, _  
.Value = New FontFamily("Times New Roman")})
```

End With

With criteriaControl

```
.ButtonOKStyle = criteriaOKButtonStyle  
.ButtonCancelStyle = criteriaCancelButtonStyle
```

End With



	Πεδίο	Σύγκριση	Τιμή
	ΤΑΞΗ	είναι ίσο με	A
ή	ΤΑΞΗ	είναι ίσο με	B



## ButtonRemoveLevelText

Καθορίζει το κείμενο (λεκτικό) για το **button** "Διαγραφή επιπέδου", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonRemoveLevelText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button** "Διαγραφή επιπέδου". Η default τιμή είναι "Διαγραφή επιπέδου".

### Dependency Property Information:

**Identifier field:** ButtonRemoveLevelTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Add Level"** για το **button** "Προσθήκη επιπέδου" και **"Remove Level"** για το **button** "Διαγραφή επιπέδου" :

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ButtonAddLevelText="Add Level"
    ButtonRemoveLevelText="Remove Level"... />
```

#### VB:

```
criteriaControl.ButtonAddLevelText = "Add Level"
criteriaControl.ButtonRemoveLevelText = "Remove Level"
```

		Πεδίο	Σύγκριση	Τιμή
		ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
<input checked="" type="checkbox"/>	ή	ΕΠΩΝΥΜΟ	αρχίζει από	ΔΙΑ

## ButtonRemoveLevelStyle

Καθορίζει το **στυλ για το button "Διαγραφή επιπέδου"**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ButtonRemoveLevelStyle As Style
```

#### Τύπος: System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Διαγραφή επιπέδου". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonRemoveLevelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα "**CriteriaLevelButtonsStyle**" και το εφαρμόζουμε στα **buttons "Προσθήκη επιπέδου"** και "**Διαγραφή επιπέδου**" :

#### XAML:

```
<Window x:Class="WinCriteria"
  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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded" >

<Window.Resources >

  ...

  <!-- Ένα style για τα Level buttons-->
  <Style x:Key="CriteriaLevelButtonsStyle" TargetType="{x:Type Button}">

    <Setter Property="FontSize" Value="12"/>
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="Background" Value="LightBlue" />
    <Setter Property="FontFamily" Value="Times New Roman"/>

  </Style>
```

```

</Window.Resources>

<Grid >

...

<z:CriteriaControl Name="criteriaControl" Width="Auto" Margin="10"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
        ButtonAddLevelStyle="{StaticResource CriteriaLevelButtonsStyle}"
        ButtonRemoveLevelStyle="{StaticResource CriteriaLevelButtonsStyle}"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        ButtonOKStyle="{StaticResource CriteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource CriteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"
        CancelClicked="CriteriaControl_CancelClicked" />

</Grid>
</Window>

```

**VB:**

```

Dim criteriaLevelButtonStyle As New Style(GetType(Button))

With criteriaLevelButtonStyle

    .Setters.Add(New Setter With {.Property = FontSizeProperty, .Value = 12.0})
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _
        .Value = FontWeights.Bold})
    .Setters.Add(New Setter With {.Property = BackgroundProperty, _
        .Value = Brushes.LightBlue})
    .Setters.Add(New Setter With {.Property = FontFamilyProperty, _
        .Value = New FontFamily("Times New Roman")})

End With

With criteriaControl

    .ButtonAddLevelStyle = criteriaLevelButtonStyle
    .ButtonRemoveLevelStyle = criteriaLevelButtonStyle

End With

```

+ Προσθήκη επιπέδου		✖ Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τιμή	
<input type="checkbox"/> και	ΕΠΩΝΥΜΟ	τελειώνει σε	ΟΥ
	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μικρότερο ή ίσο του	6/6/2019 <input type="button" value="15"/>

## ButtonsLevelContainerBackground

Καθορίζει το **χρώμα φόντου** για το **container** των **buttons** "Προσθήκη επιπέδου" και "Διαγραφή επιπέδου", τύπου **Brush**.

**Σύνταξη:**

**VB:**

```
Public Property ButtonsLevelContainerBackground As Brush
```

**Τύπος:** **System.Windows.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο **Brush** για τον χρωματισμό του φόντου του **container** των **buttons** "Προσθήκη επιπέδου" και "Διαγραφή επιπέδου".

Η default τιμή είναι **Brushes.LightGray**.

**Dependency Property Information:**

**Identifier field:** **ButtonsLevelContainerBackgroundProperty**

**Παράδειγμα:**


Στο επόμενο παράδειγμα, καθορίζουμε το **χρώμα φόντου** του **container** των **buttons** επιπέδου κριτηρίου σε μία **απόχρωση του πορτοκαλί (Amber)**.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl" Width="Auto"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    ButtonsLevelContainerBackground="#FFD54F"
    ... />
```

**VB:**

```
criteriaControl.ButtonsLevelContainerBackground = _
    New SolidColorBrush(ColorConverter.ConvertFromString("#FFD54F"))
```

Πεδίο	Σύγκριση	Τιμή
 ΤΑΞΗ	είναι ίσο με	A

OK Ακύρωση

**Σημείωση:** Ένα καλό site για τιμές χρωμάτων είναι το <https://www.materialui.co/colors>

## ButtonsOKCancelContainerBackground

Καθορίζει το **χρώμα φόντου** για το container των buttons "OK" και "Ακύρωση", τύπου **Brush**.

### Σύνταξη:

#### VB:

```
Public Property ButtonsOKCancelContainerBackground As Brush
```

**Τύπος:** **System.Windows.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο Brush για τον χρωματισμό του φόντου του container των buttons "OK" και "Ακύρωση". Η default τιμή είναι Brushes.LightGray.

### Dependency Property Information:

**Identifier field:** ButtonsOKCancelContainerBackgroundProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το **χρώμα φόντου του container των buttons "OK" και "Ακύρωση"** σε μία απόχρωση του μωβ (**Purple**).

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl" Width="Auto"
    FieldsNames="ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΤΑΞΗ, ΤΜΗΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ, ΒΑΘΜΟΣ, ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME, FIRSTNAME, CLASS, DIVISION, REGISTRATION DATE, GRADE,
    IS ACTIVE"
    ButtonsOKCancelContainerBackground= "#CE93D8"
    ... />
```

#### VB:

```
criteriaControl.ButtonsOKCancelContainerBackground = _
    New SolidColorBrush(ColorConverter.ConvertFromString("#CE93D8"))
```



Πεδίο	Σύγκριση	Τιμή
ΤΑΞΗ	είναι ίσο με	A

OK Ακύρωση

**Σημείωση:** Ένα καλό site για τιμές χρωμάτων είναι το <https://www.materialui.co/colors>

## ButtonsOKCancelWidth

Καθορίζει το πλάτος, σε pixels, των buttons "OK" και "Ακύρωση". Είναι τύπου **Double**.

### Σύνταξη:

**VB:**

```
Public Property ButtonsOKCancelWidth As Double
```

**Τύπος: System.Double**

Προσδιορίζουμε μία τιμή Double που καθορίζει το πλάτος για τα buttons "OK" και "Ακύρωση". Η default τιμή είναι 100.

### Dependency Property Information:

**Identifier field:** ButtonsOKCancelWidthProperty

### Παρατηρήσεις:

Ο λόγος που υπάρχει αυτή η ιδιότητα είναι για να καθορίσουμε γρήγορα ένα ομοιόμορφο πλάτος για τα buttons "OK" και "Ακύρωση".

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το button **"Ακύρωση"**, το λεκτικό **"Done"** για το button **"OK"** και θέτουμε το πλάτος των buttons αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση Button.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    ButtonOKText="Done"
    ButtonCancelText="Cancel"
    ButtonsOKCancelWidth="80"
    ... />
```

**VB:**

```
criteriaControl.ButtonOKText = "Done"
criteriaControl.ButtonCancelText = "Cancel"
criteriaControl.ButtonsOKCancelWidth = 80
```

+ Προσθήκη επιπέδου		✖ Διαγραφή επιπέδου	
	Πεδίο	Σύγκριση	Τμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
και	ΤΜΗΜΑ	είναι ίσο με	A1

Done Cancel

## ComboBoxBooleanValueStyle

Καθορίζει το **στυλ για το ComboBox των τιμών Boolean**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ComboBoxBooleanValueStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο ComboBox των τιμών Boolean. Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxBooleanValueStyleProperty

### Παρατηρήσεις:

Όταν ο τύπος του πεδίου κριτηρίου είναι τύπου Boolean, τότε αυτόματα παρουσιάζεται στη στήλη Τιμή ένα ComboBox για επιλογή μεταξύ των δύο boolean τιμών. Το ComboBoxBooleanValueStyle αφορά αυτό το ComboBox.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"BooleanValueComboBoxItemStyle"** που αφορά το **style** για ένα **ComboBoxItem** και ένα δεύτερο **resource style** με όνομα **"BooleanValueComboBoxStyle"** που αφορά το **style** ενός **ComboBox**. Το πρώτο, κάνει ένα **ComboBoxItem** να είναι **έντονο μωβ και μεγαλύτερο μέγεθος όταν ο δείκτης του ποντικιού είναι πάνω του**. Το δεύτερο, χρησιμοποιεί το πρώτο και στο τέλος εφαρμόζεται στην ιδιότητα **ComboBoxBooleanValueStyle** με μία **StaticResource** έκφραση.

#### XAML:

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >
```

```

<!-- To style για το ComboBoxItem-->
<Style x:Key="BooleanValueComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

    <Style.Triggers >

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Magenta"/>
        </Trigger>

    </Style.Triggers>

</Style>

<!-- To style θα εφαρμοστεί στο BooleanValue ComboBox-->
<Style x:Key="BooleanValueComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource BooleanValueComboBoxItemStyle }"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{DynamicResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        ComboBoxBooleanValueStyle="{StaticResource BooleanValueComboBoxStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim booleanValueComboBoxItemStyle As New Style(GetType(ComboBoxItem))
Dim booleanValueComboBoxStyle As New Style(GetType(ComboBox))

Dim trigger As New Trigger With {.[Property] = IsMouseOverProperty, .Value = True}

```

With trigger.Setters

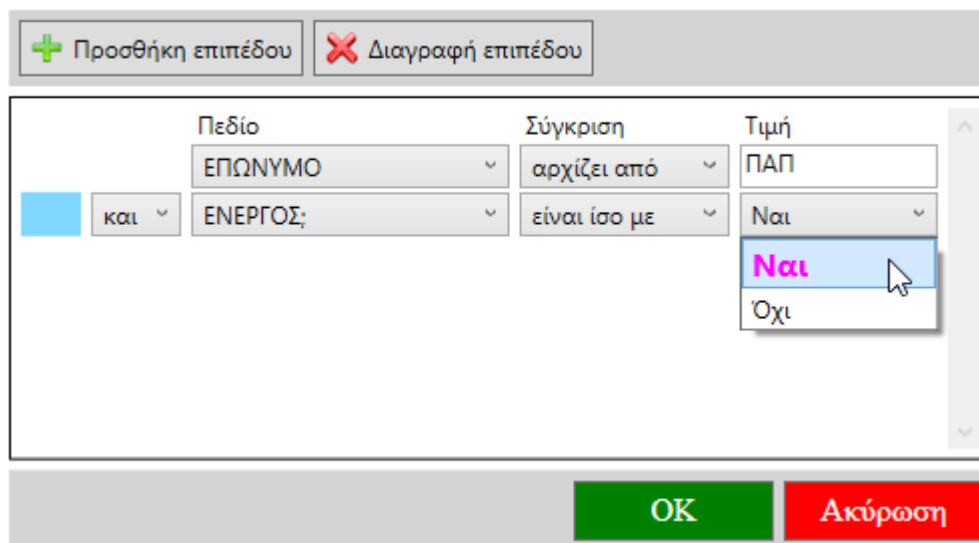
```
.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})  
.Add(New Setter With {.[Property] = ForegroundProperty, _  
    .Value = Brushes.Magenta})  
.Add(New Setter With {.[Property] = FontWeightProperty, _  
    .Value = FontWeights.Bold})
```

End With

```
booleanValueComboBoxItemStyle.Triggers.Add(trigger)
```

```
booleanValueComboBoxStyle.Setters.Add(New Setter  
    With {.[Property] = ItemsControl.ItemContainerStyleProperty, _  
    .Value = booleanValueComboBoxItemStyle})
```

```
criteriaControl.ComboBoxBooleanValueStyle = booleanValueComboBoxStyle
```



Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
και	ΕΝΕΡΓΟΣ;	είναι ίσο με
		Ναι
		Όχι

## ComboBoxComparisonStyle

Καθορίζει το **στυλ για το ComboBox των τελεστών Σύγκρισης**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ComboBoxComparisonStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο ComboBox των τελεστών σύγκρισης. Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxComparisonStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα "**ComparisonComboBoxItemStyle**" που αφορά το **style** για ένα **ComboBoxItem** και ένα δεύτερο **resource style** με όνομα "**ComparisonComboBoxStyle**" που αφορά το **style** ενός **ComboBox**. Το πρώτο, κάνει ένα **ComboBoxItem** να είναι **έντονο πορτοκαλί και μεγαλύτερο μέγεθος όταν ο δείκτης του ποντικιού είναι πάνω του**. Το δεύτερο, χρησιμοποιεί το πρώτο και στο τέλος εφαρμόζεται στην ιδιότητα **ComboBoxComparisonStyle** με μία **StaticResource** έκφραση.

**XAML:**

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- To style για το ComboBoxItem-->
    <Style x:Key="ComparisonComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

      <Style.Triggers >

        <Trigger Property="IsMouseOver" Value="True">
          <Setter Property="FontSize" Value="16"/>
        </Trigger>
      </Style.Triggers>
    </Style>
  </Window.Resources>
</Window>
```

```

        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Foreground" Value="Orange"/>
    </Trigger>

</Style.Triggers>

</Style>

<!-- To style θα εφαρμοστεί στο Comparison ComboBox-->
<Style x:Key="ComparisonComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource ComparisonComboBoxItemStyle }"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{DynamicResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        ComboBoxBComparisonStyle="{StaticResource ComparisonComboBoxStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim comparisonValueComboBoxItemStyle As New Style(GetType(ComboBoxItem))
Dim comparison ComboBoxStyle As New Style(GetType(ComboBox))

Dim trigger As New Trigger With {.[Property] = IsMouseOverProperty, .Value = True}

With trigger.Setters

    .Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Add(New Setter With {.[Property] = ForegroundProperty, .Value = Brushes.Orange})
    .Add(New Setter With {.[Property] = FontWeightProperty, _
        .Value = FontWeights.Bold})

End With

```



```
comparisonComboBoxItemStyle.Triggers.Add(trigger)

comparisonComboBoxStyle.Setters.Add(New Setter
    With {.Property = ItemsControl.ItemContainerStyleProperty, _
        .Value = comparisonComboBoxItemStyle})

criteriaControl.ComboBoxComparisonStyle = comparisonComboBoxStyle
```

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	

OK    Ακύρωση

## ComboBoxConnectorStyle

Καθορίζει το **στυλ για το ComboBox των τιμών Σύνδεσης**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ComboBoxConnectorStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο ComboBox των τιμών σύνδεσης των κριτηρίων. Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxConnectorStyleProperty

### Παρατηρήσεις:

Το ComboBox τιμών σύνδεσης εμφανίζεται όταν προσθέσουμε περισσότερα του ενός επίπεδα κριτηρίων.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"ConnectorComboBoxItemStyle"** που αφορά το **style** για ένα **ComboBoxItem** και ένα δεύτερο **resource style** με όνομα **"ConnectorComboBoxStyle"** που αφορά το **style** ενός **ComboBox**. Το πρώτο, κάνει ένα **ComboBoxItem** να είναι **έντονο κόκκινο και μεγαλύτερο μέγεθος όταν ο δείκτης του ποντικιού είναι πάνω του**. Το δεύτερο, χρησιμοποιεί το πρώτο και στο τέλος εφαρμόζεται στην ιδιότητα **ComboBoxConnectorStyle** με μία **StaticResource** έκφραση.

#### XAML:

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">
```

```
<Window.Resources >
```

```

<!-- To style για το ComboBoxItem-->
<Style x:Key="ConnectorComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

    <Style.Triggers >

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Magenta"/>
        </Trigger>

    </Style.Triggers>

</Style>

<!-- To style θα εφαρμοστεί στο Connector ComboBox-->
<Style x:Key="ConnectorComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource ConnectorComboBoxItemStyle }"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{DynamicResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        ComboBoxConnectorStyle="{StaticResource ConnectorComboBoxStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim connectorComboBoxItemStyle As New Style(GetType(ComboBoxItem))
Dim connectorComboBoxStyle As New Style(GetType(ComboBox))

Dim trigger As New Trigger With {.Property = IsMouseOverProperty, .Value = True}

With trigger.Setters

```

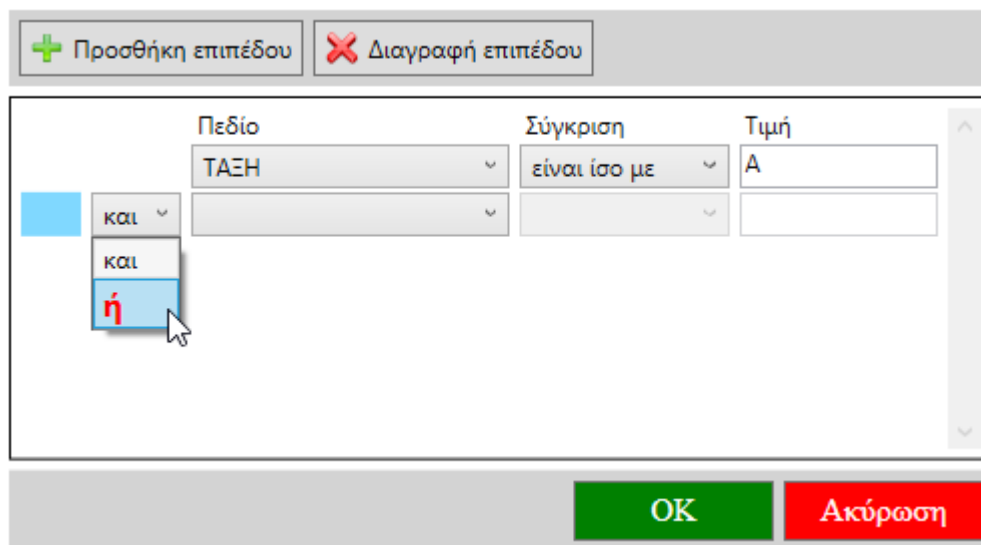
```
.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})  
.Add(New Setter With {.[Property] = ForegroundColorProperty, .Value = Brushes.Red})  
.Add(New Setter With {.[Property] = FontWeightProperty, _  
    .Value = FontWeights.Bold})
```

End With

```
connectorComboBoxItemStyle.Triggers.Add(trigger)
```

```
connectorComboBoxStyle.Setters.Add(New Setter  
    With {.[Property] = ItemsControl.ItemContainerStyleProperty, _  
        .Value = connectorComboBoxItemStyle})
```

```
criteriaControl.ComboBoxConnectorStyle = connectorComboBoxStyle
```



Πεδίο	Σύγκριση	Τιμή
ΤΑΞΗ	είναι ίσο με	A
καλ		

## ComboBoxConnectorWidth

Καθορίζει το πλάτος, σε pixels, του **ComboBox** των τιμών Σύνδεσης. Είναι τύπου **Double**.

### Σύνταξη:

#### VB:

```
Public Property ComboBoxConnectorWidth As Double
```

#### Τύπος: System.Double

Προσδιορίζουμε μία τιμή **Double** που καθορίζει το πλάτος του **ComboBox** των τιμών σύνδεσης κριτηρίων. Η default τιμή είναι 50.

### Dependency Property Information:

**Identifier field:** ComboBoxConnectorWidthProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"and"** για το την τιμή σύνδεσης **"AND"**, το λεκτικό **"or"** για την τιμή σύνδεσης **"OR"** και θέτουμε το πλάτος του **ComboBox** των τιμών σύνδεσης σε **60 pixels**.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ConnectorANDText="and"
    ConnectorORText="or"
    ComboBoxConnectorWidth="60"
    ... />
```

#### VB:

```
criteriaControl.ConnectorANDText = "and"
criteriaControl.ConnectorORText = "or"
criteriaControl.ComboBoxConnectorWidth = 60
```

	Πεδίο	Σύγκριση	Τιμή
<input type="checkbox"/>	ΤΑΞΗ	είναι ίσο με	A
<input type="checkbox"/>			

and  
and  
or

## ComboBoxFieldsNamesStyle

Καθορίζει το **στυλ για το ComboBox των πεδίων κριτηρίων**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ComboBoxFieldsNamesStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο ComboBox των πεδίων κριτηρίων. Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxFieldsNamesStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"FieldsNamesComboBoxItemStyle"** που αφορά το **style** για ένα **ComboBoxItem** και ένα δεύτερο **resource style** με όνομα **"FieldsNamesComboBoxStyle"** που αφορά το **style** ενός **ComboBox**. Το πρώτο, κάνει ένα **ComboBoxItem** να είναι **έντονο μπλε και μεγαλύτερο μέγεθος όταν ο δείκτης του ποντικιού είναι πάνω του**. Το δεύτερο, χρησιμοποιεί το πρώτο και στο τέλος εφαρμόζεται στην ιδιότητα **ComboBoxFieldsNamesStyle** με μία **StaticResource** έκφραση.

**XAML:**

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- To style για το ComboBoxItem-->
    <Style x:Key="FieldsNamesComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">

      <Style.Triggers >

        <Trigger Property="IsMouseOver" Value="True">
          <Setter Property="FontSize" Value="16"/>
        </Trigger>
      </Style.Triggers>
    </Style>
  </Window.Resources>
</Window>
```

```

        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Foreground" Value="Blue"/>
    </Trigger>

</Style.Triggers>

</Style>

<!-- To style του FieldsNames ComboBox-->
<Style x:Key="FieldsNamesComboBoxStyle" TargetType="{x:Type ComboBox }">

    <Setter Property="ItemContainerStyle"
        Value="{StaticResource FieldsNamesComboBoxItemStyle }"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim FieldsNamesComboBoxItemStyle As New Style(GetType(ComboBoxItem))
Dim FieldsNamesComboBoxStyle As New Style(GetType(ComboBox))

Dim trigger As New Trigger With {.[Property] = IsMouseOverProperty, .Value = True}

With trigger.Setters

    .Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Add(New Setter With {.[Property] = ForegroundProperty, .Value = Brushes.Blue})
    .Add(New Setter With {.[Property] = FontWeightProperty, _
        .Value = FontWeights.Bold})

End With

FieldsNamesComboBoxItemStyle.Triggers.Add(trigger)

```



```
FieldsNamesComboBoxStyle.Setters.Add(New Setter  
    With {.Property = ItemsControl.ItemContainerStyleProperty, _  
        .Value = FieldsNamesComboBoxItemStyle})
```

```
criteriaControl.ComboBoxFieldsNamesStyle = FieldsNamesComboBoxStyle
```

The screenshot displays a user interface for defining criteria. At the top, there are two buttons: a green '+ Προσθήκη επιπέδου' (Add level) and a red 'X Διαγραφή επιπέδου' (Delete level). Below these is a table with three columns: 'Πεδίο' (Field), 'Σύγκριση' (Comparison), and 'Τιμή' (Value). The first row has 'ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ' (Registration date) in the Field column, 'μεγαλύτερο από' (greater than) in the Comparison column, and '1/6/2019' in the Value column. A dropdown menu is open under the 'Πεδίο' column, showing a list of fields: ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, **ΤΑΞΗ** (highlighted), ΤΜΗΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ, ΒΑΘΜΟΣ, and ΕΝΕΡΓΟΣ;. The 'και' (and) operator is selected in the dropdown to the left of the table. At the bottom right, there are two buttons: a green 'ΟΚ' (OK) and a red 'Ακύρωση' (Cancel).

Πεδίο	Σύγκριση	Τιμή
ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019
και		

- ΕΠΩΝΥΜΟ
- ΟΝΟΜΑ
- ΤΑΞΗ**
- ΤΜΗΜΑ
- ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ
- ΒΑΘΜΟΣ
- ΕΝΕΡΓΟΣ;

ΟΚ    Ακύρωση

## ComparisonColumnHeaderText

Καθορίζει το κείμενο (λεκτικό) για τη στήλη των τιμών Σύγκρισης τύπου `String`.

### Σύνταξη:

**VB:**

```
Public Property ComparisonColumnHeaderText As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό της επικεφαλίδας της στήλης των τιμών σύγκρισης. Η default τιμή είναι "Σύγκριση".

### Dependency Property Information:

**Identifier field:** `ComparisonColumnHeaderTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "Field" για την επικεφαλίδα των πεδίων, το λεκτικό "Operator" για την επικεφαλίδα της στήλης σύγκρισης και το λεκτικό "Value" για την επικεφαλίδα των εισαγόμενων τιμών.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    FieldsColumnHeaderText="Field"
    ComparisonColumnHeaderText="Operator"
    ValueColumnHeaderText="Value" ... />
```

**VB:**

```
With criteriaControl
```

```
    .FieldsColumnHeaderText = "Field"
    .ComparisonColumnHeaderText = "Operator"
    .ComparisonColumnHeaderText = "Value"
```

```
End With
```

	Field	Operator	Value	
	ΤΑΞΗ	είναι ίσο με	A	
<input type="checkbox"/>	ή	ΤΑΞΗ	είναι ίσο με	B

## ComparisonOperatorBoolEqualText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "είναι ίσο με", για Boolean τύπο δεδομένων. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorBoolEqualText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "είναι ίσο με", για Boolean τύπο δεδομένων. Η default τιμή είναι "είναι ίσο με".

### Dependency Property Information:

**Identifier field:** `ComparisonOperatorBoolEqualTextProperty`

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό σύγκρισης (για τύπο Boolean είναι ένα το λεκτικό) της ισότητας σε "ισούται με".

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"  
    ComparisonOperatorBoolEqualText="ισούται με" ... />
```

#### VB:

```
criteriaControl.ComparisonOperatorBoolEqualText = "ισούται με"
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τιμή	
<input type="text" value="ΕΝΕΡΓΟΣ;"/>	<input type="text" value="ισούται με"/> <input type="text" value="ισούται με"/>	<input type="text" value="Ναι"/>	
<input type="button" value="OK"/>		<input type="button" value="Ακύρωση"/>	

## ComparisonOperatorBeginsWithText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "αρχίζει με", για String τύπο δεδομένων. Είναι τύπου String

### Σύνταξη:

**VB:**

```
Public Property ComparisonOperatorBeginsWithText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "αρχίζει με", για String τύπο δεδομένων. Η default τιμή είναι "αρχίζει με".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorBeginsWithTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τύπο String) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorEqualText="is equal"
    ComparisonOperatorBeginsWithText="starts with"
    ComparisonOperatorEndsWithText="ends with"
    ComparisonOperatorContainsText="contains"
    ComparisonOperatorNotEqualText="is not equal"
    ... />
```

**VB:**

```
With criteriaControl

    .ComparisonOperatorEqualText = "is equal"
    .ComparisonOperatorBeginsWithText = "starts with"
    .ComparisonOperatorEndsWithText = "ends with"
    .ComparisonOperatorContainsText = "contains"
    .ComparisonOperatorNotEqualText = "is not equal"

End With
```

**+ Προσθήκη επιπέδου**   **✖ Διαγραφή επιπέδου**

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	starts with is equal starts with ends with contains is not equal	

**OK**   **Ακύρωση**

## ComparisonOperatorContainsText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "περιέχει", για String τύπο δεδομένων. Είναι τύπου String.

**Σύνταξη:**

**VB:**

```
Public Property ComparisonOperatorContainsText As String
```

**Τύπος: System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "περιέχει", για String τύπο δεδομένων. Η default τιμή είναι "περιέχει".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorContainsTextProperty

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τύπο String) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorEqualText="is equal"
    ComparisonOperatorBeginsWithText="starts with"
    ComparisonOperatorEndsWithText="ends with"
    ComparisonOperatorContainsText="contains"
    ComparisonOperatorNotEqualText="is not equal"
    ... />
```

**VB:**

```
With criteriaControl
```

```
.ComparisonOperatorEqualText = "is equal"
.ComparisonOperatorBeginsWithText = "starts with"
.ComparisonOperatorEndsWithText = "ends with"
.ComparisonOperatorContainsText = "contains"
.ComparisonOperatorNotEqualText = "is not equal"
```

```
End With
```



**+ Προσθήκη επιπέδου**   **✖ Διαγραφή επιπέδου**

Πεδίο	Σύγκριση	Τμή
ΕΠΩΝΥΜΟ	starts with is equal starts with ends with contains is not equal	

**OK**   **Ακύρωση**

## ComparisonOperatorEndsWithText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "τελειώνει σε", για String τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorEndsWithText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "τελειώνει σε", για String τύπο δεδομένων. Η default τιμή είναι "τελειώνει σε".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorEndsWithTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τύπο String) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

#### XAML:



```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorEqualText="is equal"
    ComparisonOperatorBeginsWithText="starts with"
    ComparisonOperatorEndsWithText="ends with"
    ComparisonOperatorContainsText="contains"
    ComparisonOperatorNotEqualText="is not equal"
    ... />
```

#### VB:

```
With criteriaControl
```

```
.ComparisonOperatorEqualText = "is equal"
.ComparisonOperatorBeginsWithText = "starts with"
.ComparisonOperatorEndsWithText = "ends with"
.ComparisonOperatorContainsText = "contains"
.ComparisonOperatorNotEqualText = "is not equal"
```

```
End With
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τμή	
<input type="text" value="ΕΠΩΝΥΜΟ"/>	<input type="text" value="starts with"/>	<input type="text"/>	
	<ul style="list-style-type: none"><li>is equal</li><li>starts with</li><li>ends with</li><li>contains</li><li>is not equal</li></ul>		
		<input type="button" value="OK"/>	<input type="button" value="Ακύρωση"/>

## ComparisonOperatorEqualText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "είναι ίσο με", για String τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorEqualText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "είναι ίσο με", για String τύπο δεδομένων. Η default τιμή είναι "είναι ίσο με".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorEqualTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τύπο String) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

#### XAML:



```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorEqualText="is equal"
    ComparisonOperatorBeginsWithText="starts with"
    ComparisonOperatorEndsWithText="ends with"
    ComparisonOperatorContainsText="contains"
    ComparisonOperatorNotEqualText="is not equal"
    ... />
```

#### VB:

```
With criteriaControl

    .ComparisonOperatorEqualText = "is equal"
    .ComparisonOperatorBeginsWithText = "starts with"
    .ComparisonOperatorEndsWithText = "ends with"
    .ComparisonOperatorContainsText = "contains"
    .ComparisonOperatorNotEqualText = "is not equal"

End With
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τμή	
<input type="text" value="ΕΠΩΝΥΜΟ"/>	<input type="text" value="starts with"/>	<input type="text"/>	
	<ul style="list-style-type: none"><li>is equal</li><li><b>starts with</b></li><li>ends with</li><li>contains</li><li>is not equal</li></ul>		
		<input type="button" value="OK"/>	<input type="button" value="Ακύρωση"/>

## ComparisonOperatorNotEqualText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "δεν είναι ίσο με", για String τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorNotEqualText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "δεν είναι ίσο με", για String τύπο δεδομένων. Η default τιμή είναι "δεν είναι ίσο με".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorNotEqualTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τύπο String) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

#### XAML:



```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorEqualText="is equal"
    ComparisonOperatorBeginsWithText="starts with"
    ComparisonOperatorEndsWithText="ends with"
    ComparisonOperatorContainsText="contains"
    ComparisonOperatorNotEqualText="is not equal"
    ... />
```

#### VB:

```
With criteriaControl
```

```
.ComparisonOperatorEqualText = "is equal"
.ComparisonOperatorBeginsWithText = "starts with"
.ComparisonOperatorEndsWithText = "ends with"
.ComparisonOperatorContainsText = "contains"
.ComparisonOperatorNotEqualText = "is not equal"
```

```
End With
```

 Προσθήκη επιπέδου		 Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τμή	
<input type="text" value="ΕΠΩΝΥΜΟ"/>	<input type="text" value="starts with"/>	<input type="text"/>	
	<ul style="list-style-type: none"><li>is equal</li><li>starts with</li><li>ends with</li><li>contains</li><li>is not equal</li></ul>		
		<input type="button" value="OK"/>	<input type="button" value="Ακύρωση"/>

## ComparisonOperatorNumEqualText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "είναι ίσο με", για `Numeric` και `DateTime` τύπο δεδομένων. Είναι τύπου `String`.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorNumEqualText As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "είναι ίσο με", για `String` τύπο δεδομένων. Η default τιμή είναι "είναι ίσο με".

### Dependency Property Information:

**Identifier field:** `ComparisonOperatorNumEqualTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τους γενικούς τύπους `Numeric` και `DateTime`) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorNumEqualText="is equal"
    ComparisonOperatorNumGreaterThanText="greater than"
    ComparisonOperatorNumGreaterThanOrEqualText="greater than or equal to"
    ComparisonOperatorNumLessThanText="less than"
    ComparisonOperatorNumLessThanOrEqualText="less than or equal to"
    ComparisonOperatorNumNotEqualText="is not equal"
    ... />
```

#### VB:

```
With criteriaControl

    .ComparisonOperatorNumEqualText = "is equal"
    .ComparisonOperatorNumGreaterThanText = "greater than"
    .ComparisonOperatorNumGreaterThanOrEqualText = "greater than or equal to"
    .ComparisonOperatorNumLessThanText = "less than"
    .ComparisonOperatorNumLessThanOrEqualText = "less than or equal to"
    .ComparisonOperatorNumNotEqualText = "is not equal"

End With
```



Πεδίο	Σύγκριση	Τιμή
<input type="text" value="ΒΑΘΜΟΣ"/>	<input type="text" value="is equal"/> <input type="text" value="is equal"/> <input type="text" value="greater than"/> <input type="text" value="greater than or equal to"/> <input type="text" value="less than"/> <input type="text" value="less than or equal to"/> <input type="text" value="is not equal"/>	<input type="text"/>

## ComparisonOperatorNumGreaterThanOrEqualToText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μεγαλύτερο ή ίσο του", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

**VB:**

```
Public Property ComparisonOperatorNumGreaterThanOrEqualToText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "μεγαλύτερο ή ίσο του", για Numeric και DateTime τύπο δεδομένων. Η default τιμή είναι "μεγαλύτερο ή ίσο του".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorNumGreaterThanOrEqualToTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τους γενικούς τύπους Numeric και DateTime) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorNumEqualText ="is equal"
    ComparisonOperatorNumGreaterThanText="greater than"
    ComparisonOperatorNumGreaterThanOrEqualToText ="greater than or equal to"
    ComparisonOperatorNumLessThanText ="less than"
    ComparisonOperatorNumLessThanOrEqualToText="less than or equal to"
    ComparisonOperatorNumNotEqualText="is not equal"
    ... />
```

**VB:**

```
With criteriaControl

    .ComparisonOperatorNumEqualText = "is equal"
    .ComparisonOperatorNumGreaterThanText = "greater than"
    .ComparisonOperatorNumGreaterThanOrEqualToText = "greater than or equal to"
    .ComparisonOperatorNumLessThanText = "less than"
    .ComparisonOperatorNumLessThanOrEqualToText = "less than or equal to"
    .ComparisonOperatorNumNotEqualText = "is not equal"

End With
```

+ Προσθήκη επιπέδου ✖ Διαγραφή επιπέδου

Πεδίο	Σύγκριση	Τιμή
ΒΑΘΜΟΣ	is equal	

- is equal
- greater than
- greater than or equal to
- less than
- less than or equal to
- is not equal

OK Ακύρωση

## ComparisonOperatorNumGreaterThanText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μεγαλύτερο από", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.

**Σύνταξη:**

**VB:**

```
Public Property ComparisonOperatorNumGreaterThanText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "μεγαλύτερο από", για Numeric και DateTime τύπο δεδομένων. Η default τιμή είναι "μεγαλύτερο από".

**Dependency Property Information:**

**Identifier field:** ComparisonOperatorNumGreaterThanTextProperty

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τους γενικούς τύπους Numeric και DateTime) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorNumEqualText ="is equal"
    ComparisonOperatorNumGreaterThanText="greater than"
    ComparisonOperatorNumGreaterThanOrEqualText ="greater than or equal to"
    ComparisonOperatorNumLessThanText ="less than"
    ComparisonOperatorNumLessThanOrEqualText="less than or equal to"
    ComparisonOperatorNumNotEqualText="is not equal"
    ... />
```

**VB:**

```
With criteriaControl
```

```
.ComparisonOperatorNumEqualText = "is equal"
.ComparisonOperatorNumGreaterThanText = "greater than"
.ComparisonOperatorNumGreaterThanOrEqualText = "greater than or equal to"
.ComparisonOperatorNumLessThanText = "less than"
.ComparisonOperatorNumLessThanOrEqualText = "less than or equal to"
.ComparisonOperatorNumNotEqualText = "is not equal"
```

```
End With
```

+ Προσθήκη επιπέδου ✖ Διαγραφή επιπέδου

Πεδίο	Σύγκριση	Τιμή
ΒΑΘΜΟΣ	is equal	

is equal  
greater than  
greater than or equal to  
less than  
less than or equal to  
is not equal

OK Ακύρωση

## ComparisonOperatorNumLessThanOrEqualText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μικρότερο ή ίσο του", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorNumLessThanOrEqualText As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "μικρότερο ή ίσο του", για Numeric και DateTime τύπο δεδομένων. Η default τιμή είναι "μικρότερο ή ίσο του".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorNumLessThanOrEqualTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τους γενικούς τύπους Numeric και DateTime) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

#### XAML:



```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorNumEqualText ="is equal"
    ComparisonOperatorNumGreaterThanText="greater than"
    ComparisonOperatorNumGreaterThanOrEqualText ="greater than or equal to"
    ComparisonOperatorNumLessThanText ="less than"
    ComparisonOperatorNumLessThanOrEqualText="less than or equal to"
    ComparisonOperatorNumNotEqualText="is not equal"
    ... />
```

#### VB:

```
With criteriaControl
```



```
.ComparisonOperatorNumEqualText = "is equal"
.ComparisonOperatorNumGreaterThanText = "greater than"
.ComparisonOperatorNumGreaterThanOrEqualText = "greater than or equal to"
.ComparisonOperatorNumLessThanText = "less than"
.ComparisonOperatorNumLessThanOrEqualText = "less than or equal to"
.ComparisonOperatorNumNotEqualText = "is not equal"
```

```
End With
```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

Πεδίο	Σύγκριση	Τιμή
ΒΑΘΜΟΣ	is equal	

is equal  
greater than  
greater than or equal to  
less than  
less than or equal to  
is not equal

## ComparisonOperatorNumLessThanText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "μικρότερο από", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

#### VB:

```
Public Property ComparisonOperatorNumLessThanText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "μικρότερο από", για Numeric και DateTime τύπο δεδομένων. Η default τιμή είναι "μικρότερο από".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorNumLessThanTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τους γενικούς τύπους Numeric και DateTime) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorNumEqualText ="is equal"
    ComparisonOperatorNumGreaterThanText="greater than"
    ComparisonOperatorNumGreaterThanOrEqualText ="greater than or equal to"
    ComparisonOperatorNumLessThanText ="less than"
    ComparisonOperatorNumLessThanOrEqualText="less than or equal to"
    ComparisonOperatorNumNotEqualText="is not equal"
    ... />
```



#### VB:

```
With criteriaControl
```

```
.ComparisonOperatorNumEqualText = "is equal"
.ComparisonOperatorNumGreaterThanText = "greater than"
.ComparisonOperatorNumGreaterThanOrEqualText = "greater than or equal to"
.ComparisonOperatorNumLessThanText = "less than"
.ComparisonOperatorNumLessThanOrEqualText = "less than or equal to"
.ComparisonOperatorNumNotEqualText = "is not equal"
```



```
End With
```



 Προσθήκη επιπέδου  Διαγραφή επιπέδου

Πεδίο	Σύγκριση	Τιμή
ΒΑΘΜΟΣ	is equal	

is equal  
greater than  
greater than or equal to  
less than  
less than or equal to  
is not equal

## ComparisonOperatorNumNotEqualText

Καθορίζει το κείμενο (λεκτικό) για τον τελεστή σύγκρισης "δεν είναι ίσο με", για Numeric και DateTime τύπο δεδομένων. Είναι τύπου String.

### Σύνταξη:

**VB:**

```
Public Property ComparisonOperatorNumNotEqualText As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για τον τελεστή σύγκρισης "δεν είναι ίσο με", για String τύπο δεδομένων. Η default τιμή είναι "δεν είναι ίσο με".

### Dependency Property Information:

**Identifier field:** ComparisonOperatorNumNotEqualTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά σύγκρισης (για τους γενικούς τύπους Numeric και DateTime) αντικαθιστώντας τα από ελληνικά σε αγγλικά.

**XAML:**



```
<zeus:CriteriaControl Name="criteriaControl"
    ComparisonOperatorNumEqualText ="is equal"
    ComparisonOperatorNumGreaterThanText="greater than"
    ComparisonOperatorNumGreaterThanOrEqualText ="greater than or equal to"
    ComparisonOperatorNumLessThanText ="less than"
    ComparisonOperatorNumLessThanOrEqualText="less than or equal to"
    ComparisonOperatorNumNotEqualText="is not equal"
    ... />
```

**VB:**

```
With criteriaControl
```

```
.ComparisonOperatorNumEqualText = "is equal"
.ComparisonOperatorNumGreaterThanText = "greater than"
.ComparisonOperatorNumGreaterThanOrEqualText = "greater than or equal to"
.ComparisonOperatorNumLessThanText = "less than"
.ComparisonOperatorNumLessThanOrEqualText = "less than or equal to"
.ComparisonOperatorNumNotEqualText = "is not equal"
```

```
End With
```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

Πεδίο	Σύγκριση	Τιμή
ΒΑΘΜΟΣ	is equal	

is equal  
greater than  
greater than or equal to  
less than  
less than or equal to  
is not equal

**OK** **Ακύρωση**

## ConnectorANDText

Καθορίζει το κείμενο (λεκτικό) για τον συνδέτη κριτηρίων "AND", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ConnectorANDText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για τον συνδέτη κριτηρίων "AND". Η default τιμή είναι "και".

### Dependency Property Information:

**Identifier field:** ConnectorANDTextProperty

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "and" για τον συνδέτη "AND", το λεκτικό "or" για τον συνδέτη "OR" και θέτουμε το πλάτος του **ComboBox** των τιμών σύνδεσης σε **60 pixels**.


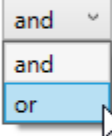
#### XAML:

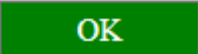
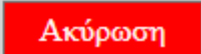
```
<zeus:CriteriaControl Name="criteriaControl"
    ConnectorANDText="and"
    ConnectorORText="or"
    ComboBoxConnectorWidth="60"
    ... />
```

#### VB:

```
criteriaControl.ConnectorANDText = "and"
criteriaControl.ConnectorORText = "or"
criteriaControl.ComboBoxConnectorWidth = 60
```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τιμή
	ΤΑΞΗ	είναι ίσο με	A
			

## ConnectorORText

Καθορίζει το κείμενο (λεκτικό) για τον συνδέτη κριτηρίων "OR", τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ConnectorOR Text As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για τον συνδέτη κριτηρίων "OR". Η default τιμή είναι "ή".

### Dependency Property Information:

**Identifier field:** ConnectorORTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "and" για τον συνδέτη "AND", το λεκτικό "or" για τον συνδέτη "OR" και θέτουμε το πλάτος του **ComboBox** των τιμών σύνδεσης σε **60 pixels**.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    ConnectorANDText="and"
    ConnectorORText="or"
    ComboBoxConnectorWidth="60"
    ... />
```

#### VB:

```
criteriaControl.ConnectorANDText = "and"
criteriaControl.ConnectorORText = "or"
criteriaControl.ComboBoxConnectorWidth = 60
```

	Πεδίο	Σύγκριση	Τιμή
<input type="checkbox"/>	ΤΑΞΗ	είναι ίσο με	A
<input type="checkbox"/>			

and  
and  
or

## CriteriaListInput

Καθορίζει μία λίστα αντικειμένων `CriteriaRowInfo`, καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) κριτηρίων. Είναι τύπου `List(of CriteriaRowInfo)`.

### Σύνταξη:

**VB:**

```
Public Property CriteriaListInput As List(Of CriteriaRowInfo)
```

**Τύπος:** `System.Collections.Generic.List(Of T)`

Προσδιορίζουμε μία λίστα αντικειμένων `CriteriaRowInfo`, που το καθένα περιέχει πληροφορίες για την κατασκευή ενός επιπέδου κριτηρίων, Η default τιμή είναι μία κενή λίστα.

### Dependency Property Information:

**Identifier field:** `CriteriaListInputProperty`

### Παρατηρήσεις:

Αν επιθυμούμε να κατασκευαστούν αυτόματα τα επίπεδα κριτηρίων στο `CriteriaControl`, ώστε ο χρήστης να τα βρει έτοιμα μπροστά του, θα χρησιμοποιήσουμε αυτήν την ιδιότητα. Θέτοντας σε αυτήν ένα αντικείμενο `List(Of CriteriaRowInfo)`, σε run-time mode, τα επίπεδα κατασκευάζονται αυτόματα.

Θα πρέπει να είμαστε προσεκτικοί στον καθορισμό των πεδίων στις ιδιότητες `FieldsNames` και `FieldsNamesOriginal` καθώς και στον τύπο που υποδηλώνεται στην `FieldType`, ώστε να συμπιπτουν στην ονοματοδοσία όπως αυτή έχει οριστεί πρωτύτερα στον XAML κώδικα κατά τον ορισμό του `CriteriaControl`.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τέσσερα έτοιμα επίπεδα κριτηρίων, με πεδία κριτηρίων **ΒΑΘΜΟΣ, ΤΑΞΗ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ, ΕΝΕΡΓΟΣ**; με πραγματικά ονόματα τα **GRADE, CLASS, REGISTRATION DATE, IS ACTIVE**. Η αναζήτηση λέει: "Να βρεθούν οι μαθητές που έχουν βαθμό  $\geq 15,5$ , είναι της Α' τάξης, έχουν εγγραφεί μετά την 1/6/2019 και είναι ενεργοί (δεν έχουν διακόψει).

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl" Width="Auto"
    FieldsNames="ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΤΑΞΗ, ΤΜΗΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ, ΒΑΘΜΟΣ, ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME, FIRSTNAME, CLASS, DIVISION, REGISTRATION DATE, GRADE,
```



```

        IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        ... />

```

**VB:**

```

criteriaControl.CriteriaListInput = New List(Of CriteriaRowInfo) From
{
    New CriteriaRowInfo With {.FieldName = "ΒΑΘΜΟΣ", _
        .FieldNameOriginal = "GRADE", _
        .FieldType = "Single",
        .ComparisonForNumAndDateValue = _
CriteriaControl.ComparisonNumAndDateOperatorsEnum.GreaterThanOrEqual, _
        .ValueText = "15,5"},



    New CriteriaRowInfo With {.ConnectorValue = _
CriteriaControl.ConnectorEnum.AndConnector, _
        .FieldName = "ΤΑΞΗ", _
        .FieldNameOriginal = "CLASS", _
        .FieldType = "String",
        .ComparisonForStringValue = _
CriteriaControl.ComparisonStringOperatorsEnum.Equal, _
        .ValueText = "A"},


    New CriteriaRowInfo With {.ConnectorValue = _
CriteriaControl.ConnectorEnum.AndConnector, _
        .FieldName = "ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ", _
        .FieldNameOriginal = "REGISTRATION DATE",
        .ComparisonForNumAndDateValue = _
CriteriaControl.ComparisonNumAndDateOperatorsEnum.GreaterThan,
        .ValueDate = "2019-06-01"},



    New CriteriaRowInfo With {.ConnectorValue = _
CriteriaControl.ConnectorEnum.AndConnector, _
        .FieldName = "ΕΝΕΡΓΟΣ;", _
        .FieldNameOriginal = "IS ACTIVE",
        .ComparisonForBooleanValue = _
CriteriaControl.ComparisonBooleanOperatorEnum.Equal,
        .ValueBoolean = CriteriaControl.BooleanValuesEnum.TrueValue}
}

```

Η παρακάτω εικόνα δείχνει το αποτέλεσμα.

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τιμή
	BAΘΜΟΣ	μεγαλύτερο ή ίσο του	15,5
και	ΤΑΞΗ	είναι ίσο με	A
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019 
και	ΕΝΕΡΓΟΣ;	είναι ίσο με	Ναι

## CriteriaListOutput

Επιστρέφει μία λίστα αντικειμένων **CriteriaRowInfo**, καθένα εκ των οποίων αναπαριστά ένα επίπεδο (γραμμή) κριτηρίων. Είναι τύπου **List(of CriteriaRowInfo)**.

### Σύνταξη:

#### VB:

```
Public ReadOnly Property CriteriaListOutput As List(Of CriteriaRowInfo)
```

#### Τύπος: **System.Collections.Generic.List(Of T)**

Επιστρέφει μία λίστα αντικειμένων **CriteriaRowInfo**, που το καθένα περιέχει πληροφορίες για κάθε επίπεδο κριτηρίων που έχει διαμορφώσει ο χρήστης, Η default τιμή είναι μία κενή λίστα.

### Dependency Property Information:

**Identifier field:** `CriteriaListOutputProperty`

### Παρατηρήσεις:

Οι πληροφορίες του κάθε **CriteriaRowInfo** αντικειμένου αφορούν: το display όνομα του πεδίου κριτηρίου(**FieldName**), το πραγματικό όνομα του πεδίου κριτηρίου, όπως είναι στη βάση (**FieldNameOriginal**), ο τύπος του (**FieldType**) και άλλες. Αυτή η λίστα διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί στο πρόγραμμα η public μέθοδος **CreateCriteria**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, στον **CriteriaControl OKClicked** event handler, διαβάζουμε ένα-ένα τα αντικείμενα **CriteriaRowInfo** της ιδιότητας **CriteriaListOutput** και τυπώνουμε συνολικά τα display names των πεδίων κριτηρίων, τα πραγματικά ονόματά τους και τον τύπο τους.

#### VB:

```
Private Sub CriteriaControl_OKClicked(sender As Object, e As RoutedEventArgs)

    Dim cList As String = ""

    For Each crit As CriteriaRowInfo In criteriaControl.CriteriaListOutput

        cList += String.Format("{0} {1} {2}" & vbCrLf, _
```

```
crit.FieldName, crit.FieldNameOriginal, crit.FieldType)
```

Next

```
MessageBox.Show(cList)
```

End Sub

Η παρακάτω εικόνα δείχνει πώς έχει διαμορφωθεί το CriteriaControl με βάση τις επιλογές του χρήστη...

	Πεδίο	Σύγκριση	Τιμή
	ΤΑΞΗ	είναι ίσο με	A
και	ΤΜΗΜΑ	είναι ίσο με	A1
και	ΒΑΘΜΟΣ	μεγαλύτερο από	15,5

... και όταν πατήσει το button OK εμφανίζεται η λίστα των συγκεκριμένων πληροφοριών για κάθε αντικείμενο CriteriaRowInfo:

ΤΑΞΗ CLASS String  
ΤΜΗΜΑ DIVISION String  
ΒΑΘΜΟΣ GRADE Single

OK

Φυσικά, σε μία πραγματική εφαρμογή, η ιδιότητα CriteriaListOutput θα μπορούσε να αξιοποιηθεί διαφορετικά.

## CriteriaStringOutput

Επιστρέφει ένα string με τα πεδία κριτηρίων (πραγματικά ονόματα, όπως είναι στη Βάση) τον τελεστή σύγκρισης, την τιμή μαζί με την σύνδεση (για το μετά το πρώτο επίπεδο), διαχωρισμένα με κόμμα.

### Σύνταξη:

#### VB:

```
Public ReadOnly Property CriteriaStringOutput As String
```

#### Τύπος: System.String

Επιστρέφει ένα String, που περιέχει τα πεδία κριτηρίων (τα πραγματικά ονόματα, όπως δηλώνονται στην ιδιότητα FieldsNamesOriginal), τον τελεστή σύγκρισης, την τιμή και τον συνδέτη (για μετά το πρώτο επίπεδο), με βάση τις επιλογές του χρήστη, Η default τιμή είναι String.Empty.

### Dependency Property Information:

**Identifier field:** CriteriaStringOutputProperty

### Παρατηρήσεις:

Ένα παράδειγμα είναι "LASTNAME LIKE ΠΑΠ% AND CLASS='A' AND GRADE<=15.5". Αυτό το string μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Where μίας SQL select query. Η CriteriaStringOutput διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί στο πρόγραμμα η public μέθοδος CreateCriteria.

#### **Το περιεχόμενο της CriteriaStringOutput εξαρτάται από την ιδιότητα DatabaseFormat.**

Κι αυτό γιατί υπάρχει μία μικρή διαφορά στο πώς διαχειρίζεται η Access, η SQLite3 και ο SQL Server τις ημερομηνίες. Επίσης, ο SQL Server για τον τύπο Boolean δεν έχει τιμές False, True αλλά ισοδύναμο τύπο Bit με 0=False και 1=True.

### Παράδειγμα:

Στο επόμενο παράδειγμα, στον CriteriaControl OKClicked event handler, τυπώνουμε την τιμή της ιδιότητας CriteriaStringOutput.

#### VB:

```
Private Sub CriteriaControl1_OKClicked(sender As Object, e As RoutedEventArgs)
```

```
MessageBox.Show(criteriaControl.CriteriaStringOutput)
```

End Sub

Η παρακάτω εικόνα δείχνει πώς έχει διαμορφωθεί το CriteriaControl με βάση τις επιλογές του χρήστη...

	Πεδίο	Σύγκριση	Τιμή
	ΒΑΘΜΟΣ	μεγαλύτερο ή ίσο του	15,5
και	ΤΑΞΗ	είναι ίσο με	A
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019
και	ΕΝΕΡΓΟΣ;	είναι ίσο με	Ναι

... και όταν πατήσει το button OK εμφανίζεται το περιεχόμενο της CriteriaStringOutput. **Να σημειωθεί ότι το περιεχόμενο της CriteriaStringOutput εξαρτάται από την ιδιότητα DatabaseFormat.**

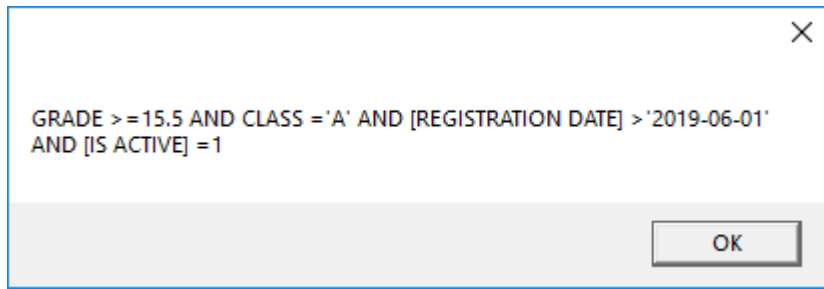
Για Access:

```
GRADE >= 15.5 AND CLASS = 'A' AND [REGISTRATION DATE] > #06/01/2019# AND [IS ACTIVE] = True
```

Για SQLite3:

```
GRADE >= '15.5' AND CLASS = 'A' AND [REGISTRATION DATE] > date('2019-06-01') AND [IS ACTIVE] = True
```

Για SQL Server:



## DatabaseFormat

Καθορίζει τον **τύπο της Βάσης Δεδομένων**, ώστε να δημιουργήσει σωστά την τιμή της ιδιότητας **CriteriaStringOutput**. Είναι τύπου **DatabaseFormatEnum**.

### Σύνταξη:

**VB:**

```
Public Property DatabaseFormat As DatabaseFormatEnum
```

**Τύπος: CriteriaControl.DatabaseFormatEnum**

Προσδιορίζουμε μία τιμή DatabaseFormatEnum, που δηλώνει τον τύπο της Βάσης Δεδομένων, για την οποία θα φτιαχτεί σωστά το CriteriaStringOutput. Η default τιμή είναι DatabaseFormatEnum.Access2007.

### Dependency Property Information:

**Identifier field:** DatabaseFormatProperty

### Παρατηρήσεις:

Το περιεχόμενο της CriteriaStringOutput μπορεί να χρησιμοποιηθεί άμεσα στο τμήμα Where μίας SQL select query. Η CriteriaStringOutput διαμορφώνεται αυτόματα μόλις ο χρήστης πατήσει το button "OK" ή κληθεί στο πρόγραμμα η public μέθοδος CreateCriteria.

**Το περιεχόμενο της CriteriatStringOutput εξαρτάται από την ιδιότητα DatabaseFormat.** Κι αυτό γιατί υπάρχει μία μικρή διαφορά στο πώς διαχειρίζεται η Access, η SQLite3 και ο SQL Server τις ημερομηνίες. Επίσης, ο SQL Server για τον τύπο Boolean δεν έχει τιμές False, True αλλά ισοδύναμο τύπο Bit με 0=False και 1=True.

### Παράδειγμα:

Στο επόμενο παράδειγμα, στον CriteriaControl OKClicked event handler, τυπώνουμε την τιμή της ιδιότητας CriteriaStringOutput.

**VB:**

```
Private Sub CriteriaControl1_OKClicked(sender As Object, e As RoutedEventArgs)  
    MessageBox.Show(criteriaControl.CriteriaStringOutput)  
End Sub
```



Η παρακάτω εικόνα δείχνει πώς έχει διαμορφωθεί το CriteriaControl με βάση τις επιλογές του χρήστη...

	Πεδίο	Σύγκριση	Τιμή
	BAΘΜΟΣ	μεγαλύτερο ή ίσο του	15,5
και	ΤΑΞΗ	είναι ίσο με	A
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019
και	ΕΝΕΡΓΟΣ;	είναι ίσο με	Ναι

Buttons: Προσθήκη επιπέδου (Add level), Διαγραφή επιπέδου (Delete level), OK, Ακύρωση (Cancel)

... και όταν πατήσει το button OK εμφανίζεται το περιεχόμενο της CriteriaStringOutput. **Να σημειωθεί ότι το περιεχόμενο της CriteriaStringOutput εξαρτάται από την ιδιότητα DatabaseFormat.**

Για Access:

```
GRADE >= 15.5 AND CLASS = 'A' AND [REGISTRATION DATE] > #06/01/2019# AND [IS ACTIVE] = True
```

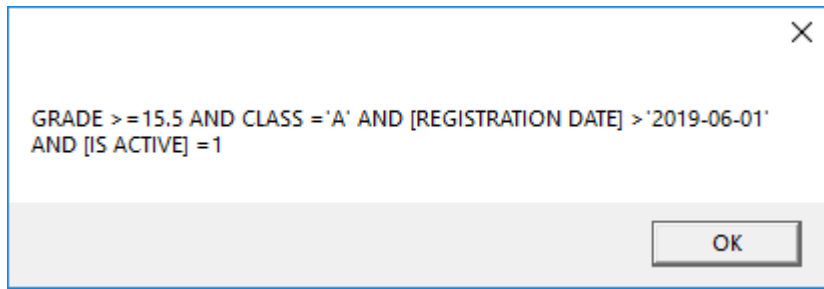
OK

Για SQLite3:

```
GRADE >= '15.5' AND CLASS = 'A' AND [REGISTRATION DATE] > date('2019-06-01') AND [IS ACTIVE] = True
```

OK

Για SQL Server:



## DatePickerCalendarStyle

Καθορίζει το **στυλ για το Calendar του DatePicker που χρησιμοποιείται για επιλογή ημερομηνιακής τιμής από τον χρήστη**. Είναι τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property DatePickerCalendarStyle As Style
```

#### Τύπος: System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο Calendar του DatePicker, που χρησιμοποιείται για επιλογή ημερομηνίας από τον χρήστη. Το default style είναι αυτό ενός Calendar element.

### Dependency Property Information:

**Identifier field:** DatePickerCalendarStyleProperty

### Παρατηρήσεις:

Το **DatePicker control εμφανίζεται αυτόματα (αντί TextBox) όταν ο τύπος του πεδίου είναι DateTime/Date**. Χρησιμοποιώντας την ιδιότητα DatePickerCalendarStyle μπορούμε να προσδώσουμε ένα ιδιαίτερο στυλ στο πτυσσόμενο Calendar του DatePicker.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"dtPickerStyle"** και το εφαρμόζουμε στην **ιδιότητα DatePickerStyle** και ένα δεύτερο **resource style** με όνομα **"calendarStyle"** και το εφαρμόζουμε στην **ιδιότητα DatePickerCalendarStyle**.

#### XAML:

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >
```

```

<!--Ένα style για το DatePicker -->
<Style x:Key="dtPickerStyle" TargetType="DatePicker">

    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="Yellow"/>

</Style>

<!--Ένα style για το Calendar του DatePicker-->
<Style x:Key="calendarStyle" TargetType="Calendar" >

    <Setter Property="Background" Value="LightBlue"/>
    <Setter Property="BorderBrush" Value="DarkBlue"/>
    <Setter Property="BorderThickness" Value="5"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{StaticResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim dtPickerStyle As New Style(GetType(DatePicker))
Dim calendarStyle As New Style(GetType(Calendar))

With dtPickerStyle.Setters

    .Add(New Setter With {.Property = ForegroundProperty, .Value = Brushes.Black})
    .Add(New Setter With {.Property = DatePicker.BackgroundProperty, _
        .Value = Brushes.Yellow})

End With

With calendarStyle.Setters

```

```
.Add(New Setter With {.[Property] = BorderBrushProperty, _  
    .Value = Brushes.DarkBlue})  
.Add(New Setter With {.[Property] = BorderThicknessProperty, _  
    .Value = New Thickness(5)})  
.Add(New Setter With {.[Property] = Calendar.BackgroundProperty, _  
    .Value = Brushes.LightBlue})
```

End With

With criteriaControl

```
.DatePickerStyle = dtPickerStyle  
.DatePickerCalendarStyle = calendarStyle
```

End With

The screenshot shows a criteria control interface with two buttons at the top: "+ Προσθήκη επιπέδου" and "X Διαγραφή επιπέδου". Below is a table with columns: Πεδίο, Σύγκριση, and Τιμή. The table contains two rows of criteria. The second row is highlighted in yellow, and a calendar is open over its value field.

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
και	είναι ίσο με	3/7/2019

The calendar is for July 2019 (Ιούλιος 2019) and shows the date 3/7/2019 selected. The calendar grid is as follows:

Δε	Τρ	Τε	Πε	Πα	Σα	Κυ
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

## DatePickerStyle

Καθορίζει το **στυλ για το DatePicker** που χρησιμοποιείται για επιλογή ημερομηνιακής τιμής από τον χρήστη. Είναι τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property DatePickerStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο DatePicker, που χρησιμοποιείται για επιλογή ημερομηνίας από τον χρήστη.

Το default style είναι αυτό ενός DatePicker element.

### Dependency Property Information:

**Identifier field:** DatePickerStyleProperty

### Παρατηρήσεις:

Το **DatePicker control εμφανίζεται αυτόματα (αντί TextBox)** όταν ο τύπος του πεδίου είναι **DateTime/Date**. Χρησιμοποιώντας την ιδιότητα **DatePickerStyle** μπορούμε να προσδώσουμε ένα ιδιαίτερο στυλ στο DatePicker.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"dtPickerStyle"** και το εφαρμόζουμε στην **ιδιότητα DatePickerStyle** και ένα δεύτερο **resource style** με όνομα **"calendarStyle"** και το εφαρμόζουμε στην **ιδιότητα DatePickerCalendarStyle**.

#### XAML:

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >
```

```

<!--Ένα style για το DatePicker -->
<Style x:Key="dtPickerStyle" TargetType="DatePicker">

    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="Yellow"/>

</Style>

<!--Ένα style για το Calendar του DatePicker-->
<Style x:Key="calendarStyle" TargetType="Calendar" >

    <Setter Property="Background" Value="LightBlue"/>
    <Setter Property="BorderBrush" Value="DarkBlue"/>
    <Setter Property="BorderThickness" Value="5"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{StaticResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim dtPickerStyle As New Style(GetType(DatePicker))
Dim calendarStyle As New Style(GetType(Calendar))

With dtPickerStyle.Setters

    .Add(New Setter With {.Property = ForegroundProperty, .Value = Brushes.Black})
    .Add(New Setter With {.Property = DatePicker.BackgroundProperty, _
        .Value = Brushes.Yellow})

End With

With calendarStyle.Setters

```

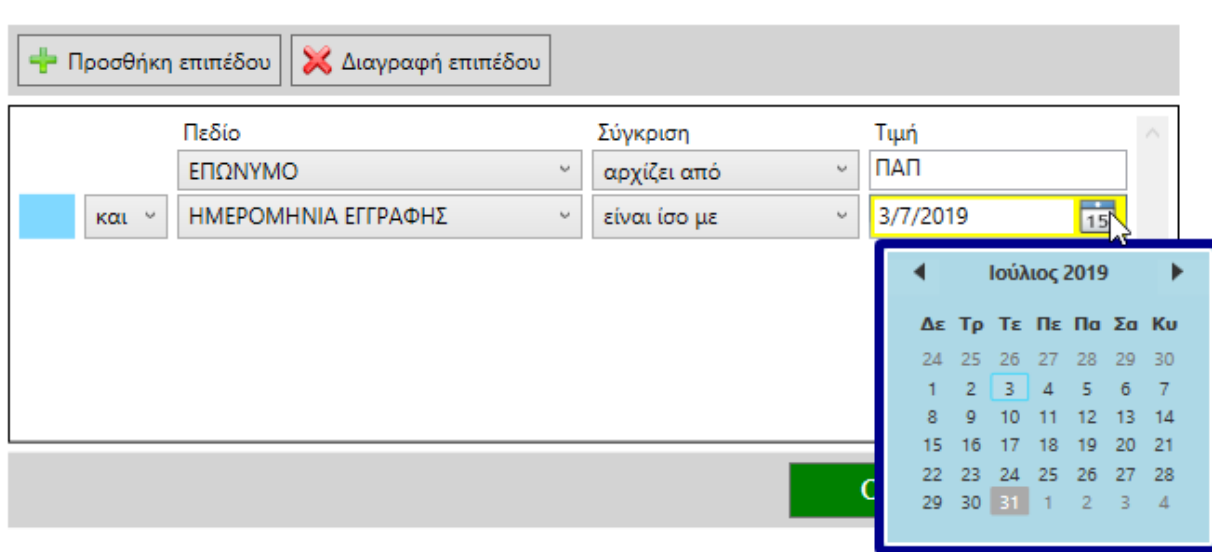
```
.Add(New Setter With {.[Property] = BorderBrushProperty, _  
    .Value = Brushes.DarkBlue})  
.Add(New Setter With {.[Property] = BorderThicknessProperty, _  
    .Value = New Thickness(5)})  
.Add(New Setter With {.[Property] = Calendar.BackgroundProperty, _  
    .Value = Brushes.LightBlue})
```

End With

With criteriaControl

```
.DatePickerStyle = dtPickerStyle  
.DatePickerCalendarStyle = calendarStyle
```

End With





## FieldsColumnHeaderText

Καθορίζει το κείμενο (λεκτικό) για τη στήλη των πεδίων κριτηρίων, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property FieldsColumnHeaderText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό της επικεφαλίδας της στήλης των πεδίων κριτηρίων.

Η default τιμή είναι "Πεδίο".

### Dependency Property Information:

**Identifier field:** `FieldsColumnHeaderTextProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Field"** για την επικεφαλίδα των πεδίων, το λεκτικό **"Operator"** για την επικεφαλίδα της στήλης σύγκρισης και το λεκτικό **"Value"** για την επικεφαλίδα των εισαγόμενων τιμών.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"
    FieldsColumnHeaderText="Field"
    ComparisonColumnHeaderText="Operator"
    ValueColumnHeaderText="Value" ... />
```

#### VB:

```
With criteriaControl
```

```
    .FieldsColumnHeaderText = "Field"
    .ComparisonColumnHeaderText = "Operator"
    .ComparisonColumnHeaderText = "Value"
```

```
End With
```

	Field	Operator	Value	
	ΤΑΞΗ	είναι ίσο με	A	
<input type="checkbox"/>	ή	ΤΑΞΗ	είναι ίσο με	B

## FieldsDataTypes

Καθορίζει τα ονόματα των τύπων δεδομένων για τα πεδία των κριτηρίων. Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FieldsDataTypes As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που περιέχει, διαχωρισμένα με κόμμα, τα ονόματα των τύπων δεδομένων για τα πεδία κριτηρίων.

Η default τιμή είναι **String.Empty**.

### Dependency Property Information:

**Identifier field:** `FieldsDataTypesProperty`

### Παρατηρήσεις:

Τα ονόματα των τύπων είναι: **"Byte", "Short", "Integer", "Long", "Single", "Double", "Decimal", "DateTime", "Date", "String", "Boolean"**.

Πρέπει να υπάρχει αντιστοιχία θέσης μεταξύ των περιεχομένων των ιδιοτήτων **"FieldsNames", "FieldsNamesOriginal"** και **"FieldsTypes"**. Ένα παράδειγμα είναι το εξής: **FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΒΑΘΜΟΣ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ"** , **FieldsNamesOriginal="LASTNAME,FIRSTNAME,GRADE,REGISTRATION DATE"** και οι τύποι των πεδίων **FieldsDataTypes="String,String,Single,Date"**.

Να σημειώσουμε ότι, εκατέρωθεν του κόμματος δεν πρέπει να υπάρχουν κενά. Για παράδειγμα, η τιμή **"String,String,,Single,Date"** είναι έγκυρη ενώ η **"String,String,Grade, Date"** όχι. Συνεπώς, θα πρέπει να είμαστε προσεκτικοί στον καθορισμό της ιδιότητας.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα πεδία κριτηρίων (**display names**) **"ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"** για επιλογή από τον χρήστη. Τα πραγματικά ονόματα, όμως, των πεδίων κριτηρίων (όπως είναι στη Βάση) είναι **"LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,IS ACTIVE"** που ορίζονται στην ιδιότητα **FieldsNamesOriginal**.

Οι τύποι των πεδίων, που τίθενται στην ιδιότητα **FieldsDataTypes** είναι **"String,String,String,String,Date,Single,Boolean"**.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl" Margin="16"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
    SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
    DatePickerStyle="{DynamicResource dtPickerStyle}"
    DatePickerCalendarStyle="{StaticResource calendarStyle}"
    ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
    ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
    OKClicked="CriteriaControl_OKClicked"/>
```

**VB:**

With criteriaControl

```
.FieldsNames = "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
.FieldsNamesOriginal = "LASTNAME,FIRSTNAME,CLASS,DIVISION,
    REGISTRATION DATE,GRADE,IS ACTIVE"
.FieldsDataTypes = "String,String,String,String,Date,Single,Boolean"
```

End With

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
ΤΑΞΗ	είναι ίσο με	B

## FieldsNames

Καθορίζει τα ονόματα των πεδίων κριτηρίων, όπως θα εμφανίζονται στο αντίστοιχο `ComboBox`. Είναι τύπου `String`.

### Σύνταξη:

**VB:**

```
Public Property FieldsNames As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που περιέχει, διαχωρισμένα με κόμμα, τα ονόματα των πεδίων κριτηρίων. Η default τιμή είναι `String.Empty`.

### Dependency Property Information:

**Identifier field:** `FieldsNamesProperty`

### Παρατηρήσεις:

Πρόκειται για τα `display names` των πεδίων κριτηρίων, δηλαδή όπως θα εμφανίζονται στο `ComboBox`. Τα πραγματικά ονόματα, δηλαδή όπως έχουν οριστεί στην πηγή (π.χ. στη Βάση) καθορίζονται στην άλλη ιδιότητα `FieldsNamesOriginal`.

Να σημειώσουμε ότι, εκατέρωθεν του κόμματος δεν πρέπει να υπάρχουν κενά. Για παράδειγμα, η τιμή "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΠΑΤΡΩΝΥΜΟ" είναι έγκυρη ενώ η "ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΠΑΤΡΩΝΥΜΟ" όχι. Συνεπώς, θα πρέπει να είμαστε προσεκτικοί στον καθορισμό της ιδιότητας. Φυσικά, το όνομα ενός πεδίου μπορεί να έχει κενά. Για παράδειγμα, "ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ, ΜΙΣΘΟΣ". Εσωτερικά το `CriteriaControl` τοποθετεί αγκύλες [ ], που ως γνωστόν χρησιμοποιούν οι `Access`, `SQLite` και `Sql Server`.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα πεδία κριτηρίων (`display names`) "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;" για επιλογή από τον χρήστη. Τα πραγματικά ονόματα, όμως, των πεδίων κριτηρίων (όπως είναι στη Βάση) είναι "LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,IS ACTIVE" που ορίζονται στην ιδιότητα `FieldsNamesOriginal`. Οι τύποι των πεδίων, που τίθενται στην ιδιότητα `FieldsDataTypes` είναι "String,String,String,String,Date,Single,Boolean".

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl" Margin="16"
```

```

FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
IS ACTIVE"
FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
DatePickerStyle="{DynamicResource dtPickerStyle}"
DatePickerCalendarStyle="{StaticResource calendarStyle}"
ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
OKClicked="CriteriaControl_OKClicked"/>

```

**VB:**

With criteriaControl

```

.FieldsNames = "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
.FieldsNamesOriginal = "LASTNAME,FIRSTNAME,CLASS,DIVISION,
REGISTRATION DATE,GRADE,IS ACTIVE"
.FieldsDataTypes = "String,String,String,String,Date,Single,Boolean"

```

End With

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
ΤΑΞΗ	είναι ίσο με	B

## FieldsNamesOriginal

Καθορίζει τα πραγματικά ονόματα των πεδίων κριτηρίων, όπως υπάρχουν στη Βάση. Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FieldsNamesOriginal As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που περιέχει, διαχωρισμένα με κόμμα, τα πραγματικά ονόματα των πεδίων κριτηρίων. Η default τιμή είναι **String.Empty**.

### Dependency Property Information:

**Identifier field:** **FieldsNamesOriginalProperty**

### Παρατηρήσεις:

Πρόκειται για τα πραγματικά ονόματα των πεδίων κριτηρίων, δηλαδή όπως υπάρχουν στη Βάση. Για τα display ονόματα, δηλαδή όπως αυτά θα εμφανιστούν προς επιλογή στο σχετικό **ComboBox** καθορίζονται στην άλλη ιδιότητα **FieldsNames**.

Να σημειώσουμε ότι, εκατέρωθεν του κόμματος δεν πρέπει να υπάρχουν κενά. Για παράδειγμα, η τιμή **"FIRSTNAME, LASTNAME, FATHER NAME"** είναι έγκυρη ενώ η **"FIRSTNAME, LASTNAME, FATHER NAME"** όχι. Συνεπώς, θα πρέπει να είμαστε προσεκτικοί στον καθορισμό της ιδιότητας. Φυσικά, το όνομα ενός πεδίου μπορεί να έχει κενά., όπως το **"FATHER NAME"**. Εσωτερικά το **CriteriaControl** τοποθετεί αγκύλες [ ], που ως γνωστόν χρησιμοποιούν οι **Access**, **SQLite** και **Sql Server**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα πεδία κριτηρίων (**display names**) **"ΕΠΩΝΥΜΟ, ΟΝΟΜΑ, ΤΑΞΗ, ΤΜΗΜΑ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ, ΒΑΘΜΟΣ, ΕΝΕΡΓΟΣ;"** για επιλογή από τον χρήστη. Τα πραγματικά ονόματα, όμως, των πεδίων κριτηρίων (όπως είναι στη Βάση) είναι **"LASTNAME, FIRSTNAME, CLASS, DIVISION, REGISTRATION DATE, GRADE, IS ACTIVE"** που ορίζονται στην ιδιότητα **FieldsNamesOriginal**. Οι τύποι των πεδίων, που τίθενται στην ιδιότητα **FieldsDataTypes** είναι **"String, String, String, String, Date, Single, Boolean"**.

**XAML:**

```
<zeus:CriteriaControl Name="criteriaControl" Margin="16"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal = "LASTNAME, FIRSTNAME, CLASS, DIVISION, REGISTRATION DATE, GRADE,
        IS ACTIVE"
    FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
    SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
    DatePickerStyle="{DynamicResource dtPickerStyle}"
    DatePickerCalendarStyle="{StaticResource calendarStyle}"
    ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
    ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
    OKClicked="CriteriaControl_OKClicked"/>
```

**VB:**

With criteriaControl

```
.FieldsNames = "ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
.FieldsNamesOriginal = "LASTNAME, FIRSTNAME, CLASS, DIVISION,
    REGISTRATION DATE, GRADE, IS ACTIVE"
.FieldsDataTypes = "String,String,String,String,Date,Single,Boolean"
```

End With

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
ΤΑΞΗ	είναι ίσο με	B



## GridCriteriaRowsContainerBackground

Καθορίζει το **χρώμα φόντου για το container των επιπέδων κριτηρίων**, τύπου **Brush**.

### Σύνταξη:

#### VB:

```
Public Property GridCriteriaRowsContainerBackground As Brush
```

**Τύπος:** **System.Windows.Media.Brush**

Προσδιορίζουμε ένα αντικείμενο Brush για τον χρωματισμό του φόντου του container των επιπέδων κριτηρίων. Η default τιμή είναι Brushes.Transparent.

### Dependency Property Information:

**Identifier field:** GridCriteriaRowsContainerBackgroundProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το **χρώμα φόντου του container των επιπέδων κριτηρίων** σε μία **απόχρωση του ανοιχτού πράσινου (Light Green)**.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl" Width="Auto" Margin="16"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
    GridCriteriaRowsContainerBackground="#8BC34A"
    ... />
```

#### VB:

```
criteriaControl.GridSortRowsContainerBackground = _
    New SolidColorBrush(ColorConverter.ConvertFromString("#8BC34A"))
```

	Πεδίο	Σύγκριση	Τιμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΔΙΑ
<input type="button" value="και"/>	ΤΑΞΗ	είναι ίσο με	A

**Σημείωση:** Ένα καλό site για τιμές χρωμάτων είναι το <https://www.materialui.co/colors>

## RowHeaderStyle

Καθορίζει το **στυλ του Row Header** που εμφανίζεται σε κάθε επίπεδο (γραμμή) κριτηρίου (εκτός του επιλεγμένου) στο αριστερό μέρος, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property RowHeaderStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα **Style** που θα εφαρμοστεί στο **Row Header** σε κάθε επίπεδο κριτηρίων (εκτός του επιλεγμένου, για το οποίο υπάρχει αντίστοιχη ιδιότητα). Το default style είναι αυτό ενός **TextBlock** element.

### Dependency Property Information:

**Identifier field:** `RowHeaderStyleProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα **"RowHeaderStyle"** που αφορά το **style** για τα **row headers** των επιπέδων (γραμμών) κριτηρίων πλην του επιλεγμένου. Για το **επιλεγμένο row header** καθορίζουμε ένα **ξεχωριστό resource style** με όνομα **"SelectedRowHeaderStyle"**.

### XAML:

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- για το κανονικό row header χρώμα φόντου ελαφρύ γκρι. -->
    <Style x:Key="RowHeaderStyle" TargetType="{x:Type TextBlock }">

      <Setter Property="Background" Value="#EFEFE9"/>

    </Style>
```

```

<!-- για το επιλεγμένο row header χρώμα φόντου πράσινο. -->
<Style x:Key="SelectedRowHeaderStyle" TargetType="{x:Type TextBlock }">

    <Setter Property="Background" Value="Green"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <z:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        RowHeaderStyle="{StaticResource RowHeaderStyle}"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim rowHeaderStyle As New Style(GetType(TextBlock))
Dim selectedRowHeaderStyle As New Style(GetType(TextBlock))

With rowHeaderStyle

    .Setters.Add(New Setter With {.Property = TextBlock.BackgroundProperty, _
        .Value = "#E9EBE9"})

End With

With selectedRowHeaderStyle

    .Setters.Add(New Setter With {.Property = TextBlock.BackgroundProperty, _
        .Value = Brushes.Green})



End With


With criteriaControl



    .RowHeaderStyle = rowHeaderStyle
    .SelectedRowHeaderStyle = selectedRowHeaderStyle

End With

```

 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τμή
	ΤΑΞΗ	είναι ίσο με	A
ή	ΤΑΞΗ	είναι ίσο με	B
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019 

## RowHeaderWidth

Καθορίζει το πλάτος, σε pixels, του Row Header που εμφανίζεται σε κάθε επίπεδο (γραμμή) κριτηρίων.. Είναι τύπου **Double**.

### Σύνταξη:

#### VB:

```
Public Property RowHeaderWidth As Double
```

#### Τύπος: **System.Double**

Προσδιορίζουμε μία τιμή Double που καθορίζει το πλάτος του row header.  
Η default τιμή είναι 30.

### Dependency Property Information:

**Identifier field:** RowHeaderWidthProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το πλάτος της επικεφαλίδας γραμμής (row header) σε 40 pixels.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl" RowHeaderWidth="40" ... />
```

#### VB:

```
criteriaControl.RowHeaderWidth = 40
```

## SelectedRowHeaderStyle

Καθορίζει το **στυλ του Row Header** που εμφανίζεται για το επιλεγμένο επίπεδο (γραμμή) κριτηρίων στο αριστερό μέρος (με το default γαλάζιο χρώμα). Είναι τύπου **Style**.

**Σύνταξη:**

**VB:**

```
Public Property SelectedRowHeaderStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα **Style** που θα εφαρμοστεί στο Row Header στο επιλεγμένο επίπεδο κριτηρίων.

Το default style είναι αυτό ενός **TextBlock** element με γαλάζιο φόντο.

### Dependency Property Information:

**Identifier field:** SelectedRowHeaderStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style** με όνομα "**RowHeaderStyle**" που αφορά το **style** για τα row headers των επιπέδων (γραμμών) κριτηρίων πλην του επιλεγμένου. Για το **επιλεγμένο row header** καθορίζουμε ένα **ξεχωριστό resource style** με όνομα "**SelectedRowHeaderStyle**".

**XAML:**

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
  WindowStartupLocation="CenterScreen">

  <Window.Resources >

    <!-- για το κανονικό row header χρώμα φόντου ελαφρύ γκρι. -->
    <Style x:Key="RowHeaderStyle" TargetType="{x:Type TextBlock }">

      <Setter Property="Background" Value="#EFEFE9"/>

    </Style>
```

```

<!-- για το επιλεγμένο row header χρώμα φόντου πράσινο. -->
<Style x:Key="SelectedRowHeaderStyle" TargetType="{x:Type TextBlock }">

    <Setter Property="Background" Value="Green"/>

</Style>

...

</Window.Resources>

<Grid>

...

    <z:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        RowHeaderStyle="{StaticResource RowHeaderStyle}"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim rowHeaderStyle As New Style(GetType(TextBlock))
Dim selectedRowHeaderStyle As New Style(GetType(TextBlock))

With rowHeaderStyle

    .Setters.Add(New Setter With {.Property = TextBlock.BackgroundProperty, _
        .Value = "#E9E9E9"})

End With

With selectedRowHeaderStyle

    .Setters.Add(New Setter With {.Property = TextBlock.BackgroundProperty, _
        .Value = Brushes.Green})

End With



With criteriaControl


    .RowHeaderStyle = rowHeaderStyle
    .SelectedRowHeaderStyle = selectedRowHeaderStyle



End With

```



 Προσθήκη επιπέδου  Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τμή
	ΤΑΞΗ	είναι ίσο με	A
ή	ΤΑΞΗ	είναι ίσο με	B
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019 

## UseCommaAsDecimalSeparator

Καθορίζει αν θα χρησιμοποιείται το κόμμα ως διαχωριστή δεκαδικών τιμών. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property UseCommaAsDecimalSeparator As Boolean
```

**Τύπος:** System.Boolean

Προσδιορίζουμε μία τιμή Boolean που θα καθορίζει αν θα χρησιμοποιείται το κόμμα για τις δεκαδικές αριθμητικές τιμές. Η default τιμή είναι "True".

### Dependency Property Information:

**Identifier field:** UseCommaAsDecimalSeparatorProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ότι δεν θα χρησιμοποιείται το κόμμα ως διαχωριστή δεκαδικών τιμών. Συνεπώς, η τελεία θα είναι ο διαχωριστής.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"  
    UseCommaAsDecimalSeparator="False" ... />
```

#### VB:

```
criteriaControl.UseCommaAsDecimalSeparator= False
```

## ValueColumnHeaderText

Καθορίζει το κείμενο (λεκτικό) για τη στήλη των Τιμών, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ValueColumnHeaderText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό της επικεφαλίδας της στήλης των τιμών. Η default τιμή είναι "Τιμή".

### Dependency Property Information:

**Identifier field:** ValueColumnHeaderTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Field"** για την επικεφαλίδα των πεδίων, το λεκτικό **"Operator"** για την επικεφαλίδα της στήλης σύγκρισης και το λεκτικό **"Value"** για την επικεφαλίδα των εισαγόμενων τιμών.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl"  
    FieldsColumnHeaderText="Field"  
    ComparisonColumnHeaderText="Operator"  
    ValueColumnHeaderText="Value" ... />
```

#### VB:

```
With criteriaControl
```

```
    .FieldsColumnHeaderText = "Field"  
    .ComparisonColumnHeaderText = "Operator"  
    .ComparisonColumnHeaderText = "Value"
```

```
End With
```

	Field	Operator	Value	
	ΤΑΞΗ	είναι ίσο με	A	
<input type="checkbox"/>	ή	ΤΑΞΗ	είναι ίσο με	B

## ValueTextBoxErrorStyle

Καθορίζει το **στυλ για το TextBox** σε περίπτωση που ο χρήστης εισάγει μία λανθασμένη αριθμητική τιμή. Είναι τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ValueTextBoxErrorStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο TextBox των τιμών (για αριθμητικούς τύπους). Το default style είναι αυτό ενός TextBox element.

### Dependency Property Information:

**Identifier field:** ValueTextBoxErrorStyleProperty

### Παρατηρήσεις:

Το CriteriaControl κάνει αυτόματα έλεγχο μίας αριθμητικής τιμής. Δηλαδή, ανάλογα με τον τύπο που έχουμε καθορίσει για το συγκεκριμένο πεδίο (π.χ. Single) γίνεται εσωτερικά έλεγχος εγκυρότητας. Αν δεν περάσει τον έλεγχο τότε εφαρμόζεται το "στυλ λάθους" στο TextBox. Φυσικά, στην εφαρμογή μας, πρέπει να έχουμε ορίσει ένα τέτοιο style (π.χ. ως resource style) και να το θέσουμε στην ιδιότητα ValueTextBoxErrorStyle με μία StaticResource ή DynamicResource έκφραση.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα **resource style "λάθους"** με όνομα **"valueTextBoxErrorStyle"** και το θέτουμε στην **ιδιότητα ValueTextBoxErrorStyle** του CriteriaControl. Το στυλ "λάθους" θέτει το χρώμα μελάνης σε κόκκινο. Όταν ο χρήστης εισάγει λανθασμένη τιμή σε ένα αριθμητικό πεδίο, για παράδειγμα τον ΒΑΘΜΟ, τότε το στυλ "λάθους" εφαρμόζεται στο αντίστοιχο TextBox τιμής.

Στον VB κώδικα, μπορούμε να εκμεταλλευτούμε τη **μέθοδο ValidateCriteria** για να ελέγξει αν υπάρχει κάποιο πρόβλημα λανθασμένης αριθμητικής τιμής.

### XAML:

```
<Window x:Class="winCriteria"
    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.Controls.DataControls;assembly=ZeusDataControls"
mc:Ignorable="d"
Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="325.05" Width="573.642"
WindowStartupLocation="CenterScreen">

<Window.Resources >

<!-- To style για τις λανθασμένες αριθμητικές τιμές εισόδου. -->
<Style x:Key="valueTextBoxErrorStyle" TargetType="TextBox">

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

</Style>

...

</Window.Resources>

<Grid>

...

<zeus:CriteriaControl Name="criteriaControl" Margin="16"
    FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
    FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
        IS ACTIVE"
    FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
    SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
    ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
    ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
    ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
    ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
    OKClicked="CriteriaControl_OKClicked"/>

</Grid>

</Window>

```

**VB:**

```

Dim valueTextBoxErrorStyle As New Style(GetType(TextBox))

valueTextBoxErrorStyle.Setters.Add(New Setter
    With {.[Property] = ForegroundProperty, .Value = Brushes.Red})

criteriaControl.ValueTextBoxErrorStyle = valueTextBoxErrorStyle

'Ελέγχει τα αριθμητικού τύπου TextBoxes των κριτηρίων, μήπως δεν υπάρχει έγκυρη τιμή.
Private Sub btnCheckCriteria_Click(sender As Object, e As RoutedEventArgs)

    If Not criteriaControl.ValidateCriteria Then

        MessageBox.Show("Κάποιες τιμές δεν είναι έγκυρες στα κριτήρια")
        criteriaControl.Focus()
    End If
End Sub

```

End If

End Sub

+ Προσθήκη επιπέδου		✖ Διαγραφή επιπέδου	
Πεδίο	Σύγκριση	Τιμή	
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ	
και	μεγαλύτερο από	15,1 φ	
ΒΑΘΜΟΣ			

OK Ακύρωση

## ValueTextBoxStyle

Καθορίζει το **στυλ για το TextBox** που χρησιμοποιείται για εισαγωγή από τον χρήστη μίας αριθμητικής τιμής ή κειμένου. Είναι τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ValueTextBoxStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο TextBox, που χρησιμοποιείται για εισαγωγή αριθμητικής ή αλφαριθμητικής από τον χρήστη. Το default style είναι αυτό ενός TextBox element.

### Dependency Property Information:

**Identifier field:** ValueTextBoxStyleProperty

### Παρατηρήσεις:

Το **TextBox control** εμφανίζεται αυτόματα όταν ο **τύπος του πεδίου είναι αριθμητικός ή String**. Χρησιμοποιώντας την ιδιότητα ValueTextBoxStyle μπορούμε να προσδώσουμε ένα ιδιαίτερο στυλ σε αυτό το TextBox.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"ValueInsertingTextBoxStyle"** και το εφαρμόζουμε στην **ιδιότητα ValueTextBoxStyle**. Το στυλ αυτό ορίζει ότι το χρώμα μελάνης θα είναι μπλε και όταν έχει την εστίαση πληκτρολογίου τότε το χρώμα φόντου να είναι κίτρινο.

**XAML:**

```
<Window x:Class="winCriteria"
  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.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Κριτήρια αναζήτησης (Φιλτράρισμα)" Height="377.963" Width="800.936"
  WindowStartupLocation="CenterScreen">
```

```
<Window.Resources >
```



```

<Style x:Key="ValueInsertingTextBoxStyle" TargetType="{x:Type TextBox }">
    <Setter Property="Foreground" Value="Blue"/>
    <Style.Triggers >
        <Trigger Property="IsKeyboardFocused" Value="True">
            <Setter Property="Background" Value="Yellow"/>
        </Trigger>
    </Style.Triggers>
</Style>
...
</Window.Resources>

<Grid>
    ...
    <zeus:CriteriaControl Name="criteriaControl" Margin="16"
        FieldsNames="ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΤΑΞΗ,ΤΜΗΜΑ,ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ,ΒΑΘΜΟΣ,ΕΝΕΡΓΟΣ;"
        FieldsNamesOriginal ="LASTNAME,FIRSTNAME,CLASS,DIVISION,REGISTRATION DATE,GRADE,
            IS ACTIVE"
        FieldsDataTypes="String,String,String,String,Date,Single,Boolean"
        SelectedRowHeaderStyle="{StaticResource SelectedRowHeaderStyle}"
        ComboBoxFieldsNamesStyle="{StaticResource FieldsNamesComboBoxStyle}"
        ValueTextBoxStyle="{StaticResource ValueInsertingTextBoxStyle}"
        ValueTextBoxErrorStyle="{StaticResource valueTextBoxErrorStyle}"
        DatePickerStyle="{StaticResource dtPickerStyle}"
        DatePickerCalendarStyle="{StaticResource calendarStyle}"
        ButtonOKStyle="{StaticResource criteriaOKButtonStyle}"
        ButtonCancelStyle="{StaticResource criteriaCancelButtonStyle}"
        OKClicked="CriteriaControl_OKClicked"/>
</Grid>
</Window>

```

**VB:**

```

Dim valueInsertingTextBoxStyle As New Style(GetType(TextBox))
valueInsertingTextBoxStyle.Setters.Add(New Setter
    With {.[Property] = ForegroundProperty, .Value = Brushes.Blue})
Dim trigger As New Trigger With {.[Property] = IsKeyboardFocusedProperty, _
    .Value = True}
trigger.Setters.Add(New Setter With {.[Property] = TextBox.BackgroundProperty, _
    .Value = Brushes.Yellow})
valueInsertingTextBoxStyle.Triggers.Add(trigger)

```

criteriaControl.ValueTextBoxStyle = valueInsertingTextBoxStyle

Πεδίο	Σύγκριση	Τιμή
ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ

## Συμβάντα

Όνομα	Περιγραφή
OKClicked	Ενεργοποιείται όταν ο χρήστης πατήσει το button "OK".
CancelClicked	Ενεργοποιείται όταν ο χρήστης πατήσει το button "Ακύρωση".

## OKClicked

Ενεργοποιείται όταν ο χρήστης πατήσει το button "OK".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event OKClicked As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:CriteriaControl OKClicked="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε τοποθετήσει αυτόνομα το CriteriaControl σε ένα παράθυρο (Window). Όταν ο χρήστης πατήσει το button "OK", το περιεχόμενο της ιδιότητας CriteriaStringOutput τοποθετείται σε μία public ιδιότητα ReturnCriteriaString και το παράθυρο κλείνει.

Προφανώς, η ιδιότητα ReturnCriteriaString θα αξιοποιηθεί σε μία SQL select query για να φέρει κάποια δεδομένα από τη Βάση με βάση κάποια κριτήρια.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl" ...  
    OKClicked="CriteriaControl_OKClicked"  
    CancelClicked="CriteriaControl_CancelClicked"/>
```

#### VB:

```
Private Sub CriteriaControl_OKClicked(sender As Object, e As RoutedEventArgs)  
  
    ReturnCriteriaString = criteriaControl.CriteriaStringOutput  
    Me.Close()  
  
End Sub
```

## CancelClicked

Ενεργοποιείται όταν ο χρήστης πατήσει το button "Ακύρωση".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event CancelClicked As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:CriteriaControl CancelClicked="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε τοποθετήσει αυτόνομα το CriteriaControl σε ένα παράθυρο (Window). Όταν ο χρήστης πατήσει το button "Ακύρωση", το παράθυρο κλείνει.

#### XAML:

```
<zeus:CriteriaControl Name="criteriaControl" ...  
    OKClicked="CriteriaControl_OKClicked"  
    CancelClicked="CriteriaControl_CancelClicked"/>
```

#### VB:

```
Private Sub CriteriaControl_CancelClicked(sender As Object, e As RoutedEventArgs)  
    Me.Close()  
End Sub
```

## Μέθοδοι

Όνομα	Περιγραφή
CreateCriteria()	<b>Δημιουργεί το περιεχόμενο των ιδιοτήτων CriteriaStringOutput και CriteriaListOutput.</b>
LoadFromFile(filePath As String)	<b>Φορτώνει το περιεχόμενο του CriteriaControl (τα επίπεδα κριτηρίων) από xml αρχείο.</b>
SaveToFile(filePath As String)	<b>Αποθηκεύει το περιεχόμενο του CriteriaControl (τα επίπεδα κριτηρίων) σε xml αρχείο.</b>
ValidateCriteria	<b>Ελέγχει τα αριθμητικού τύπου TextBoxes αν έχουν ορθές τιμές με βάση τον τύπο τους (π.χ. Short, Integer, Double κλπ).</b>

## CreateCriteria

Δημιουργεί το περιεχόμενο των ιδιοτήτων CriteriaStringOutput και CriteriaListOutput.

**Σύνταξη:**

**VB :**

```
Public Sub CreateCriteria()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η δημιουργία ήταν επιτυχής.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καλούμε την CreateCriteria πριν αποθηκεύσουμε το CriteriaControl στο αρχείο Criteria.xml:

**VB:**

```
If criteriaControl.CreateCriteria Then
```

```
criteriaControl.SaveToFile("Criteria.xml")
```

```
MessageBox.Show("Τα κριτήρια αποθηκεύτηκαν με επιτυχία")
```

```
End If
```

## LoadFromFile

Φορτώνει το περιεχόμενο του CriteriaControl (τα επίπεδα κριτηρίων) από xml αρχείο.

Σύνταξη:

**VB :**

```
Public Sub LoadFromFile(  
    filePath As String  
)
```

Παράμετροι:

*filePath*

Τύπος: String

Το όνομα του xml αρχείου που περιέχει το CriteriaControl .

Επιστρεφόμενη τιμή:

Τύπος: Boolean.

True, αν η φόρτωση ήταν επιτυχής.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, φορτώνουμε τα επίπεδα κριτηρίων, που είναι αποθηκευμένα στο αρχείο Criteria.xml, στο CriteriaControl:

**VB:**

```
Private Sub btnLoadCriteria_Click(sender As Object, e As RoutedEventArgs)  
  
    criteriaControl.LoadFromFile("Criteria.xml")  
  
End Sub
```

Αν το περιεχόμενο του αρχείου Criteria.xml είναι το παρακάτω:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>  
<CriteriaControl>  
  <CriteriaLevels>  
    <CriteriaLevel CoonectorValue="-1" FieldName="ΕΠΩΝΥΜΟ"  
      FieldNameOriginal="LASTNAME" FieldType="String"  
      ComparisonForStringValue="BeginsWith"  
      ComparisonForNumAndDateValue="-1"  
      ComparisonForBooleanValue="-1"  
      ValueText="ΠΑΠ" ValueBoolean="-1"  
      ValueDate="2019-08-02T19:14:26.5862786+03:00" />  
  
    <CriteriaLevel CoonectorValue="AndConnector" FieldName="ΒΑΘΜΟΣ"  
      FieldNameOriginal="GRADE" FieldType="Single"  
      ComparisonForStringValue="-1"  
      ComparisonForNumAndDateValue="GreaterThan"
```



```

ComparisonForBooleanValue="-1"
ValueText="16,5" ValueBoolean="-1"
ValueDate="2019-08-02T19:14:26.5862786+03:00" />

<CriteriaLevel CoonectorValue="AndConnector"
  FieldName="ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ"
  FieldNameOriginal="REGISTRATION DATE" FieldType="Date"
  ComparisonForStringValue="-1"
  ComparisonForNumAndDateValue="GreaterThan"
  ComparisonForBooleanValue="-1"
  ValueText="" ValueBoolean="-1"
  ValueDate="2019-06-01T00:00:00" />

<CriteriaLevel CoonectorValue="AndConnector" FieldName="ΕΝΕΡΓΟΣ;"
  FieldNameOriginal="IS ACTIVE" FieldType="Boolean"
  ComparisonForStringValue="-1"
  ComparisonForNumAndDateValue="-1"
  ComparisonForBooleanValue="Equal"
  ValueText="" ValueBoolean="TrueValue"
  ValueDate="2019-08-02T19:14:26.5882997+03:00" />
</CriteriaLevels>
</CriteriaControl>

```

τότε η φόρτωσή του στο **CriteriaControl** θα το διαμορφώσει όπως φαίνεται στην παρακάτω εικόνα:

+ Προσθήκη επιπέδου
✖ Διαγραφή επιπέδου

	Πεδίο	Σύγκριση	Τιμή
	ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
και	ΒΑΘΜΟΣ	μεγαλύτερο από	16,5
και	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019 <input type="text" value="15"/>
και	ΕΝΕΡΓΟΣ;	είναι ίσο με	Ναι

OK
Ακύρωση

Αποθήκευση κριτηρίων
Φόρτωση κριτηρίων
Έλεγχος κριτηρίων

## SaveToFile

Αποθηκεύει το περιεχόμενο του **CriteriaControl** (τα επίπεδα κριτηρίων) σε xml αρχείο.

**Σύνταξη:**

**VB :**

```
Public Sub SaveToFile(  
    filePath As String  
)
```

Παράμετροι:

*filePath*

Τύπος: `String`

Το όνομα του xml αρχείου στο οποίο θα γίνει η αποθήκευση .

*Επιστρεφόμενη τιμή:*

Τύπος: `Boolean`.

`True`, αν η αποθήκευση ήταν επιτυχής.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, αποθηκεύουμε τα επίπεδα κριτηρίων του **CriteriaControl**, στο αρχείο `Criteria.xml`:

**VB:**

```
Private Sub btnSaveCriteria_Click(sender As Object, e As RoutedEventArgs)  
  
    If criteriaControl.CreateCriteria Then  
  
        criteriaControl.SaveToFile("Criteria.xml")  
        MessageBox.Show("Τα κριτήρια αποθηκεύτηκαν με επιτυχία")  
  
    End If  
  
End Sub
```

Αν το **CriteriaControl** έχει διαμορφωθεί όπως στην παρακάτω εικόνα:

		Πεδίο	Σύγκριση	Τιμή
		ΕΠΩΝΥΜΟ	αρχίζει από	ΠΑΠ
και		ΒΑΘΜΟΣ	μεγαλύτερο από	16,5
και		ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	μεγαλύτερο από	1/6/2019 <input type="text" value="15"/>
και		ΕΝΕΡΓΟΣ;	είναι ίσο με	Ναι

τότε η **αποθήκευση στο αρχείο Criteria.xml** θα δημιουργήσει το xml περιεχόμενο των επιπέδων κριτηρίων:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<CriteriaControl>
  <CriteriaLevels>
    <CriteriaLevel CoonectorValue="-1" FieldName="ΕΠΩΝΥΜΟ"
      FieldNameOriginal="LASTNAME" FieldType="String"
      ComparisonForStringValue="BeginsWith"
      ComparisonForNumAndDateValue="-1"
      ComparisonForBooleanValue="-1"
      ValueText="ΠΑΠ" ValueBoolean="-1"
      ValueDate="2019-08-02T19:14:26.5862786+03:00" />

    <CriteriaLevel CoonectorValue="AndConnector" FieldName="ΒΑΘΜΟΣ"
      FieldNameOriginal="GRADE" FieldType="Single"
      ComparisonForStringValue="-1"
      ComparisonForNumAndDateValue="GreaterThan"
      ComparisonForBooleanValue="-1"
      ValueText="16,5" ValueBoolean="-1"
      ValueDate="2019-08-02T19:14:26.5862786+03:00" />

    <CriteriaLevel CoonectorValue="AndConnector"
      FieldName="ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ"
      FieldNameOriginal="REGISTRATION DATE" FieldType="Date"
      ComparisonForStringValue="-1"
      ComparisonForNumAndDateValue="GreaterThan"
      ComparisonForBooleanValue="-1"
      ValueText="" ValueBoolean="-1"
      ValueDate="2019-06-01T00:00:00" />

    <CriteriaLevel CoonectorValue="AndConnector" FieldName="ΕΝΕΡΓΟΣ;"
      FieldNameOriginal="IS ACTIVE" FieldType="Boolean"
      ComparisonForStringValue="-1"
      ComparisonForNumAndDateValue="-1"
      ComparisonForBooleanValue="Equal"
      ValueText="" ValueBoolean="TrueValue"
      ValueDate="2019-08-02T19:14:26.5882997+03:00" />
  </CriteriaLevels>
</CriteriaControl>
```

```
</CriteriaLevels>  
</CriteriaControl>
```

## ValidateCriteria

Ελέγχει τα TextBoxes που έχουν αριθμούς με βάση τον αριθμητικό τύπο τους (Byte, Short, Integer κλπ). Με άλλα λόγια, κάνει έλεγχο εγκυρότητας των αριθμητικών τιμών.

**Σύνταξη:**

**VB :**

```
Public Sub ValidateCriteria()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν ο έλεγχος δεν βρήκε προβλήματα εγκυρότητας.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καλούμε την ValidateCriteria σε έναν button event handler:

**VB:**

```
Private Sub btnCheckCriteria_Click(sender As Object, e As RoutedEventArgs)
```

```
    If Not criteriaControl.ValidateCriteria Then
```

```
        MessageBox.Show("Κάποιες αριθμητικές τιμές δεν είναι έγκυρες")
```

```
    End If
```

```
End Sub
```

- Να σημειώσουμε ότι, αν έχουμε θέσει κάποιο error style στην ιδιότητα ValueTextBoxErrorStyle, θα εφαρμοστεί στο αντίστοιχο TextBox που έχει λανθασμένη τιμή.

## Styles και Templates

- Parts και States
- To default Style και ControlTemplate

## Parts και States

Το default `ControlTemplate` περιλαμβάνει κάποια **part names** και **visual states**. Μπορείτε να τροποποιήσετε το default `ControlTemplate` ώστε να δώσετε στο control μία μοναδική εμφάνιση.

Ο παρακάτω πίνακας εμφανίζει τα **part names** του **CriteriaControl**:

Part	Τύπος	Περιγραφή
PART_AddLevel	Button	Το button που προσθέτει ένα επίπεδο κριτηρίων("Προσθήκη επιπέδου").
PART_RemoveLevel	Button	Το button που αφαιρεί ένα επίπεδο κριτηρίων("Διαγραφή επιπέδου").
PART_OK	Button	Το button επιβεβαίωσης ("OK").
PART_Cancel	Button	Το button ακύρωσης ("Ακύρωση").
PART_CriteriaRowsContainer	Grid	Το container των επιπέδων (γραμμών) κριτηρίων.

Ο παρακάτω πίνακας εμφανίζει τα **visual states** του **CriteriaControl**:

VisualState	VisualStateGroup	Περιγραφή
Normal	CommonStates	Το default state.
MouseOver	CommonStates	Το state όταν ο δείκτης του ποντικιού είναι πάνω από το control.
Disabled	CommonStates	Το state όταν το control είναι disabled.
Focused	FocusStates	Το state όταν το control έχει το focus.
Unfocused	FocusStates	Το state όταν το control δεν έχει το focus.

*Το default `ControlTemplate` έχει καθορισμένο μόνο το `Disabled` state. Μπορείτε να δημιουργήσετε ένα custom `ControlTemplate` για να το αλλάξετε ή/και για να καθορίσετε τα υπόλοιπα.*

## To default Style και ControlTemplate

Ο XAML κώδικας για το **default Style** και **ControlTemplate** φαίνεται παρακάτω. Μπορείτε να βασιστείτε σε αυτόν για να δημιουργήσετε μία μικρή ή μεγάλη παραλλαγή του δικού σας custom Style και ControlTemplate:

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.DataControls">

<Style x:Key="CriteriaRowHeaderDefaultStyle" TargetType="{x:Type TextBlock }">
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Background" Value="White"/>
    <Setter Property="FontWeight" Value="Normal"/>
</Style>

<Style x:Key="SelectedCriteriaHeaderRowDefaultStyle" TargetType="{x:Type TextBlock}">
    <Setter Property="Background" Value="#80D8FF"/>
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="FontWeight" Value="Normal"/>
</Style>

<Style TargetType="{x:Type z:CriteriaControl}">
    <Setter Property="RowHeaderStyle"
        Value="{StaticResource CriteriaRowHeaderDefaultStyle }" />
    <Setter Property="SelectedRowHeaderStyle"
        Value="{StaticResource SelectedCriteriaHeaderRowDefaultStyle }" />
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type z:CriteriaControl}">
                <ControlTemplate.Resources >
                    <z:DoubleToGridLengthConverter x:Key="doubleToGridLengthConverter"/>
                </ControlTemplate.Resources>

                <!-- Root element -->
                <Border Background="{TemplateBinding Background}"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}"
                    >

                    <!-- Visual States -->
                    <VisualStateManager.VisualStateGroups >

                        <VisualStateGroup Name="CommonStates">

                            <VisualStateGroup.Transitions >
                                <VisualTransition GeneratedDuration="0:0:0.2">
                                    </VisualTransition>
                                </VisualStateGroup.Transitions>

```



```

        <VisualState Name="Normal"/>
        <VisualState Name="MouseOver" />
        <VisualState Name="Disabled">
            <Storyboard >
                <ColorAnimationUsingKeyFrames
Storyboard.TargetName="borderButtonsLevelContainerBackground"
                Storyboard.TargetProperty ="Background.Color" >
                    <DiscreteColorKeyFrame KeyTime="0"
Value="{StaticResource {x:Static SystemColors.InactiveBorderColorKey }}" />
                </ColorAnimationUsingKeyFrames>
                <ColorAnimationUsingKeyFrames
Storyboard.TargetName="borderButtonsOKCancelContainerBackground"
                Storyboard.TargetProperty ="Background.Color" >
                    <DiscreteColorKeyFrame KeyTime="0"
Value="{StaticResource {x:Static SystemColors.InactiveBorderColorKey }}" />
                </ColorAnimationUsingKeyFrames>
            </Storyboard>
        </VisualState>
    </VisualStateGroup>

    <VisualStateGroup Name="FocusStates">

        <VisualStateGroup.Transitions >
            <VisualTransition GeneratedDuration="0:0:0.2">
            </VisualTransition>
        </VisualStateGroup.Transitions>

        <VisualState Name="Focused" />
        <VisualState Name="Unfocused"/>

    </VisualStateGroup>

</VisualStateManager.VisualStateGroups>

<!-- Content -->
<DockPanel IsEnabled="{TemplateBinding IsEnabled}"
Background="Transparent"
Margin="{TemplateBinding Padding}"
LastChildFill="True" >

    <!-- StackPanel container για τα AddLevel/Remove level buttons-->
    <Border Name="borderButtonsLevelContainerBackground"
DockPanel.Dock="Top"
Background="{TemplateBinding ButtonsLevelContainerBackground}" >

        <StackPanel Name="stkLevelButtons"
Orientation="Horizontal" Margin="5" >

            <Button Name="PART_AddLevel" Padding="5"
Style="{TemplateBinding ButtonAddLevelStyle}" >
                <Button.Content>
                    <StackPanel Orientation="Horizontal" >
                        <Image
Source="/ZeusDataControls;component/Images/Add_32x32.png"
Width="16" Height="16" />
                        <TextBlock
Text="{TemplateBinding ButtonAddLevelText}"
VerticalAlignment="Center"
Margin="5,0,0,0"/>
                    </StackPanel>
            </Button>
        </StackPanel>
    </Border>

```

```
        </Button.Content>
    </Button>

    <Button Name="PART_RemoveLevel" Margin="5,0,0,0"
        Padding="5"
        Style="{TemplateBinding ButtonRemoveLevelStyle}">
    <Button.Content>
        <StackPanel Orientation="Horizontal" >
            <Image
Source="/ZeusDataControls;component/Images/Delete_32x32.png"
Width="16" Height="16" />
            <TextBlock
                Text="{TemplateBinding ButtonRemoveLevelText}"
                VerticalAlignment="Center"
                Margin="5,0,0,0"/>
        </StackPanel>
    </Button.Content>

</Button>

</StackPanel>
</Border>

<!-- StackPanel container για τα OK/Cancel buttons-->
<Border Name="borderButtonsOKCancelContainerBackground"
    DockPanel.Dock="Bottom"
Background="{TemplateBinding ButtonsOKCancelContainerBackground }" >

    <StackPanel Name="stkConfirmationButtons"
        Orientation="Horizontal" Margin="5"
        HorizontalAlignment="Right" >

        <Button Name="PART_OK"
            Content="{TemplateBinding ButtonOKText}"
            Padding="5"
            Width="{TemplateBinding ButtonsOKCancelWidth }"
            Style="{TemplateBinding ButtonOKStyle}"/>

        <Button Name="PART_Cancel"
            Content="{TemplateBinding ButtonCancelText}"
            Padding="5" Margin="5,0,0,0"
            Width="{TemplateBinding ButtonsOKCancelWidth }"
            Style="{TemplateBinding ButtonCancelStyle}"/>

    </StackPanel>

</Border>

<Border BorderBrush="Black" BorderThickness="1"
    Margin="0,5,0,5">

    <ScrollViewer Margin="5" >

        <Grid Name="PART_CriteriaRowsContainer"
Background="{TemplateBinding GridCriteriaRowsContainerBackground }">

            <Grid.RowDefinitions >
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>

            <Grid.ColumnDefinitions >
```

```

        <ColumnDefinition
            Width="{TemplateBinding RowHeaderWidth,
Converter={StaticResource doubleToGridLengthConverter}}" />
        <ColumnDefinition
            Width="{TemplateBinding ComboBoxConnectorWidth,
Converter={StaticResource doubleToGridLengthConverter}}"/>
        <ColumnDefinition Width="1.5*"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>

    <!--Headers-->
    <TextBlock
        Text="{TemplateBinding FieldsColumnHeaderText}"
        Grid.Column="2" Margin="8,0,0,2"/>

    <TextBlock
        Text="{TemplateBinding ComparisonColumnHeaderText}"
        Grid.Column="3" Margin="8,0,0,2"/>

    <TextBlock
        Text="{TemplateBinding ValueColumnHeaderText}"
        Grid.Column="4" Margin="8,0,0,2"/>

    <!-- 1st Criteria row, ready for user selection.
Initially we display a textbox for the value in 5th column.-->
    <TextBlock Grid.Row="1" Grid.Column="0" />

    <ComboBox Grid.Row="1" Grid.Column="1"
        Margin="5,0,0,0"
        Style="{TemplateBinding ComboBoxConnectorStyle }"
        Visibility="Collapsed" />

    <ComboBox Grid.Row="1" Grid.Column="2"
        Margin="5,0,0,0"
        Style="{TemplateBinding ComboBoxFieldsNamesStyle }"/>

    <ComboBox Grid.Row="1" Grid.Column="3"
        Margin="5,0,0,0"
        Style="{TemplateBinding ComboBoxComparisonStyle }"/>

    <TextBox Grid.Row="1" Grid.Column="4"
        Margin="5,0,5,0"
        Style="{TemplateBinding ValueTextBoxStyle }"
        HorizontalAlignment="Stretch" />

    <ComboBox Grid.Row="1" Grid.Column="4"
        Margin="5,0,5,0"
        Style="{TemplateBinding ComboBoxBooleanValueStyle }"
        HorizontalAlignment="Stretch"
        Visibility="Collapsed" />

    <DatePicker Grid.Row="1" Grid.Column="4"
        Margin="5,0,5,0"
        Style="{TemplateBinding DatePickerStyle}"
        CalendarStyle="{TemplateBinding DatePickerCalendarStyle }"
        HorizontalAlignment="Stretch"
        Visibility="Collapsed"/>

    </Grid>

</ScrollViewer>

```

```
        </Border>
    </DockPanel>
</Border>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
```

## Κλάσεις

Το CriteriaControl χρησιμοποιεί τις παρακάτω **κλάσεις** για τη λειτουργία του:

Όνομα	Περιγραφή
CriteriaRowInfo	Περιλαμβάνει <b>πληροφορίες για ένα επίπεδο (γραμμή) κριτηρίων.</b>

## CriteriaRowInfo

Περιλαμβάνει πληροφορίες για ένα επίπεδο (γραμμή) κριτηρίων.

**Σύνταξη:**

**VB:**

```
Public Class CriteriaRowInfo
```

**Μέλη:**

Ιδιότητα	Τύπος	Περιγραφή
ComparisonForBooleanValue	CriteriaControl.ConnectorEnum	Ο τελεστής σύγκρισης για πεδίο τύπου Boolean.
ComparisonForNumAndDateValue	CriteriaControl.ComparisonNumAndDateOperatorsEnum	Ο τελεστής σύγκρισης για αριθμητικό και ημερομηνιακό τύπο πεδίου.
ComparisonForStringValue	CriteriaControl.ComparisonStringOperatorsEnum	Ο τελεστής σύγκρισης για πεδίο τύπου String.
ConnectorValue	CriteriaControl.ConnectorEnum	Ο συνδέτης του κριτηρίου (and/or).
FieldName	String	Το display όνομα του πεδίου κριτηρίου.

FieldNameOriginal	String	Το πραγματικό όνομα του πεδίου κριτηρίου, όπως έχει οριστεί στην Βάση.
FieldType	String	Ο τύπος δεδομένων του πεδίου κριτηρίου (String, Boolean, Date/DateTime, Integer κλπ).
ValueBoolean	CriteriaControl.BooleanValuesEnum	Η τιμή Boolean.
ValueDate	Nullable(Of Date)	Η τιμή DateTime.
ValueText	String	Η τιμή String (για String, και αριθμητικό τύπο).

- Να σημειώσουμε ότι, οι πληροφορίες αυτές χρησιμοποιούνται για την αποθήκευση/φόρτωση ενός *CriteriaControl* σε/από ένα *xml* αρχείο.



## DataNavigator

Ένα **Control** που παρέχει τη δυνατότητα στον χρήστη να **πλοηγηθεί σε ένα σύνολο εγγραφών/αντικειμένων**. Προαιρετικά, μπορούμε να δώσουμε επιπλέον τη **δυνατότητα στον χρήστη να προσθέσει/διορθώσει/αποθηκεύσει/διαγράψει μία εγγραφή/αντικείμενο**.

Πρόκειται, για μία επεκταμένη περίπτωση του παλιού καλού Data control της VB6.

 <p><b>Μόνο πλοήγηση (Simple).</b> Γαλάζιο φόντο. Default styling στα buttons.</p>	 <p><b>Πλοήγηση και διαχείριση (Full).</b> Γαλάζιο φόντο. Default styling στα buttons.</p>
 <p><b>Μόνο πλοήγηση (Simple).</b> Γαλάζιο φόντο. Με styling στα buttons.</p>	 <p><b>Πλοήγηση και διαχείριση (Full).</b> Γαλάζιο φόντο. Με styling στα buttons.</p>

### Σύνταξη:

#### VB:

```
<TemplateVisualState(Name="Normal", GroupName="CommonStates"),
TemplateVisualState(Name="MouseOver", GroupName="CommonStates"),
TemplateVisualState(Name="Disabled", GroupName="CommonStates"),
TemplateVisualState(Name="Focused", GroupName="FocusStates"),
TemplateVisualState(Name="Unfocused", GroupName="FocusStates"),
TemplatePart(Name="PART_First", Type:=GetType(Button)),
TemplatePart(Name="PART_Previous", Type:=GetType(Button)),
TemplatePart(Name="PART_Next", Type:=GetType(Button)),
TemplatePart(Name="PART_Last", Type:=GetType(Button)),
TemplatePart(Name="PART_InfoText", Type:=GetType(Label)),
TemplatePart(Name="PART_AddNew", Type:=GetType(Button)),
```



```

TemplatePart(Name:="PART_Edit", Type:=GetType(Button)),
TemplatePart(Name:="PART_Commit", Type:=GetType(Button)),
TemplatePart(Name:="PART_Remove", Type:=GetType(Button))>
Public Class DataNavigator
    Inherits Control

```

## XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:DataNavigator ... />
```

## Παρατηρήσεις:

Πολλές φορές έχουμε την ανάγκη να παρέχουμε στον χρήστη μία cursor-mode πλοήγηση στα δεδομένα μίας συλλογής. Παλαιότερα, επί εποχής VB6, η εργαλειοθήκη του Visual Studio παρείχε το εξαιρετικό Data control. Με την έλευση του .NET ένα τέτοιο control εξαφανίστηκε! Συνεπώς, έπρεπε να καταφύγουμε σε controls τρίτων ή α κατασκευάσουμε εμείς ένα παρόμοιο. Ένα τέτοιο control έχω κατασκευάσει για τις Windows Forms και τώρα παρουσιάζω ένα καλύτερο και πιο ευέλικτο.

**Όλα τα γραφικά στοιχεία του control καθώς και τα λεκτικά μπορούν να παραμετροποιηθούν μέσω των αντίστοιχων ιδιοτήτων.** Για παράδειγμα, αν δεν μας ικανοποιεί το style των buttons μπορούμε να ορίσουμε το δικό μας (ως resource style) και να το θέσουμε στην αντίστοιχη ιδιότητα. Ή να ρυθμίσουμε με ιδιότητες-διακόπτες αν θέλουμε να εμφανίζονται καθένα από τα buttons διαχείρισης (AddNew/Edit/Commit/Remove). Ή αν επιθυμούμε να εμφανίζουμε τα tooltips των buttons καθώς και το λεκτικό των tooltip αυτών. Όπως, επίσης και τί είδους πληροφορία θέλουμε να εμφανίζεται στο πληροφοριακό label (μεταξύ των button πλοήγησης).

Η **βασική ιδιότητα του DataNavigator** είναι η **CView**, τύπου **CollectionView**. Δηλαδή, δέχεται μία συλλογή δεδομένων, μέσα στην οποία θα γίνει πλοήγηση ή/και διαχείριση (προσθήκη/διόρθωση/διαγραφή). Οι sub-classes αυτής είναι δύο: **ListCollectionView** όταν η συλλογή υλοποιεί το **IList** ή **ICollection** interface (για παράδειγμα μία **ObservableCollection**) και **BindingListCollectionView** όταν τα δεδομένα προέρχονται από **ADO.NET** objects (**DataTable**, **DataRowView**). Συνεπώς, σε όποια πηγή έχουμε δεδομένα θα πρέπει να τα μεταφέρουμε σε ένα αντικείμενο μίας από τις παραπάνω sub-classes, ώστε να γίνει η πλοήγηση/διαχείριση και να το θέσουμε στην ιδιότητα **CView**.

## Παράδειγμα:

Στο παράδειγμα που ακολουθεί, πλοηγούμαστε στους μαθητές ενός Σχολείου καρτέλα-καρτέλα (cursor-mode). Η κάθε καρτέλα περιέχει πληροφορίες για τον μαθητή όπως

Όνομα, Επώνυμο, Ημερομηνία εγγραφής, Βαθμός (μέσος ετήσιος), Τάξη, Τμήμα κι αν είναι Ενεργός (δεν έχει διακόψει).

Η στρατηγική που θα ακολουθήσουμε είναι να διαμορφώσουμε δύο ομάδες controls. Μία όταν είναι καρτέλα είναι σε *read-only* και μία όταν θα είναι *add/edit*. Για παράδειγμα, όταν η καρτέλα είναι σε *read-only mode*, όλες οι πληροφορίες εμφανίζονται σε *TextBlocks*. Όταν είναι σε *add/edit mode*, ενεργοποιούνται τα *input controls*. Μερικά, είναι *TextBoxes*, άλλα είναι *ComboBoxes*, ένα *DatePicker* κι ένα *CheckBox*. Θεωρώ, ότι αυτή η στρατηγική ταιριάζει απόλυτα στο συνοδευτικό *DataNavigator*. Το σημείο-κλειδί είναι η ιδιότητα *IsAddingNewOrIsEditing*, η οποία καθορίζει την ορατότητα των δύο ομάδων controls (*read-only* και *add/edit*). Επίσης, έχουμε προσαρμόσει το *style* των buttons με ένα ιδιαίτερο *resource style* με όνομα *NavigatorButtonsStyle*.

```
<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:p="clr-namespace:DataControls_Sample_Project"
  xmlns:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="DataControls Sample Project" Height="744.412" Width="988.183"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded">

<Window.Resources >

  <p:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"/>
  <p:BooleanReverseToVisibilityConverter
    x:Key="booleanReverseToVisibilityConverter"/>
  <p:BooleanReverseConverter x:Key="booleanReverseConverter"/>
  <p:BooleanToStringConverter x:Key="booleanToStringConverter"/>

</Window.Resources>

<!-- = ===== -->
<!-- DataNavigator-->
<!-- = ===== -->

<Border Name="brdStudentRecord" Grid.Row="2" BorderBrush="Blue"
  BorderThickness="1" CornerRadius="7" Margin="0,7,0,0" Padding="7"
  Background="#F0F4C3">

  <Grid Name="grdDetails" >

    <Grid.Resources >

      <!-- Το ControlTemplate για τα ValidationRules στα αντίστοιχα πεδία. Εδώ,
      μόνο στον Βαθμό (Grade) -->
      <ControlTemplate x:Key="validationErrorTemplate">

        <StackPanel Orientation="Horizontal">

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

          <TextBlock Grid.Column="1" Foreground="Red" FontSize="16">
```

```

        Text="!" VerticalAlignment="Center"
        Margin="5,0,0,0">
    <TextBlock.ToolTip >
        <TextBlock Text="{Binding Path=/ErrorContent}"/>
    </TextBlock.ToolTip>
</TextBlock>

</StackPanel>

</ControlTemplate>

<!-- Style για τα Navigation buttons.-->
<Style x:Key="NavigatorButtonsStyle" TargetType="{x: Type Button}">

    <Setter Property="Padding" Value="5"/>
    <Setter Property="Template" >
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Border Name="border"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}"
                    CornerRadius="5"
                    Background="{TemplateBinding Background}"
                    Padding="{TemplateBinding Padding}">
                    <ContentPresenter VerticalAlignment="Center"
                        HorizontalAlignment="Center" />
                </Border>
            </ControlTemplate>
        </Setter.Value>
    </Setter>

    <Style.Triggers >

        <Trigger Property="IsEnabled" Value="False">
            <Setter Property="Background" Value="#BDBDBD" />
            <Setter Property="Opacity" Value="0.5" />
        </Trigger>
        <Trigger Property="IsEnabled" Value="True">
            <Setter Property="Background" Value="#EEEEEE" />
        </Trigger>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="Yellow" />
        </Trigger>
        <Trigger Property="IsPressed" Value="True">
            <Setter Property="Background" Value="LightGreen" />
        </Trigger>

    </Style.Triggers>

</Style>

<!-- Style για το InfoText του DataNavigator. -->
<Style x:Key="InfoStyle" TargetType="{x:Type Label}">

    <Setter Property="FontSize" Value="16"/>
    <Setter Property="Background" Value="WhiteSmoke" />
    <Setter Property="Margin" Value="2,0"/>
    <Setter Property="Padding" Value="5"/>

</Style>

```

```
<!-- Style για τα Labels της details φόρμας. -->
<Style x:Key="RowLabelStyle" TargetType="{x:Type TextBlock}">
  <Setter Property="FontSize" Value="14"/>
  <Setter Property="HorizontalAlignment" Value="Right"/>
  <Setter Property="VerticalAlignment" Value="Center" />
  <Setter Property="Margin" Value="0,0,5,0"/>
</Style>

<!-- Style για τα TextBoxes της details φόρμας, όταν είναι σε add/edit mode.
-->
<Style x:Key="RowTextBoxStyle" TargetType="{x:Type TextBox}">
  <Setter Property="FontSize" Value="14"/>
  <Setter Property="FontWeight" Value="Bold" />
  <Setter Property="HorizontalAlignment" Value="Left"/>
  <Setter Property="VerticalAlignment" Value="Center" />
</Style>

<!-- Style για τα TextBoxes της details φόρμας με έλεγχο λαθών
(για το πεδίο Grade) -->
<Style x:Key="RowTextBoxWithErrorsStyle" TargetType="{x:Type TextBox}"
  BasedOn="{StaticResource RowTextBoxStyle }">
  <Style.Triggers>
    <Trigger Property="Validation.HasError" Value="True">
      <Setter Property="ToolTip"
        Value="{Binding RelativeSource={x:Static RelativeSource.Self},
          Path=(Validation.Errors)[0].ErrorContent}"/>
      <Setter Property="Foreground" Value="Red"/>
      <Setter Property="BorderBrush" Value="Red"/>
      <Setter Property="BorderThickness" Value="2"/>
    </Trigger>
  </Style.Triggers>
</Style>

<!-- Style για τα TextBlocks όταν η details φόρμα είναι σε read-mode.-->
<Style x:Key="RowTextBlockStyle" TargetType="{x:Type TextBlock }">
  <Setter Property="FontSize" Value="14"/>
  <Setter Property="Margin" Value="5,0,0,0"/>
  <Setter Property="FontWeight" Value="Bold" />
  <Setter Property="HorizontalAlignment" Value="Left"/>
  <Setter Property="VerticalAlignment" Value="Center" />
</Style>

</Grid.Resources>

<Grid.ColumnDefinitions >

  <!-- Πρώτη στήλη τα labels και Δεύτερη στήλη τα input controls.-->
  <ColumnDefinition Width="Auto"/>
  <ColumnDefinition Width="*/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions >

  <!-- Ο τίτλος με το DataNavigator control.-->
  <RowDefinition Height="Auto"/>
  <!-- Τα πεδία για την εγγραφή του μαθητή.-->
  <RowDefinition Height="30"/>
  <RowDefinition Height="30"/>
  <RowDefinition Height="25"/>
  <RowDefinition Height="30"/>
  <RowDefinition Height="30"/>
</Grid.RowDefinitions >
```

```

        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>

    </Grid.RowDefinitions>

    <StackPanel Orientation="Horizontal" Margin="0,10,0,10" Grid.ColumnSpan="2">

        <TextBlock Text="Καρτέλα μαθητή" FontSize="26"
            Foreground="Blue" VerticalAlignment="Center"/>

        <z:DataNavigator Name="DatStudents" Padding="10" Background="Cyan"
            Width="420" Height="50" Margin="45,0,0,0"
            InfoTextStyle="{StaticResource InfoStyle}"
            ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
            ShowEdit="True"
            ShowToolTipsForButtons="True"
            ShowToolTipForInfo="True"
            ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
            MoveStarted="DatStudents_MoveStarted"
            MoveCompleted="DatStudents_MoveCompleted"
            AddNewStarted="DatStudents_AddNewStarted"
            EditItemStarted="DatStudents_EditItemStarted"
            CommitStarted="DatStudents_CommitStarted"
            CommitCompleted="DatStudents_CommitCompleted"
            RemoveStarted="DatStudents_RemoveStarted">

        </z:DataNavigator>

        <TextBlock Name="tbStatus" FontSize="20" VerticalAlignment="Center"
            Margin="30,0,0,0"/>

    </StackPanel>

    <!-- Λεπτομέρειες εγγραφής. Κάνουμε ένα τρυκ. Όταν το DataNavigator είναι σε normal
    mode εμφανίζονται διαφορετικά controls (συνήθως TextBlocks) ώστε να υποδηλώνει
    display-mode.
    Όταν είναι σε Adding/Editing-mode εμφανίζει input controls (TextBoxes,
    DatePicker, ComboBoxes και CheckBox). Κοιτάξτε την ιδιότητα Visibility!
    Φυσικά, μπορείτε να ακολουθήσετε διαφορετική στρατηγική.-->

    <TextBlock Grid.Row="1" Text="Όνομα:"
        Style="{StaticResource RowLabelStyle }" />
    <TextBlock Grid.Row="1" Grid.Column="1"
        Style="{StaticResource RowTextBlockStyle }"
        HorizontalAlignment="Stretch"
        Text="{Binding FirstName}"
        Visibility="{Binding ElementName=DatStudents,
            Path=IsAddingNewOrIsEditing,
            Converter={StaticResource booleanReverseToVisibilityConverter}}"/>

    <TextBox Name="txtFirstName" Grid.Row="1" Grid.Column="1" MaxLength="50"
        Style="{StaticResource RowTextBoxStyle }"
        HorizontalAlignment="Stretch"
        Text="{Binding FirstName}"
        Visibility="{Binding ElementName=DatStudents,
            Path=IsAddingNewOrIsEditing,
            Converter={StaticResource booleanToVisibilityConverter}}" />

    <TextBlock Grid.Row="2" Text="Επώνυμο:"
        Style="{StaticResource RowLabelStyle }"/>

```

```

<TextBlock Grid.Row="2" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }"
    HorizontalAlignment="Stretch"
    Text="{Binding LastName}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<TextBox Name="txtLastName" Grid.Row="2" Grid.Column="1" MaxLength="50"
    Style="{StaticResource RowTextBoxStyle }"
    HorizontalAlignment="Stretch"
    Text="{Binding LastName}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}" />

<TextBlock Grid.Row="3" Text="Ημερομ. Εγγραφής:"
    Style="{StaticResource RowLabelStyle }" />
<TextBlock Grid.Row="3" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }" Width="150"
    Text="{Binding RegistrationDate, StringFormat={}{0:d}}"
    Language="el"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<DatePicker Name="dtRegistrationDate" Grid.Row="3" Grid.Column="1"
    Width="150" HorizontalAlignment="Left" Language="el"
    FontWeight="Bold" FontSize="14"
    SelectedDate="{Binding RegistrationDate}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}"/>

<TextBlock Grid.Row="4" Text="Βαθμός:"
    Style="{StaticResource RowLabelStyle }"/>
<TextBlock Grid.Row="4" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }" Width="80"
    Text="{Binding Grade}" Language="el"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<TextBox Name="txtGrade" Grid.Row="4" Grid.Column="1" Width="80"
    HorizontalAlignment="Left" Language="el"
    Style="{StaticResource RowTextBoxStyle }"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}"
    Validation.ErrorTemplate="{StaticResource validationErrorTemplate}"

    <Binding Path="Grade" StringFormat="{}{0:N1}">
        <Binding.ValidationRules>
            <p:ValidateSingle ErrorMessage="0 βαθμός δεν είναι έγκυρος" />
        </Binding.ValidationRules>
    </Binding>

</TextBox>

<TextBlock Grid.Row="5" Text="Τάξη:"
    Style="{StaticResource RowLabelStyle }"/>

```

```
<TextBlock Grid.Row="5" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }" Width="100"
    Text="{Binding ClassName}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<ComboBox Name="cboClass" Grid.Row="5" Grid.Column="1" Width="100"
    HorizontalAlignment="Left" VerticalAlignment="Center"
    FontWeight="Bold"
    SelectedValuePath="IDClass"
    SelectedValue="{Binding ClassID}"
    SelectionChanged="CboClass_SelectionChanged"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}" />

<TextBlock Grid.Row="6" Text="Τμήμα:"
    Style="{StaticResource RowLabelStyle }"/>
<TextBlock Grid.Row="6" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }" Width="100"
    Text="{Binding DivisionName}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<ComboBox Name="cboDivision" Grid.Row="6" Grid.Column="1" Width="100"
    HorizontalAlignment="Left" VerticalAlignment="Center"
    FontWeight="Bold"
    SelectedValuePath="IDDivision"
    SelectedValue="{Binding DivisionID}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}" />

<TextBlock Grid.Row="7" Text="Ενεργός:"
    Style="{StaticResource RowLabelStyle }" />
<TextBlock Grid.Row="7" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }" Width="50"
    Text="{Binding IsActive,
        Converter={StaticResource booleanToStringConverter},
        ConverterParameter=Ναι:Όχι}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<CheckBox Name="chkIsActive" Grid.Row="7" Grid.Column="1" Width="100"
    HorizontalAlignment="Left" VerticalAlignment="Center"
    IsChecked="{Binding IsActive}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}" />

</Grid>

</Border>

</Window>
```

Στον VB κώδικα:

```
'Η συλλογή αντικειμένων Students, SchoolClasses, SchoolDivisions.
Private obsStudents As ObservableCollection(Of Student)
Private obsSchoolClasses As ObservableCollection(Of SchoolClass)
Private obsSchoolDivisions As ObservableCollection(Of SchoolDivision)

Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)

    obsStudents = GetStudents()
    obsSchoolClasses = GetSchoolClasses()
    obsSchoolDivisions = GetSchoolDivisions()

    DatStudents.CView = _
    CType(CollectionViewSource.GetDefaultView(obsStudents), ListCollectionView)

    'Η πηγή δεδομένων για τις λίστες cboClass και cboDivision.
    cboClass.ItemsSource = obsSchoolClasses
    cboDivision.ItemsSource = obsSchoolDivisions

    If DatStudents.Count > 0 Then

        DatStudents.MoveToFirst()
        'Να πάρει μόνο τα τμήματα της συγκεκριμένης τάξης.
        cboDivision.ItemsSource =
        GetDivisionsOfAClassFromCollection(CType(DatStudents.CurrentItem, _
                                                Student).ClassID

    End If

    grdDetails.DataContext = DatStudents.CView
    RefreshDatStudentsInfo()

End Sub

Private Sub RefreshDatStudentsInfo()

    With DatStudents

        .InfoText = .CurrentPosition + 1 & " / " & .Count

    End With

End Sub
```

Έχουμε δημιουργήσει κάποια data αντικείμενα όπως Student, SchoolClass και SchoolDivision και κάποιες ObservableCollection που κρατάνε συλλογές από αυτά τα αντικείμενα. Χρησιμοποιούμε κάποιες μεθόδους GetStudents, GetSchoolClasses και GetSchoolDivisions που φέρνουν πληροφορίες από τη Βάση και τις εναποθέτουν στις αντίστοιχες συλλογές. **Βλέπουμε πώς τίθεται η ιδιότητα CView με βάση την συλλογή obsStudents.**

Πιο κάτω, το αντικείμενο CView τίθεται στην ιδιότητα DataContext του Grid που περιέχει τα controls, τα οποία γίνονται binding στις αντίστοιχες ιδιότητες του αντικειμένου Student.



Στις παρακάτω εικόνες βλέπουμε μερικά δείγματα:

	Read -Only mode
	Edit mode

Καρτέλα μαθητή

+ / 21

Προσθήκη

Όνομα:

Επώνυμο:

Ημερομ. Εγγραφής: 2/8/2019

Βαθμός: 0,0

Τάξη: A

Τμήμα: A1

Ενεργός:

AddNew mode

## Ιδιότητες

Όνομα	Περιγραφή
ButtonsWidth	Καθορίζει το πλάτος, σε pixels, των buttons, τύπου Double.
ButtonsStyle	Καθορίζει το στυλ για τα buttons, τύπου Style.
CView	Καθορίζει το αντικείμενο <b>CollectionView</b> (πιο συγκεκριμένα, ένα από τις sub-classes <b>ListCollectionView</b> ή <b>BindingListCollectionView</b> ).  Πάνω σε αυτό θα γίνει η πλοήγηση/διαχείριση των source εγγραφών/αντικειμένων.
IsAddingNew	Καθορίζει αν το control βρίσκεται σε διαδικασία προσθήκης εγγραφής/αντικειμένου. Είναι τύπου Boolean.
IsAddingNewOrIsEditing	Επιστρέφει True, αν το control βρίσκεται σε διαδικασία προσθήκης/διόρθωσης εγγραφής/αντικειμένου. Είναι τύπου Boolean.
IsEditing	Καθορίζει αν το control βρίσκεται σε διαδικασία διόρθωσης εγγραφής/αντικειμένου. Είναι τύπου Boolean.
InfoText	Καθορίζει το κείμενο που θα εμφανίζεται στο πληροφοριακό Label, που βρίσκεται μεταξύ των buttons πλοήγησης. Είναι τύπου String.
InfoTextStyle	Καθορίζει το στυλ για το πληροφοριακό Label, που βρίσκεται μεταξύ των buttons πλοήγησης. Είναι τύπου Style.
ShowAddNew	Καθορίζει αν θα εμφανίζεται το button AddNew (προσθήκης). Είναι τύπου Boolean.
ShowCommit	Καθορίζει αν θα εμφανίζεται το button Commit (επιβεβαίωσης-αποθήκευσης). Είναι τύπου Boolean.

ShowEdit	Καθορίζει αν θα εμφανίζεται το button Edit (διόρθωσης). Είναι τύπου <b>Boolean</b> .
ShowRemove	Καθορίζει αν θα εμφανίζεται το button Remove (διαγραφής). Είναι τύπου <b>Boolean</b> .
ShowToolTipsForButtons	Καθορίζει αν θα εμφανίζονται τα ToolTips για τα buttons. Είναι τύπου <b>Boolean</b> .
ShowToolTipForInfo	Καθορίζει αν θα εμφανίζεται το ToolTip για το πληροφοριακό Label. Είναι τύπου <b>Boolean</b> .
ToolTipTextForAddNew	Καθορίζει το κείμενο για το ToolTip του button AddNew. Είναι τύπου <b>String</b> .
ToolTipTextForCancelEdit	Καθορίζει το κείμενο για το ToolTip του button Remove, όταν αυτό είναι σε κατάσταση διόρθωσης. Είναι τύπου <b>String</b> .  Αφορά την ακύρωση διόρθωσης.
ToolTipTextForCancelNew	Καθορίζει το κείμενο για το ToolTip του button Remove, όταν αυτό είναι σε κατάσταση προσθήκης. Είναι τύπου <b>String</b> .  Αφορά την ακύρωση προσθήκης νέας εγγραφής/αντικειμένου.
ToolTipTextForCommit	Καθορίζει το κείμενο για το ToolTip του button Commit. Είναι τύπου <b>String</b> .
ToolTipTextForEdit	Καθορίζει το κείμενο για το ToolTip του button Edit. Είναι τύπου <b>String</b> .
ToolTipTextForInfo	Καθορίζει το κείμενο για το ToolTip του πληροφοριακού Label. Είναι τύπου <b>String</b> .
ToolTipTextForMoveToFirst	Καθορίζει το κείμενο για το ToolTip του button MoveToFirst. Είναι τύπου <b>String</b> .
ToolTipTextForMoveToLast	Καθορίζει το κείμενο για το ToolTip του button MoveToLast. Είναι τύπου <b>String</b> .

ToolTipTextForMoveToNext	Καθορίζει το κείμενο για το ToolTip του button <b>MoveToNext</b> . Είναι τύπου <b>String</b> .
ToolTipTextForMoveToPrevious	Καθορίζει το κείμενο για το ToolTip του button <b>MoveToPrevious</b> . Είναι τύπου <b>String</b> .
ToolTipTextForRemove	Καθορίζει το κείμενο για το ToolTip του button <b>Remove</b> , όταν αυτό είναι σε κανονική κατάσταση (όχι προσθήκης/διόρθωσης). Είναι τύπου <b>String</b> .

## ButtonsWidth

Καθορίζει το **πλάτος, σε pixels, των buttons**. Είναι τύπου **Double**.

### Σύνταξη:

#### VB:

```
Public Property ButtonsWidth As Double
```

#### Τύπος: System.Double

Προσδιορίζουμε μία τιμή Double που καθορίζει το πλάτος για όλα τα buttons του control. Η default τιμή είναι 30.

### Dependency Property Information:

**Identifier field:** ButtonsWidthProperty

### Παρατηρήσεις:

Ο λόγος που υπάρχει αυτή η ιδιότητα είναι για να καθορίσουμε γρήγορα ένα ομοιόμορφο πλάτος για όλα τα buttons.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το **πλάτος των buttons** σε **40 pixels**.

#### XAML:

```
<zeus:Datanavigator Name="DatStudents" ButtonsWidth="40" ... />
```

#### VB:

```
DatStudents.ButtonsWidth = 40
```

## ButtonsStyle

Καθορίζει το **στυλ για τα buttons**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ButtonsStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί ομοιόμορφα σε όλα τα buttons. Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonsStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"NavigatorButtonsStyle"** και το εφαρμόζουμε στην **ιδιότητα ButtonsStyle** του **DataNavigator**. Τα buttons γίνονται πιο στρογγυλεμένα και θέτουμε κάποια triggers όταν ενεργοποιούνται/απενεργοποιούνται, όταν το ποντίκι ίπταται πάνω τους και όταν πατιούνται.

#### XAML:

```
<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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="DataControls Sample Project" Height="744.412" Width="988.183"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded">

  <Window.Resources >

    <!-- Style για τα Navigation buttons.-->
    <Style x:Key="NavigatorButtonsStyle" TargetType="{x:Type Button}">

      <Setter Property="Padding" Value="5"/>
      <Setter Property="Template" >
        <Setter.Value>
          <ControlTemplate TargetType="Button">
            <Border Name="border"
              BorderBrush="{TemplateBinding BorderBrush}"
              BorderThickness="{TemplateBinding BorderThickness}">
```

```
        CornerRadius="5"
        Background="{TemplateBinding Background}"
        Padding="{TemplateBinding Padding}">
        <ContentPresenter VerticalAlignment="Center"
            HorizontalAlignment="Center" />
    </Border>
</ControlTemplate>
</Setter.Value>
</Setter>

<Style.Triggers >

    <Trigger Property="IsEnabled" Value="False">
        <Setter Property="Background" Value="#BDBDBD" />
        <Setter Property="Opacity" Value="0.5" />
    </Trigger>
    <Trigger Property="IsEnabled" Value="True">
        <Setter Property="Background" Value="#EEEEEE" />
    </Trigger>
    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Background" Value="Yellow" />
    </Trigger>
    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Background" Value="LightGreen" />
    </Trigger>

</Style.Triggers>

</Style>

...

</Window.Resources>

<Grid>

...

    <z:DataNavigator Name="DatStudents" Padding="10" Background="Cyan"
        Width="420" Height="50" Margin="45,0,0,0"
        InfoTextStyle="{StaticResource InfoStyle}"
        ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
        ShowAddNew="True"
        ShowEdit="True"
        ShowCommit="True"
        ShowRemove="True"
        ShowToolTipsForButtons="True"
        ShowToolTipForInfo="True"
        ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
        MoveStarted="DatStudents_MoveStarted"
        MoveCompleted="DatStudents_MoveCompleted"
        AddNewStarted="DatStudents_AddNewStarted"
        EditItemStarted="DatStudents_EditItemStarted"
        CommitStarted="DatStudents_CommitStarted"
        CommitCompleted="DatStudents_CommitCompleted"
        RemoveStarted="DatStudents_RemoveStarted">

</z:DataNavigator>
```



</Grid>

</Window>

Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα:



## CView

Καθορίζει το αντικείμενο **CollectionView** (πιο συγκεκριμένα, ένα από τις sub-classes **ListCollectionView** ή **BindingListCollectionView**), που περιέχει τη συλλογή των εγγραφών/αντικειμένων.

### Σύνταξη:

#### VB:

```
Public Property CView As CollectionView
```

**Τύπος:** **System.Windows.Data.CollectionView**

Προσδιορίζουμε ένα αντικείμενο **CollectionView** που περιέχει τα data για την πλοήγηση/διαχείριση.

### Dependency Property Information:

**Identifier field:** `CViewProperty`

### Παρατηρήσεις:

Το **CollectionView** αντικείμενο, που χρησιμοποιεί εσωτερικά το **DataNavigator**, μπορεί γίνει άμεσα προσβάσιμο στον κώδικα, αν το επιθυμεί ο προγραμματιστής. Σε γενικές γραμμές, όμως, κάτι τέτοιο δεν χρειάζεται αφού το **DataNavigator** αποτελεί wrapper του **CollectionView**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε στον `Window_Loaded` handler, τη φόρτωση των εγγραφών των μαθητών στη συλλογή `obsStudents`. Η μέθοδος `GetStudents()` απλά λαμβάνει τα δεδομένα όλων των μαθητών από τη βάση και τα τοποθετεί στο αντικείμενο `ObservableCollection(Of Student)`. Μόλις γίνει αυτό, τότε δημιουργούμε μία "άποψη" της συλλογής αυτής σε ένα αντικείμενο `ListCollectionView` και το εναποθέτουμε στην ιδιότητα `CView`. Παρακάτω, θέτουμε την ιδιότητα `CView` (δηλαδή, το εκχωρημένο αντικείμενο `ListCollectionView`) ως πηγή δεδομένων στην `DataContext` του `Grid` που περιέχει τα controls της καρτέλας του μαθητή.

Να σημειώσουμε, ότι αν κάναμε χρήση των ADO.NET data objects (`DataTable`, `DataView`) τότε αντί ενός `ListCollectionView` θα χρησιμοποιούσαμε ένα `BindingListCollectionView`.

**VB:**

```
Private obsStudents As ObservableCollection(Of Student)
```

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
```

```
    obsStudents = GetStudents()
```

```
    DatStudents.CView = _
```

```
        CType(CollectionViewSource.GetDefaultView(obsStudents), ListCollectionView)
```

```
    ...
```

```
    grdDetails.DataContext = DatStudents.CView
```

```
End Sub
```

## IsAddingNew

Καθορίζει αν το control βρίσκεται σε διαδικασία προσθήκης εγγραφής/αντικειμένου. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property IsAddingNew As Boolean
```

**Τύπος:** System.Boolean

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν το control είναι σε διαδικασία προσθήκης νέων δεδομένων. Η default τιμή είναι "False".

### Dependency Property Information:

**Identifier field:** IsAddingNewProperty

### Παρατηρήσεις:

Όταν ο χρήστης πατήσει το button AddNew τότε αυτόματα η ιδιότητα IsAddingNew γίνεται True. Αν επιθυμεί ο προγραμματιστής, μπορεί να θέσει την ιδιότητα IsAddingNew χειροκίνητα. Σε κάθε περίπτωση, όλα τα buttons απενεργοποιούνται εκτός των Commit και CancelNew.

Η συνήθης χρήση με την ιδιότητα IsAddingNew είναι να ελέγχουμε αν έχει ξεκινήσει μία διαδικασία προσθήκης.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ότι ξεκινάμε μία προσθήκη νέας εγγραφής/αντικειμένου.


#### XAML:

```
<zeus:DataNavigator Name="DatStudents" IsAddingNew ="True" ... />
```

#### VB:


```
DatStudents.IsAddingNew = True
```

**Μαθητής**



Όνομα:

Επώνυμο:

Ημερομ. Εγγραφής:  

Βαθμός:

Τάξη:  ▾

Τμήμα:  ▾

Ενεργός:

## IsAddingNewOrIsEditing

Επιστρέφει True, αν το control βρίσκεται σε διαδικασία προσθήκης/διόρθωσης εγγραφής/αντικειμένου. Είναι τύπου Boolean.

Σύνταξη:

VB:

```
Public Property IsAddingNewOrIsEditing As Boolean
```

Τύπος: System.Boolean

Προσδιορίζουμε μία τιμή Boolean που θα καθορίζει αν το control είναι σε διαδικασία προσθήκης/διόρθωσης δεδομένων. Η default τιμή είναι "False".

### Dependency Property Information:

Identifier field: IsAddingNewOrIsEditingProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, συνδέουμε την ιδιότητα IsAddingNewOrIsEditingProperty του DataNavigator με την ιδιότητα Visibility δύο controls: ενός TextBlock που απλά εμφανίζει (display) το περιεχόμενο της ιδιότητας FirstName κι ενός TextBox που δέχεται (input) το περιεχόμενο της ίδιας ιδιότητας FirstName. Το πρώτο θα εμφανίζεται όταν η ιδιότητα IsAddingNewOrIsEditingProperty είναι False, και το δεύτερο θα εμφανίζεται όταν η ιδιότητα IsAddingNewOrIsEditingProperty είναι True. Βασικό ρόλο εδώ στις εκφράσεις Binding παίζουν τα ValueConverters.

XAML:

```
<Window.Resources >
    <p:BooleanToVisibilityConverter x:Key="booleanToVisibilityConverter"/>
    <p:BooleanReverseToVisibilityConverter
        x:Key="booleanReverseToVisibilityConverter"/>
</Window.Resources>
...
<zeus:DataNavigator Name="DatStudents" Padding="10" Background="Cyan"
    Width="420" Height="50" Margin="45,0,0,0"
    InfoTextStyle="{StaticResource InfoStyle}"
    ButtonsStyle="{SShowAddNew="True"
    ShowEdit="True"
    ShowCommit="True"
    ShowRemove="True" ShowEdit="True"
    ShowToolTipsForButtons="True"
```

```
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
CommitStarted="DatStudents_CommitStarted"
CommitCompleted="DatStudents_CommitCompleted"
RemoveStarted="DatStudents_RemoveStarted">

</zeus:DataNavigator>

...

<TextBlock Grid.Row="1" Grid.Column="1"
    Style="{StaticResource RowTextBlockStyle }"
    HorizontalAlignment="Stretch"
    Text="{Binding FirstName}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanReverseToVisibilityConverter}}" />

<TextBox Name="txtFirstName" Grid.Row="1" Grid.Column="1" MaxLength="50"
    Style="{StaticResource RowTextBoxStyle }"
    HorizontalAlignment="Stretch"
    Text="{Binding FirstName}"
    Visibility="{Binding ElementName=DatStudents,
        Path=IsAddingNewOrIsEditing,
        Converter={StaticResource booleanToVisibilityConverter}}"
/>
```

## IsEditing

Καθορίζει αν το control βρίσκεται σε διαδικασία διόρθωσης εγγραφής/αντικειμένου. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property IsEditing As Boolean
```

#### Τύπος: **System.Boolean**

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν το control είναι σε διαδικασία προσθήκης νέων δεδομένων. Η default τιμή είναι "False".

### Dependency Property Information:

**Identifier field:** IsEditingProperty

### Παρατηρήσεις:

Όταν ο χρήστης πατήσει το button Edit τότε αυτόματα η ιδιότητα IsEditing γίνεται True. Αν επιθυμεί ο προγραμματιστής, μπορεί να θέσει την ιδιότητα IsEditing χειροκίνητα. Σε κάθε περίπτωση, όλα τα buttons απενεργοποιούνται εκτός των Commit και CancelEdit.

Η συνήθης χρήση με την ιδιότητα IsEditing είναι να ελέγχουμε αν έχει ξεκινήσει μία διαδικασία διόρθωσης.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ότι ξεκινάμε μία διόρθωση του τρέχοντος αντικειμένου.

#### XAML:

```
<zeus:DataNavigator Name="DatStudents" IsEditing ="True" ... />
```

#### VB:

```
DatStudents.IsEditing = True
```



**Μαθητής**

4 / 21

Όνομα: ΕΛΕΑΝΑ

Επώνυμο: ΔΙΑΛΥΝΑ

Ημερομ. Εγγραφής: 4/6/2019 15

Βαθμός: 8,5

Τάξη: B

Τμήμα: B1

Ενεργός:

## InfoText

Καθορίζει το κείμενο που θα εμφανίζεται στο πληροφοριακό Label, που βρίσκεται μεταξύ των buttons πλοήγησης. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property InfoText As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το πληροφοριακό κείμενο για το Label. Η default τιμή είναι "0 / 0".

### Dependency Property Information:

**Identifier field:** InfoTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε την πληροφορία που θα εμφανίζεται όταν ξεκινάει μία διαδικασία προσθήκης (AddNewStarted event handler) και την πληροφορία που θα εμφανίζεται όταν ολοκληρωθεί η μετακίνηση σε μία εγγραφή (MoveCompleted event handler).

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
    Width="420" Height="50" Margin="45,0,0,0"
    InfoTextStyle="{StaticResource InfoStyle}"
    ButtonsStyle="{SShowAddNew="True"
    ShowEdit="True"
    ShowCommit="True"
    ShowRemove="True" ShowEdit="True"
    ShowToolTipsForButtons="True"
    ShowToolTipForInfo="True"
    ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
    MoveStarted="DatStudents_MoveStarted"
    MoveCompleted="DatStudents_MoveCompleted"
    AddNewStarted="DatStudents_AddNewStarted"
    EditItemStarted="DatStudents_EditItemStarted"
    CommitStarted="DatStudents_CommitStarted"
    CommitCompleted="DatStudents_CommitCompleted"
    RemoveStarted="DatStudents_RemoveStarted">

</zeus:DataNavigator>
```

**VB:**

'Όταν ολοκληρωθεί η μετακίνηση.

**Private Sub** DatStudents\_MoveCompleted(sender **As Object**, e **As RoutedEventArgs**)    **With** DatStudents        **.InfoText** = **.CurrentPosition** + 1 & " / " & **.Count**    **End With****End Sub**

'Ξεκίνησε η προσθήκη νέου μαθητή.

**Private Sub** DatStudents\_AddNewStarted(sender **As Object**, e **As RoutedEventArgs**)    **With** DatStudents        **.InfoText** = "+ / " & **.Count** - 1    **End With****End Sub**

## InfoTextStyle

Καθορίζει το **στυλ για το πληροφοριακό Label**, που βρίσκεται μεταξύ των **buttons** πλοήγησης. Είναι τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property InfoTextStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο πληροφοριακό Label. Το default style είναι αυτό ενός Label element.

### Dependency Property Information:

**Identifier field:** InfoTextStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"NavigatorButtonsStyle"** και το εφαρμόζουμε στην **ιδιότητα ButtonsStyle** του **DataNavigator**. Τα buttons γίνονται πιο στρογγυλεμένα και θέτουμε κάποια triggers όταν ενεργοποιούνται/απενεργοποιούνται, όταν το ποντίκι ίπταται πάνω τους και όταν πατιούνται.

#### XAML:

```
<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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="DataControls Sample Project" Height="744.412" Width="988.183"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded">

  <Window.Resources >

    <!-- Style για το InfoText του DataNavigator. -->
    <Style x:Key="InfoStyle" TargetType="{x:Type Label}">

      <Setter Property="FontSize" Value="16"/>
      <Setter Property="Background" Value="#B39DDB" />
      <Setter Property="Foreground" Value="White" />
      <Setter Property="Margin" Value="2,0"/>
      <Setter Property="Padding" Value="5"/>
    </Style>
  </Window.Resources >
</Window>
```

```

        </Style>

        ...

    </Window.Resources>

    <Grid>

        ...

        <z:DataNavigator Name="DatStudents" Padding="10" Background="Cyan"
            Width="420" Height="50" Margin="45,0,0,0"
            InfoTextStyle="{StaticResource InfoStyle}"
            ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
            ShowAddNew="True"
            ShowEdit="True"
            ShowCommit="True"
            ShowRemove="True"
            ShowToolTipsForButtons="True"
            ShowToolTipForInfo="True"
            ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
            MoveStarted="DatStudents_MoveStarted"
            MoveCompleted="DatStudents_MoveCompleted"
            AddNewStarted="DatStudents_AddNewStarted"
            EditItemStarted="DatStudents_EditItemStarted"
            CommitStarted="DatStudents_CommitStarted"
            CommitCompleted="DatStudents_CommitCompleted"
            RemoveStarted="DatStudents_RemoveStarted">

            </z:DataNavigator>

        </Grid>

    </Window>

```

**VB:**

```
Dim infoStyle As New Style(GetType(Label))
```

```
With infoStyle
```

```

    .Setters.Add(New Setter With {.[Property] = FontSizeProperty, .Value = 16.0})
    .Setters.Add(New Setter With {.[Property] = ForegroundColorProperty, _
        .Value = Brushes.White})
    .Setters.Add(New Setter With {.[Property] = BackgroundProperty, _
        .Value = _
            New SolidColorBrush(ColorConverter.ConvertFromString("#B39DDB"))})
    .Setters.Add(New Setter With {.[Property] = MarginProperty, _
        .Value = New Thickness(2, 0, 2, 0)})
    .Setters.Add(New Setter With {.[Property] = PaddingProperty, _
        .Value = New Thickness(5)})

```

```
End With
```

```
DatStudents.InfoTextStyle = infoStyle
```



## ShowAddNew

Καθορίζει αν θα εμφανίζεται το button AddNew (προσθήκης). Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowAddNew As Boolean
```

#### Τύπος: **System.Boolean**

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν θα εμφανίζεται το button AddNew. Η default τιμή είναι "True".

### Dependency Property Information:

**Identifier field:** ShowAddNewProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να μην εμφανίζεται το button AddNew.

#### XAML:

```
<zeus:DataNavigator Name="DatStudents" ShowAddNew ="False" ... />
```

#### VB:

```
DatStudents.ShowAddNew = False
```

## ShowCommit

Καθορίζει αν θα εμφανίζεται το `button Commit` (επιβεβαίωσης-αποθήκευσης). Είναι τύπου `Boolean`.

### Σύνταξη:

#### VB:

```
Public Property ShowCommit As Boolean
```

#### Τύπος: `System.Boolean`

Προσδιορίζουμε μία τιμή `Boolean` που καθορίζει αν θα εμφανίζεται το `button Commit`. Η default τιμή είναι `"True"`.

### Dependency Property Information:

**Identifier field:** `ShowCommitProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να μην εμφανίζεται το `button Commit`.

#### XAML:

```
<zeus:DataNavigator Name="DatStudents" ShowCommit="False" ... />
```

#### VB:

```
DatStudents.ShowCommit = False
```



## ShowEdit

Καθορίζει αν θα εμφανίζεται το button Edit (διόρθωσης). Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowEdit As Boolean
```

**Τύπος:** System.Boolean

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν θα εμφανίζεται το button Edit. Η default τιμή είναι "False".

### Dependency Property Information:

**Identifier field:** ShowEditProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να εμφανίζεται το button Edit.

#### XAML:

```
<zeus:DataNavigator Name="DatStudents" ShowEdit ="True" ... />
```

#### VB:

```
DatStudents.ShowEdit = True
```

## ShowRemove

Καθορίζει αν θα εμφανίζεται το **button Remove (διαγραφής)**. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowRemove As Boolean
```

**Τύπος:** **System.Boolean**

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν θα εμφανίζεται το button Remove. Η default τιμή είναι "True".

### Dependency Property Information:

**Identifier field:** ShowRemoveProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να μην εμφανίζεται το button Remove.

#### XAML:

```
<zeus:DataNavigator Name="DatStudents" ShowRemove ="False" ... />
```

#### VB:

```
DatStudents.ShowRemove = False
```

## ShowToolTipsForButtons

Καθορίζει αν θα εμφανίζονται τα ToolTips για τα buttons. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowToolTipsForButtons As Boolean
```

#### Τύπος: System.Boolean

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν θα εμφανίζονται τα tooltips στα buttons. Η default τιμή είναι "True".

### Dependency Property Information:

**Identifier field:** ShowToolTipsForButtonsProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να μην εμφανίζονται τα tooltips στα buttons.

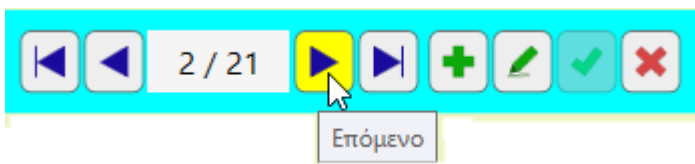
#### XAML:

```
<zeus:DataNavigator Name="DatStudents" ShowToolTipsForButtons ="False" ... />
```

#### VB:

```
DatStudents.ShowToolTipsForButtons = False
```

Στην παρακάτω εικόνα, το tooltip για το button MoveToNext εμφανίζεται με το default κείμενο. Αν θέσουμε την ιδιότητα ShowToolTipsForButtons σε False τότε δεν θα εμφανίζεται κανένα tooltip στα buttons.



## ShowToolTipForInfo

Καθορίζει αν θα εμφανίζεται το ToolTip για το πληροφοριακό Label. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowToolTipForInfo As Boolean
```

#### Τύπος: System.Boolean

Προσδιορίζουμε μία τιμή Boolean που καθορίζει αν θα εμφανίζεται το tooltip στο πληροφοριακό Label. Η default τιμή είναι "True".

### Dependency Property Information:

**Identifier field:** ShowToolTipForInfoProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να μην εμφανίζεται το tooltip στο Label:

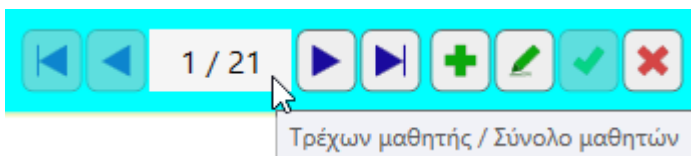
#### XAML:

```
<zeus:DataNavigator Name="DatStudents"  
    TooltipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"  
    ShowToolTipsForInfo ="False"  
    ... />
```

#### VB:

```
DatStudents.ShowToolTipsForInfo = False
```

Στην παρακάτω εικόνα, το tooltip για το Label εμφανίζεται. Αν θέσουμε την ιδιότητα ShowToolTipsForInfo σε False τότε θα αποκρύπτεται.



## ToolTipTextForAddNew

Καθορίζει το κείμενο για το **ToolTip** του **button AddNew**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForAddNew As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το tooltip του button AddNew. Η default τιμή είναι "Προσθήκη".

### Dependency Property Information:

**Identifier field:** ToolTipTextForAddNewProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Add new record" για το button AddNew. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) ShowToolTipsForButtons είναι True.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForAddNew="Add new record"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForAddNew = "Add new record"
```

## ToolTipTextForCancelEdit

Καθορίζει το κείμενο για το **ToolTip** του **button Remove**, όταν αυτό είναι σε κατάσταση διόρθωσης (*ακύρωση διόρθωσης*). Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForCancelEdit As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το tooltip του **button Remove**, όταν το **control** είναι σε κατάσταση διόρθωσης.

Η default τιμή είναι "Ακύρωση διόρθωσης".

### Dependency Property Information:

**Identifier field:** `ToolTipTextForCancelEditProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Cancel edit" για το **button Remove**, όταν το **control** βρίσκεται σε κατάσταση διόρθωσης. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) `ShowToolTipsForButtons` είναι `True`.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForCancelEdit="Cancel edit"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForCancelEdit = "Cancel edit"
```

## ToolTipTextForCancelNew

Καθορίζει το κείμενο για το **ToolTip** του **button Remove**, όταν αυτό είναι σε κατάσταση προσθήκης (*ακύρωση προσθήκης*). Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForCancelNew As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το tooltip του **button Remove**, όταν το **control** είναι σε κατάσταση προσθήκης.

Η default τιμή είναι "Ακύρωση προσθήκης".

### Dependency Property Information:

**Identifier field:** `ToolTipTextForCancelNewProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Cancel New" για το **button Remove**, όταν το **control** βρίσκεται σε κατάσταση προσθήκης. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) `ShowToolTipsForButtons` είναι `True`.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForCancelNew="Cancel new"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForCancelNew = "Cancel new"
```

## ToolTipTextForCommit

Καθορίζει το κείμενο για το **ToolTip** του **button Commit**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForCommit As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το tooltip του `button Commit`. Η default τιμή είναι "Αποθήκευση".

### Dependency Property Information:

**Identifier field:** `ToolTipTextForCommitProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Save record" για το `button Commit`. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) `ShowToolTipsForButtons` είναι `True`.

#### XAML:

```
<zeus:DataNavigator Name="DatStudents"
    ShowToolTipsForButtons="True"
    ToolTipTextForCommit="Save record"
    ... />

</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForCommit = "Save record"
```



## ToolTipTextForEdit

Καθορίζει το κείμενο για το **ToolTip** του **button Edit**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForEdit As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το tooltip του button **Edit**. Η default τιμή είναι "Διόρθωση".

### Dependency Property Information:

**Identifier field:** `ToolTipTextForEditProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Edit record" για το button **Edit**. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) `ShowToolTipsForButtons` είναι `True`.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForEdit="Edit record"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForEdit = "Edit record"
```

## ToolTipTextForInfo

Καθορίζει το κείμενο για το **ToolTip** του πληροφοριακού **Label**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForInfo As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το tooltip του πληροφοριακού **Label**. Η default τιμή είναι "Τρέχον / Σύνολο".

### Dependency Property Information:

**Identifier field:** ToolTipTextForInfoProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Record / Total" για το πληροφοριακό **Label**. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) **ShowToolTipForInfo** είναι **True**.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipForInfo="True"  
    ToolTipTextForInfo="Record / Total"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForInfo = "Record / Total"
```

## ToolTipTextForMoveToFirst

Καθορίζει το κείμενο για το **ToolTip** του **button MoveToFirst**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForMoveToFirst As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το tooltip του button MoveToFirst. Η default τιμή είναι "Πρώτο".

### Dependency Property Information:

**Identifier field:** ToolTipTextForMoveToFirstProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Go to First" για το button MoveToFirst. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) ShowToolTipsForButtons είναι True.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForMoveToFirst="Go to First"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForMoveToFirst = "Go to First"
```

## ToolTipTextForMoveToLast

Καθορίζει το κείμενο για το **ToolTip** του **button MoveToLast**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForMoveToLast As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το tooltip του button MoveToLast. Η default τιμή είναι "Τελευταίο".

### Dependency Property Information:

**Identifier field:** ToolTipTextForMoveToLastProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Go to Last" για το button MoveToLast. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) ShowToolTipsForButtons είναι True.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForMoveToLast="Go to Last"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForMoveToLast = "Go to Last"
```

## ToolTipTextForMoveToNext

Καθορίζει το κείμενο για το **ToolTip** του **button MoveToNext**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForMoveToNext As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το tooltip του button MoveToNext. Η default τιμή είναι "Επόμενο".

### Dependency Property Information:

**Identifier field:** ToolTipTextForMoveToNextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Go to Next" για το button MoveToNext. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) ShowToolTipsForButtons είναι True.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForMoveToNext="Go to Next"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForMoveToNext = "Go to Next"
```

## ToolTipTextForMoveToPrevious

Καθορίζει το κείμενο για το **ToolTip** του **button MoveToPrevious**. Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForMoveToPrevious As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το tooltip του **button MoveToPrevious**. Η default τιμή είναι "Προηγούμενο".

### Dependency Property Information:

**Identifier field:** `ToolTipTextForMoveToPreviousProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Go to Previous" για το **button MoveToPrevious**. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) **ShowToolTipsForButtons** είναι **True**.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForMoveToPrevious="Go to Previous"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForMoveToPrevious = "Go to Previous"
```

## ToolTipTextForRemove

Καθορίζει το κείμενο για το **ToolTip** του **button Remove**, όταν αυτό είναι σε κανονική κατάσταση (δηλαδή εκτός προσθήκης/διαγραφής). Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForRemove As String
```

#### Τύπος: System.String

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το tooltip του **button Remove**, όταν το control είναι σε κανονική κατάσταση πλοήγησης.  
Η default τιμή είναι "Διαγραφή".

### Dependency Property Information:

**Identifier field:** `ToolTipTextForRemoveProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το tooltip "Delete record" για το **button Remove**, όταν το control βρίσκεται σε κανονική κατάσταση. Να σημειώσουμε ότι το tooltip θα εμφανίζεται όταν η ιδιότητα (διακόπτης) **ShowToolTipsForButtons** είναι **True**.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents"  
    ShowToolTipsForButtons="True"  
    ToolTipTextForRemove="Delete record"  
    ... />  
  
</zeus:DataNavigator>
```

#### VB:

```
DatStudents.ToolTipTextForRemove = "Delete record"
```

## Συμβάντα

Όνομα	Περιγραφή
AddNewStarted	Ενεργοποιείται όταν ο χρήστης πατήσει το button "AddNew" ή όταν η ιδιότητα <code>IsAdding</code> τίθεται σε <code>True</code> ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος <code>AddNew</code> .
CommitCompleted	Ενεργοποιείται όταν κληθεί η μέθοδος <code>Commit</code> , οπότε ολοκληρώνεται η διαδικασία επιβεβαίωσης-αποθήκευσης.
CommitStarted	Ενεργοποιείται όταν ο χρήστης πατήσει το button "Commit" ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος <code>CommitStart</code> , οπότε ξεκινάει η διαδικασία επιβεβαίωσης-αποθήκευσης.
EditItemStarted	Ενεργοποιείται όταν ο χρήστης πατήσει το button "Edit" ή όταν η ιδιότητα <code>IsEditing</code> τίθεται σε <code>True</code> ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος <code>EditItem</code> .
MoveCompleted	Ενεργοποιείται όταν ολοκληρωθεί η μετακίνηση σε μία εγγραφή/αντικείμενο.
MoveStarted	Ενεργοποιείται όταν ξεκινήσει η μετακίνηση σε μία εγγραφή/αντικείμενο.
RemoveCompleted	Ενεργοποιείται όταν ολοκληρωθεί η διαγραφή μίας εγγραφής/αντικειμένου από τη συλλογή.
RemoveStarted	Ενεργοποιείται όταν ο χρήστης πατήσει το button "Remove".



## AddNewStarted

Ενεργοποιείται όταν ο χρήστης πατήσει το button "AddNew" ή όταν η ιδιότητα `IsAdding` τίθεται σε `True` ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος `AddNew`.

Σύνταξη:

**VB (ορισμός):**

```
Public Custom Event AddNewStarted As RoutedEventHandler
```

**XAML attribute usage:**

```
<zeus:DataNavigator AddNewStarted="eventHanlder" ... />
```

**Παρατηρήσεις:**

Όταν ο χρήστης πατήσει το button "AddNew" ή όταν η ιδιότητα `IsAdding` τίθεται σε `True` ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος `AddNew` τότε τα input controls της φόρμας (π.χ. `TextBoxes`) που σχετίζονται (με `binding`) με το αντικείμενο `CView` του `DataNavigator` γίνονται κενά, για να υποδεχτούν τις νέες τιμές. Όλα τα buttons απενεργοποιούνται και τα μόνο ενεργά είναι τα "Commit" και "Cancel New" (το "Remove" αυτόματα παίρνει αυτόν τον ρόλο σε κατάσταση προσθήκης).

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καθορίζουμε τον `DatStudents_AddNewStarted` event handler, ο οποίος εκτελείται όταν ο χρήστης πατήσει το button `AddNew`. Σε αυτόν τον event handler μπορούμε να κάνουμε κάποιες ενέργειες προετοιμασίας, όπως για παράδειγμα να ορίσουμε το κείμενο του πληροφοριακού `Label`, να θέσουμε την εστίαση σε κάποιο input control κ.α.

**XAML:**

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
CommitStarted="DatStudents_CommitStarted"
CommitCompleted="DatStudents_CommitCompleted"
RemoveStarted="DatStudents_RemoveStarted">
```

```
</zeus:DataNavigator>
```

VB:

'Όταν αρχίζει η προσθήκη μίας νέας εγγραφής. Αυτόματα, εμφανίζει κενά πεδία στη φόρμα και η ιδιότητα `DatStudents.IsAddingNew` γίνεται `True`.

```
Private Sub DatStudents_AddNewStarted(sender As Object, e As RoutedEventArgs)
```

```
    'Ενημερώνει το InfoText του DatStudents.  
    With DatStudents
```

```
        .InfoText = "+ / " & .Count - 1
```

```
    End With
```

```
    'Πήγαινε το focus στο πρώτο πεδίο της φόρμας.  
    txtFirstName.Focus()
```

```
End Sub
```

The screenshot shows a web form titled "Μαθητής" (Student) with a light green background. At the top right, there is a navigation bar with buttons for back, forward, and search, and a status indicator "+ / 21". The form contains the following fields:

- Όνομα: [Text input field]
- Επώνυμο: [Text input field]
- Ημερομ. Εγγραφής: [Date picker showing 4/8/2019]
- Βαθμός: [Text input field with value 0,0]
- Τάξη: [Dropdown menu showing A]
- Τμήμα: [Dropdown menu showing A1]
- Ενεργός: [Checked checkbox]

## CommitCompleted

Ενεργοποιείται όταν κληθεί η μέθοδος `Commit`, οπότε ολοκληρώνεται η διαδικασία επιβεβαίωσης-αποθήκευσης.

Σύνταξη:

**VB (ορισμός):**

```
Public Custom Event CommitCompleted As RoutedEventHandler
```

**XAML attribute usage:**

```
<zeus:DataNavigator CommitCompleted="eventHanlder" ... />
```

**Παρατηρήσεις:**

Η μέθοδος `Commit` ολοκληρώνει την διαδικασία επιβεβαίωσης-αποθήκευσης. Αν το control βρισκόταν σε κατάσταση προσθήκης/διόρθωσης τότε βγαίνει από αυτήν (Οι ιδιότητες `IsAddingNew` και `IsEditing` γίνονται `False`, επίσης ενεργοποιούνται ξανά τα buttons) και ενημερώνεται η υποκείμενη συλλογή (`CView`).

Στο συμβάν αυτό θα εκτελέσουμε όλες τις ενέργειες που ακολουθούν τη διαδικασία επιβεβαίωσης-αποθήκευσης, όπως, για παράδειγμα, την ενημέρωση του πληροφοριακού `Label`.

Να σημειώσουμε ότι, τις διαδικασίες αποθήκευσης των δεδομένων στη Βάση και οποιεσδήποτε άλλες που αφορούν τα source δεδομένα πραγματοποιούνται στον `CommitStarted` event handler. Απο εκεί, θα κληθεί η μέθοδος `Commit`. Αν, για κάποιο λόγο, αποτύχει η αποθήκευση των δεδομένων στην Βάση, τότε δεν καλούμε την `Commit` και συνεπώς δεν ενεργοποιείται το συμβάν `CommitCompleted`.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, καθορίζουμε τον `DatStudents_CommitCompleted` event handler, ο οποίος εκτελείται στον VB κώδικα αφού κληθεί η μέθοδος `Commit`. Σε αυτόν τον event handler μπορούμε να κάνουμε κάποιες τελικές ενέργειες, όπως για παράδειγμα να ορίσουμε το κείμενο του πληροφοριακού `Label`.

**XAML:**

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
```

```
ShowToolTipForInfo="True"  
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"  
MoveStarted="DatStudents_MoveStarted"  
MoveCompleted="DatStudents_MoveCompleted"  
AddNewStarted="DatStudents_AddNewStarted"  
EditItemStarted="DatStudents_EditItemStarted"  
CommitStarted="DatStudents_CommitStarted"  
CommitCompleted="DatStudents_CommitCompleted"  
RemoveStarted="DatStudents_RemoveStarted">
```

```
</zeus:DataNavigator>
```

**VB:**

```
'Ενημερώνει το InfoText του DatStudents, όταν ολοκληρωθεί η αποθήκευση.  
Private Sub DatStudents_CommitCompleted(sender As Object, e As RoutedEventArgs)  
  
    'Ενημερώνει το InfoText του DatStudents.  
    With DatStudents  
  
        .InfoText = .CurrentPosition + 1 & " / " & .Count  
  
    End With  
  
End Sub
```

## CommitStarted

Ενεργοποιείται όταν ο χρήστης πατήσει το button "Commit" ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος CommitStart, οπότε ξεκινάει η διαδικασία επιβεβαίωσης-αποθήκευσης.

Σύνταξη:

VB (ορισμός):

```
Public Custom Event CommitStarted As RoutedEventHandler
```

XAML attribute usage:

```
<zeus:DataNavigator CommitStarted="eventHanlder" ... />
```

Παρατηρήσεις:

Σε αυτόν τον event handler κάνουμε την αποθήκευση των δεδομένων στη Βάση ή σε οποιαδήποτε άλλη source πηγή δεδομένων (καλώντας από εδώ τις δικές μας ρουτίνες αποθήκευσης). Επίσης, μπορούμε να εκτελέσουμε κι άλλες ενέργειες που σχετίζονται με την αποθήκευση. Μόλις, ολοκληρωθούν όλα αυτά επιτυχώς, τότε καλούμε τη μέθοδο Commit. Η κλήση της μεθόδου Commit θα επικυρώσει την προσθήκη ή διόρθωση στη συλλογή CView και θα ενεργοποιήσει το συμβάν CommitCompleted.

Να σημειώσουμε ότι εδώ μπορούμε να κάνουμε κι έλεγχο εγκυρότητας των εισαγόμενων δεδομένων της φόρμας (λογικά, πριν την κλήση των ρουτινών αποθήκευσης στη Βάση).

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον **DatStudents\_CommitStarted** event handler. Στον VB κώδικα, εξετάζουμε σε τί κατάσταση είναι το control (προσθήκης ή διαγραφής), καλούμε τις ρουτίνες αποθήκευσης ή οτιδήποτε άλλο που σχετίζεται με την αποθήκευση, και στο τέλος καλείται η μέθοδος Commit.

XAML:

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
```

```

CommitStarted="DatStudents_CommitStarted"
CommitCompleted="DatStudents_CommitCompleted"
RemoveStarted="DatStudents_RemoveStarted">

```

```
</zeus:DataNavigator>
```

VB:

```

Private Sub DatStudents_CommitStarted(sender As Object, e As RoutedEventArgs)

    'Έλεγχος εγκυρότητας. Τον κάνουμε εδώ.
    If Not IsValid() Then

        MessageBox.Show("Η φόρμα έχει λάθη. Παρακαλώ, διορθώστε", _
            "Αποθήκευση", MessageBoxButton.OK, _
            MessageBoxImage.Error)

        Exit Sub

    End If

    If DatStudents.IsAddingNew Then

        If SaveStudentInAccess(DatStudents.CurrentAddItem, True) Then

            Dim pos = DatStudents.CurrentPosition
            obsStudents(pos).ID = newStudentID
            obsStudents(pos).ClassName =
                GetClassNameByIDFromCollection(obsStudents(pos).ClassID)
            obsStudents(pos).DivisionName =
                GetDivisionNameByIDFromCollection(obsStudents(pos).DivisionID)
            DatStudents.Commit()

        End If

    ElseIf DatStudents.IsEditing Then

        If SaveStudentInAccess(DatStudents.CurrentEditItem, False) Then

            Dim pos = DatStudents.CurrentPosition
            obsStudents(pos).ClassName =
                GetClassNameByIDFromCollection(obsStudents(pos).ClassID)
            obsStudents(pos).DivisionName =
                GetDivisionNameByIDFromCollection(obsStudents(pos).DivisionID)
            DatStudents.Commit()

        End If

    End If

End Sub

```

Στον event handler βλέπουμε ότι εφόσον η αποθήκευση στην Access (μέσω της ρουτίνας SaveStudentInAccess) ήταν επιτυχής, γίνονται κάποιες επιπλέον σχετικές ενέργειες (οι λεπτομέρειες των οποίων δεν μας ενδιαφέρουν εδώ) και στο τέλος καλείται πάντα η μέθοδος Commit. Έτσι, θα επικυρωθεί η προσθήκη ή διόρθωση αντικειμένου Student στη συλλογή CView του DatStudents.

## EditItemStarted

Ενεργοποιείται όταν ο χρήστης πατήσει το button "Edit" ή όταν η ιδιότητα `IsEditing` τίθεται σε `True` ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος `EditItem`.

Σύνταξη:

VB (ορισμός):

```
Public Custom Event EditItemStarted As RoutedEventHandler
```

XAML attribute usage:

```
<zeus:DataNavigator EditItemStarted="eventHanlder" ... />
```

Παρατηρήσεις:

Όταν ο χρήστης πατήσει το button "Edit" ή όταν η ιδιότητα `IsEditing` τίθεται σε `True` ή όταν κληθεί εσωτερικά από το πρόγραμμα η μέθοδος `EditItem` τότε όλα τα buttons απενεργοποιούνται και τα μόνο ενεργά είναι τα "Commit" και "Cancel Edit" (το "Remove" αυτόματα παίρνει αυτόν τον ρόλο σε κατάσταση διόρθωσης).

Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον `DatStudents_EditItemStarted` event handler, ο οποίος εκτελείται όταν ο χρήστης πατήσει το button `Edit`. Σε αυτόν τον event handler μπορούμε να κάνουμε κάποιες ενέργειες προετοιμασίας της διόρθωσης, όπως για παράδειγμα να πάρουμε backup την τρέχουσα εγγραφή/αντικείμενο ώστε αν ο χρήστης ακυρώσει την διόρθωση να επιστρέψουν πίσω τα παλιά δεδομένα.

XAML:

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
CommitStarted="DatStudents_CommitStarted"
CommitCompleted="DatStudents_CommitCompleted"
RemoveStarted="DatStudents_RemoveStarted">
```

```
</zeus:DataNavigator>
```

**VB:**

'Όταν αρχίζει η διόρθωση μίας υπάρχουσας (τρέχουσας) εγγραφής. Η ιδιότητα `DataNavigator.IsEditing` είναι `True`.

```
Private Sub DatStudents_EditItemStarted(sender As Object, e As RoutedEventArgs)
```

```
    BackupStudent()
```

```
    'Πήγαινε το focus στο πρώτο πεδίο και το σημείο εισαγωγής στο τέλος.  
    With txtFirstName
```

```
        .Focus()  
        .SelectionStart = .MaxLength
```

```
    End With
```

```
End Sub
```

```
Private backupStudentData As Student
```

'Για το edit mode, κρατάμε τα δεδομένα του μαθητή, ώστε να ξαναγυρίσουμε σε αυτά αν ο χρήστης κάνει cancel edit.

```
Private Sub BackupStudent()
```

```
    Dim currentStudent As Student = CType (DatStudents.CurrentItem, Student)  
    backupStudentData = New Student
```

```
    With backupStudentData  
        .LastName = currentStudent.LastName  
        .FirstName = currentStudent.FirstName  
        .RegistrationDate = currentStudent.RegistrationDate  
        .Grade = currentStudent.Grade  
        .ClassID = currentStudent.ClassID  
        .ClassName = currentStudent.ClassName  
        .DivisionID = currentStudent.DivisionID  
        .DivisionName = currentStudent.DivisionName  
        .IsActive = currentStudent.IsActive
```

```
    End With
```

```
End Sub
```



**Μαθητής**

4 / 21

Όνομα: ΕΛΕΑΝΑ

Επώνυμο: ΔΙΑΛΥΝΑ

Ημερομ. Εγγραφής: 4/6/2019 15

Βαθμός: 8,5

Τάξη: B

Τμήμα: B1

Ενεργός:

## MoveCompleted

Ενεργοποιείται όταν ολοκληρωθεί η μετακίνηση σε μία εγγραφή/αντικείμενο.

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event MoveCompleted As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:DataNavigator MoveCompleted="eventHanlder" ... />
```

### Παρατηρήσεις:

Όταν ο χρήστης πατήσει κάποιο από τα buttons "MoveToFirst", "MoveToPrevious", "MoveToNext", "MoveToLast", "AddNew" ο cursor προχωράει στην επόμενη εγγραφή/αντικείμενο. Όταν ολοκληρωθεί η μετακίνηση τότε καλείται το συμβάν MoveCompleted.

Συνήθως, σε αυτόν τον event handler, ενημερώνουμε το πληροφοριακό Label ή οτιδήποτε σχετίζεται με την μετακίνηση.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον **DatStudents\_MoveCompleted** event handler. Σε αυτόν τον event handler, ενημερώνουμε το πληροφοριακό Label (ιδιότητα InfoText) με το κατάλληλο μήνυμα, λαμβάνοντας υπόψη την ιδιαίτερη περίπτωση μήπως έχει πατηθεί το "AddNew".

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
CommitStarted="DatStudents_CommitStarted"
CommitCompleted="DatStudents_CommitCompleted"
RemoveStarted="DatStudents_RemoveStarted">
```

```
</zeus:DataNavigator>
```

**VB:**

```
'Όταν ολοκληρώνεται η μετακίνηση σε άλλη εγγραφή (προς υπάρχουσα ή νέα).  
Private Sub DatStudents_MoveCompleted(sender As Object, e As RoutedEventArgs)  
    With DatStudents  
        .InfoText = If(.IsAddingNew, "+ / " & .Count - 1, _  
                       .CurrentPosition + 1 & " / " & .Count)  
    End With  
End Sub
```

## MoveStarted

Ενεργοποιείται όταν ξεκινήσει η μετακίνηση σε μία εγγραφή/αντικείμενο.

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event MoveStarted As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:DataNavigator MoveStarted="eventHanlder" ... />
```

### Παρατηρήσεις:

Όταν ο χρήστης πατήσει κάποιο από τα buttons "MoveToFirst", "MoveToPrevious", "MoveToNext", "MoveToLast", "AddNew" ο cursor προχωράει στην επόμενη εγγραφή/αντικείμενο. Όταν ξεκινήσει αυτή η μετακίνηση τότε καλείται το συμβάν MoveStarted.

Συνήθως, σε αυτόν τον event handler, κρατάμε κάποια στοιχεία της τρέχουσας εγγραφή/αντικειμένου πριν γίνει η μετάβαση σε άλλη. Επίσης, είναι το ιδανικό σημείο για να κρατήσουμε το index του αντικειμένου της συλλογής CView όταν γίνεται μετάβαση σε νέα εγγραφή ("AddNew"). Αν ο χρήστης ακυρώσει τη νέα εγγραφή ("Cancel New") μπορούμε να μεταβούμε στην προηγούμενη που έβλεπε με βάση το index και τη μέθοδο MoveToPosition (αν δεν το κάνουμε, πηγαίνει στην τελευταία εγγραφή). Το παράδειγμα παρακάτω, δείχνει τον τρόπο.

Να σημειώσουμε ότι **δεν μπορούμε να ακυρώσουμε την μετακίνηση**. Αν επιζητούμε κάτι τέτοιο, τότε μπορούμε να χειριστούμε απ'ευθείας τον CurrentChanging event handler του CView, όπου η παράμετρος e έχει την ιδιότητα Cancel, οπότε θέτουμε e.Cancel = True.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον **DatStudents\_MoveStarted** event handler. Κρατάμε το index του τρέχοντος αντικειμένου Student πριν γίνει η μετάβαση στη νέα εγγραφή ("AddNew"). Αν ο χρήστης ακυρώσει την προσθήκη της νέας εγγραφής ("Cancel New") τότε, στον **DatStudents\_RemoveStarted** event handler, γίνεται μετάβαση με τη μέθοδο MoveToPosition σε αυτήν που εβλεπε προηγουμένως. Κάτι τέτοιο, είναι σημαντικό για την ευχρηστία της εφαρμογής μας.

**XAML:**

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
    Width="420" Height="50" Margin="45,0,0,0"
    InfoTextStyle="{StaticResource InfoStyle}"
    ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
    ShowEdit="True"
    ShowToolTipsForButtons="True"
    ShowToolTipForInfo="True"
    ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
    MoveStarted="DatStudents_MoveStarted"
    MoveCompleted="DatStudents_MoveCompleted"
    AddNewStarted="DatStudents_AddNewStarted"
    EditItemStarted="DatStudents_EditItemStarted"
    CommitStarted="DatStudents_CommitStarted"
    CommitCompleted="DatStudents_CommitCompleted"
    RemoveStarted="DatStudents_RemoveStarted">

</zeus:DataNavigator>
```

**VB:**

```
Private oldPosition As Integer = 0

'Όταν αρχίζει η μετακίνηση σε άλλη εγγραφή (προς υπάρχουσα ή νέα).
Private Sub DatStudents_MoveStarted(sender As Object, e As RoutedEventArgs)

    'Κρατάει την προηγούμενη θέση. Χρήσιμο, ώστε να επιστρέψει εκεί αν γίνει cancel
    new.
    If Not DatStudents.IsAddingNew Then oldPosition = DatStudents.CurrentPosition

End Sub

...

Private Sub DatStudents_RemoveStarted(sender As Object, e As RoutedEventArgs)

    If DatStudents.IsAddingNew Then 'Cancel New.

        DatStudents.RemoveCurrentItem()
        DatStudents.MoveToPosition(oldPosition)

    End If

    ...

End Sub
```

## RemoveCompleted

Ενεργοποιείται όταν ολοκληρωθεί η διαγραφή μίας εγγραφής/αντικειμένου από τη συλλογή.

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event RemoveCompleted As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:DataNavigator RemoveCompleted="eventHanlder" ... />
```

### Παρατηρήσεις:

Στον RemoveCompleted event handler τοποθετούμε VB κώδικα αφού έχει ολοκληρωθεί η διαγραφή μίας εγγραφής/αντικειμένου. Συνήθως, ένα ενημερωτικό μήνυμα.

Αν επιθυμούμε να δώσουμε τη δυνατότητα στον χρήστη να επιβεβαιώσει τη διαγραφή, θα το κάνουμε στον RemoveStarted event handler. Εκεί, θα κάνουμε και τις ενέργειες διαγραφής της εγγραφής/αντικειμένου από τη Βάση καλώντας τις ρουτίνες διαγραφής ή θα εκτελέσουμε κι άλλες ενέργειες που σχετίζονται με τη διαγραφή.

Να σημειώσουμε ότι ο event handler εκτελείται κι όταν γίνει ακύρωση προσθήκης νέας εγγραφής/αντικειμένου ("Cancel New").

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον **DatStudents\_RemoveCompleted** event handler. Σε αυτόν τον event handler, απλά ενημερώνουμε τον χρήστη ότι έγινε επιτυχώς η διαγραφή ή η ακύρωση προσθήκης.

#### XAML:

```
<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
CommitStarted="DatStudents_CommitStarted"
```

```
CommitCompleted="DatStudents_CommitCompleted"  
RemoveStarted="DatStudents_RemoveStarted"  
RemoveCompleted="DatStudents_RemoveCompleted">
```

```
</zeus:DataNavigator>
```

**VB:**

```
'Ολοκλήρωση της διαγραφής. Ισχύει και για την ακύρωση προσθήκης νέας εγγραφής.  
Private Sub DatStudents_RemoveCompleted(sender As Object, e As RoutedEventArgs)
```

```
    MessageBox.Show("Η διαγραφή ή ακύρωση προσθήκης έγινε με επιτυχία", _  
                    "Διαγραφή μαθητή", _  
                    MessageBoxButton.OK, _  
                    MessageBoxImage.Information)
```

```
End Sub
```

## RemoveStarted

Ενεργοποιείται όταν ο χρήστης πατήσει το button "Remove".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event RemoveStarted As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:DataNavigator RemoveStarted="eventHanlder" ... />
```

### Παρατηρήσεις:

Το button "Remove" έχει τριπλό ρόλο: Να διαγράψει μία εγγραφή/αντικείμενο (όταν έχει τον ρόλο "Delete"), να ακυρώσει την προσθήκη νέας εγγραφής/αντικειμένου (όταν έχει τον ρόλο "Cancel New") και να ακυρώσει την διόρθωση μίας εγγραφής/αντικειμένου (όταν έχει τον ρόλο "Cancel Edit").

Σε αυτόν τον event handler θα ελέγξουμε σε τί κατάσταση είναι το control (με τις ιδιότητες-διακόπτες `IsAddingNew` και `IsEditing`) ώστε να διαπιστώσουμε τί ρόλο έχει εκείνη τη στιγμή και να εκτελέσουμε τις κατάλληλες ενέργειες. Εδώ, θα εμφανίσουμε ένα **μήνυμα επιβεβαίωσης διαγραφής-ακύρωσης** κι αν ο χρήστης επιβεβαιώσει, για παράδειγμα την διαγραφή, να καλέσουμε τις ρουτίνες διαγραφής της εγγραφής/αντικειμένου από τη Βάση ή να εκτελέσουμε σχετικές ενέργειες.

Εκτός από την περίπτωση της ακύρωσης διόρθωσης, στο τέλος **μετά την επιτυχή διαγραφή από τη Βάση ή την ακύρωση προσθήκης πρέπει να καλέσουμε την μέθοδο `RemoveCurrentItem`**. Η κλήση αυτής της μεθόδου, θα απομακρύνει την εγγραφή/αντικείμενο από τη συλλογή `CollectionView`, θα ενεργοποιήσει τα buttons και θα ενεργοποιήσει, επίσης και το συμβάν `RemoveCompleted`.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον **`DatStudents_RemoveStarted`** event handler. Ελέγχουμε σε τί κατάσταση είναι το control, εμφανίζουμε σχετικό μήνυμα επιβεβαίωσης κι αν ο χρήστης δώσει το OK τότε εκτελούνται οι αντίστοιχες ενέργειες. Στην περίπτωση του "Cancel Edit" απλά επαναφέρουμε τα προηγούμενα δεδομένα και απενεργοποιούμε την κατάσταση διόρθωσης (`IsEditing=False`) ενώ στις άλλες περιπτώσεις ("Delete" και "Cancel New") πρέπει στο τέλος να καλέσουμε τη μέθοδο `RemoveCurrentItem`.



**XAML:**

```

<zeus:DataNavigator Name ="DatStudents" Padding="10" Background="Cyan"
Width="420" Height="50" Margin="45,0,0,0"
InfoTextStyle="{StaticResource InfoStyle}"
ButtonsStyle="{StaticResource NavigatorButtonsStyle}"
ShowEdit="True"
ShowToolTipsForButtons="True"
ShowToolTipForInfo="True"
ToolTipTextForInfo="Τρέχων μαθητής / Σύνολο μαθητών"
MoveStarted="DatStudents_MoveStarted"
MoveCompleted="DatStudents_MoveCompleted"
AddNewStarted="DatStudents_AddNewStarted"
EditItemStarted="DatStudents_EditItemStarted"
CommitStarted="DatStudents_CommitStarted"
CommitCompleted="DatStudents_CommitCompleted"
RemoveStarted="DatStudents_RemoveStarted">

</zeus:DataNavigator>

```

**VB:**

```
Private oldPosition As Integer = 0
```

'Όταν αρχίζει η μετακίνηση σε άλλη εγγραφή (προς υπάρχουσα ή νέα).

```
Private Sub DatStudents_MoveStarted(sender As Object, e As RoutedEventArgs)
```

```
    'Κρατάει την προηγούμενη θέση. Χρήσιμο, ώστε να επιστρέψει εκεί αν γίνει cancel new.
```

```
    If Not DatStudents.IsAddingNew Then oldPosition = DatStudents.CurrentPosition
```

```
End Sub
```

'Όταν ξεκινάει η διαδικασία διαγραφής ή ακύρωση προσθήκης ή ακύρωση διόρθωσης.

```
Private Sub DatStudents_RemoveStarted(sender As Object, e As RoutedEventArgs)
```

```
    Dim strMessage As String = ""
```

```
    Dim strTitle As String = ""
```

```
    If DatStudents.IsAddingNew Then
```

```
        strMessage = "Ακύρωση εισαγωγής;"
```

```
        strTitle = "Εισαγωγή"
```

```
    ElseIf DatStudents.IsEditing Then
```

```
        strMessage = "Ακύρωση διόρθωσης;"
```

```
        strTitle = "Διόρθωση"
```

```
    Else
```

```
        Dim currentStudent As Student = CType (DatStudents.CurrentItem, Student)
```

```
        strMessage = _
```

```
            String.Format("Θέλετε να διαγράψετε τον/την μαθητή/τρια: {0} {1};", _
                currentStudent.LastName, currentStudent.FirstName)
```

```
        strTitle = "Διαγραφή"
```

```
    End If
```

```
    If MessageBox.Show(strMessage, strTitle, MessageBoxButton.YesNo) = _
```

---

`MessageBoxResult.Yes Then`

```
If DatStudents.IsEditing Then 'Cancel edit
    RestoreStudent()
    DatStudents.IsEditing = False

    ElseIf DatStudents.IsAddingNew Then 'Cancel New

        DatStudents.RemoveCurrentItem()
        DatStudents.MoveToPosition(oldPosition)

    Else 'remove/delete

        If DeleteStudentInAccess(DatStudents.CurrentItem) Then
            DatStudents.RemoveCurrentItem()
        End If

    End If

End If

End Sub
```

```
Private backupStudentData As Student
```

```
'Επαναφέρει τα δεδομένα του μαθητή, αν κάνει cancel edit.
```

```
Private Sub RestoreStudent()
```

```
    Dim currentStudent As Student = obsStudents(DatStudents.CurrentPosition)
```

```
    With currentStudent
```

```
        .LastName = backupStudentData.LastName
        .FirstName = backupStudentData.FirstName
        .RegistrationDate = backupStudentData.RegistrationDate
        .Grade = backupStudentData.Grade
        .ClassID = backupStudentData.ClassID
        .ClassName = backupStudentData.ClassName
        .DivisionID = backupStudentData.DivisionID
        .DivisionName = backupStudentData.DivisionName
        .IsActive = backupStudentData.IsActive
    End With
```

```
End Sub
```

Στον `EditItemStarted` event handler έχουμε ήδη πάρει backup τα δεδομένα του μαθητή πριν ξεκινήσει η διόρθωση από τον χρήστη. Σε περίπτωση που κάνει ακύρωση διόρθωσης ("Cancel Edit"), τα επαναφέρουμε μέσω της μεθόδου `RestoreStudent`.

Στην παρακάτω εικόνα, εμφανίζεται το μήνυμα επιβεβαίωσης για τη διαγραφή του μαθητή ΑΝΤΥΠΙΑ ΘΑΝΑΣΗ:

**Μαθητής**

2 / 21

Όνομα: ΘΑΝΑΣΗΣ  
Επώνυμο: ΑΝΤΥΠΑΣ  
Ημερομ. Εγγραφής: 4/6/2019  
Βαθμός: 15,2  
Τάξη: Γ  
Τμήμα: Γ1  
Ενεργός: Ναι

Διαγραφή

Θέλετε να διαγράψετε τον/την μαθητή/τρια: ΑΝΤΥΠΑΣ ΘΑΝΑΣΗΣ;

Ναι Όχι

*Φυσικά, αν είχαμε πληροφορία για το φύλο του μαθητή το μήνυμα επιβεβαίωσης θα ήταν πιο σαφές.*

## Μέθοδοι

Όνομα	Περιγραφή
AddNew()	Ξεκινάει μία διαδικασία προσθήκης νέας εγγραφής/αντικειμένου (add transaction).  Επιστρέφει την υπό έγκριση (commit) νέα εγγραφή/αντικείμενο κι ενεργοποιεί το συμβάν AddNewStarted.
AddNewItem(newItem As Object)	Ξεκινάει μία διαδικασία προσθήκης νέας εγγραφής/αντικειμένου (add transaction) και προσθέτει το <b>newItem</b> στη συλλογή <b>CView</b> (μόνο για sub-class <i>ListCollectionView</i> ). Επιστρέφει την υπό έγκριση (commit) νέα εγγραφή/αντικείμενο κι ενεργοποιεί το συμβάν AddNewStarted.
Commit()	<b>Ολοκληρώνει μία διαδικασία προσθήκης/διόρθωσης (add/edit transaction).</b>  Ενεργοποιεί το συμβάν CommitCompleted.
CommitStart()	Ξεκινάει τη διαδικασία επιβεβεβαίωσης-αποθήκευσης.  Ενεργοποιεί το συμβάν CommitStarted.
Count()	Επιστρέφει τον αριθμό εγγραφών/αντικειμένων στη συλλογή <b>CView</b> .
CurrentAddItem()	Επιστρέφει την υπό έγκριση νέα εγγραφή/αντικείμενο, όταν μία διαδικασία προσθήκης έχει ξεκινήσει (add transaction).
CurrentEditItem()	Επιστρέφει την υπό έγκριση εγγραφή/αντικείμενο, όταν μία διαδικασία διόρθωσης έχει ξεκινήσει (edit transaction).
CurrentItem()	Επιστρέφει την τρέχουσα εγγραφή/αντικείμενο.
CurrentPosition()	Επιστρέφει την θέση της τρέχουσας εγγραφής/αντικειμένου.

EditItem(item As Object)	<b>Ξεκινάει μία διαδικασία διόρθωσης του item (edit transaction).</b> Ενεργοποιεί το συμβάν EditItemStarted.
HasItems()	<b>Επιστρέφει True αν η συλλογή CView έχει εγγραφές/αντικείμενα.</b>
MoveToFirst()	<b>Μετακινεί τον cursor στην πρώτη εγγραφή/αντικείμενο της συλλογής CView.</b>
MoveToLast()	<b>Μετακινεί τον cursor στην τελευταία εγγραφή/αντικείμενο της συλλογής CView.</b>
MoveToNext()	<b>Μετακινεί τον cursor στην επόμενη εγγραφή/αντικείμενο της συλλογής CView.</b>
MoveToPosition(pos As Integer)	<b>Μετακινεί τον cursor στην συγκεκριμένη θέση pos της συλλογής CView.</b>
MoveToPrevious()	<b>Μετακινεί τον cursor στην προηγούμενη εγγραφή/αντικείμενο της συλλογής CView.</b>

## AddNew

Ξεκινάει μία διαδικασία προσθήκης νέας εγγραφής/αντικείμενου (add transaction). Επιστρέφει την υπό έγκριση (commit) νέα εγγραφή/αντικείμενο κι ενεργοποιεί το συμβάν AddNewStarted.

### Σύνταξη:

**VB :**

```
Public Function AddNew()
```

*Επιστρεφόμενη τιμή:*

Object.

Το item που θα προστεθεί στη συλλογή CView.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καλούμε την AddNew, από ένα δικό μας button "Add", η οποία επιστρέφει ένα νέο αντικείμενο τύπου Student και προ-συμπληρώνουμε αυτόματα μερικές τιμές. Αυτό το νέο αντικείμενο Student δεν έχει προστεθεί στη συλλογή CView μέχρι να κάνουμε Commit.

**VB:**

```
Private Sub btnAdd_Click(sender As Object, e As RoutedEventArgs)
```

```
    Dim student As Student = DatStudents.AddNew
```

```
    student.RegistrationDate = #6/15/2019#
```

```
    student.Grade = 20
```

```
End Sub
```

## AddNewItem

Ξεκινάει μία διαδικασία προσθήκης νέας εγγραφής/αντικειμένου (add transaction) και προσθέτει το `newItem` στη συλλογή `CView` (μόνο για sub-class `ListCollectionView`).

Επιστρέφει την υπό έγκριση (commit) νέα εγγραφή/αντικείμενο κι ενεργοποιεί το συμβάν `AddNewStarted`.

### Σύνταξη:

#### VB :

```
Public Function AddNewItem (  
    newItem As Object  
)
```

Παράμετροι:

*newItem*

Τύπος: Object

Το αντικείμενο που θα προστεθεί στη συλλογή `CView`.

Επιστρεφόμενη τιμή:

Τύπος: Object.

Το αντικείμενο που θα προστεθεί στη συλλογή `CView`.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καλούμε την `AddNewItem`, από ένα δικό μας (custom) button "AddNew", με ένα έτοιμο αντικείμενο τύπου `Student` που έχει συμπληρωμένες τιμές. Αυτό το νέο αντικείμενο `Student` δεν έχει προστεθεί στη συλλογή `CView` μέχρι να κάνουμε `Commit`.

#### VB:

```
Private Sub btnAdd_Click(sender As Object, e As RoutedEventArgs)  
  
    Dim student As New Student  
  
    With student  
        .FirstName = "Χρήστος"  
        .LastName = "Μουρατίδης"  
        .RegistrationDate = #6/15/2019#  
        .Grade = 20  
        .IsActive = True  
        .ClassID = 3 'Γ  
        .DivisionID = 6 'Γ1  
    End With  
  
    DatStudents.AddNewItem(student)  
  
End Sub
```

## Commit

Ολοκληρώνει μία διαδικασία προσθήκης/διόρθωσης (add/edit transaction). Ενεργοποιεί το συμβάν CommitCompleted.

### Σύνταξη:

#### VB :

```
Public Function Commit()
```

Επιστρεφόμενη τιμή:

Boolean.

True, αν η επιβεβαίωση ήταν επιτυχής.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καλούμε την Commit, μέσα στον CommitStarted event handler, και εφόσον έχει γίνει επιτυχώς η αποθήκευση της εγγραφής/αντικειμένου στη Βάση Access μέσω της ρουτίνας SaveStudentInAccess. Όπως βλέπουμε, λαμβάνεται υπόψη, αν είναι σε add ή edit transaction. Μπορούμε να εκτελέσουμε κι άλλες ενέργειες που σχετίζονται με την αποθήκευση της εγγραφής αλλά, σε κάθε περίπτωση, καλούμε στο τέλος πάντα την Commit για να ολοκληρώσει το add/edit transaction. Τεχνικά μιλώντας, η κλήση της μεθόδου Commit, επιβεβαιώνει την προσθήκη/διόρθωση της εγγραφής στη συλλογή CView κι επιστρέφει το control στην κανονική κατάσταση.

#### VB:

```
'CommitStarted event handler.
```

```
Private Sub DatStudents_CommitStarted(sender As Object, e As RoutedEventArgs)
```

```
    If DatStudents.IsAddingNew Then
```

```
        If SaveStudentInAccess(DatStudents.CurrentAddItem, True) Then
```

```
            Dim pos As Integer = DatStudents.CurrentPosition
```

```
            obsStudents(pos).ID = newStudentID
```

```
            obsStudents(pos).ClassName = _
```

```
                GetClassNameByIDFromCollection(obsStudents(pos).ClassID)
```

```
            obsStudents(pos).DivisionName = _
```

```
                GetDivisionNameByIDFromCollection(obsStudents(pos).DivisionID)
```

```
            DatStudents.Commit()
```

```
        End If
```

```
    ElseIf DatStudents.IsEditing Then
```

```
        If SaveStudentInAccess(DatStudents.CurrentEditItem, False) Then
```

```
            Dim pos As Integer = DatStudents.CurrentPosition
```

```
            obsStudents(pos).ClassName = _
```

```
                GetClassNameByIDFromCollection(obsStudents(pos).ClassID)
```

```
            obsStudents(pos).DivisionName = _
```

```
                GetDivisionNameByIDFromCollection(obsStudents(pos).DivisionID)
```

```
            DatStudents.Commit()
```



End If

End If

End Sub

## CommitStart

Ξεκινάει τη διαδικασία επιβεβαίωσης-αποθήκευσης. Ενεργοποιεί το συμβάν CommitStarted.

### Σύνταξη:

**VB :**

```
Public Function CommitStart()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η εκκίνηση της επιβεβαίωσης-αποθήκευσης ήταν επιτυχής.

### Παρατηρήσεις:

Το συμβάν CommitStarted ενεργοποιείται όταν ο χρήστης πατήσει το button "Commit" του DataNavigator. Αν επιθυμούμε να εκκινήσουμε τη διαδικασία επιβεβαίωσης από άλλο σημείο (για παράδειγμα, από ένα δικό μας custom button "Commit"), τότε θα καλέσουμε από εκεί τη μέθοδο CommitStart.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καλούμε την μέθοδο CommitStart από ένα custom button με όνομα btnCommitStart.

**VB:**

```
'btnCommitStart Click event handler.
```

```
Private Sub btnCommitStart_Click(sender As Object, e As RoutedEventArgs)
```

```
    DatStudents.CommitStart()
```

```
End Sub
```

```
'CommitStarted event handler.
```

```
Private Sub DatStudents_CommitStarted(sender As Object, e As RoutedEventArgs)
```

```
    If DatStudents.IsAddingNew Then
```

```
        If SaveStudentInAccess(DatStudents.CurrentAddItem, True) Then
```

```
            Dim pos = DatStudents.CurrentPosition
```

```
            obsStudents(pos).ID = newStudentID
```

```
            obsStudents(pos).ClassName = _
```

```
                GetClassNameByIDFromCollection(obsStudents(pos).ClassID)
```

```
            obsStudents(pos).DivisionName = _
```

```
                GetDivisionNameByIDFromCollection(obsStudents(pos).DivisionID)
```

```
            DatStudents.Commit()
```

```
        End If
```

```
ElseIf DatStudents.IsEditing Then

    If SaveStudentInAccess(DatStudents.CurrentEditItem, False) Then
        Dim pos = DatStudents.CurrentPosition
        obsStudents(pos).ClassName = _
            GetClassNameByIDFromCollection(obsStudents(pos).ClassID)
        obsStudents(pos).DivisionName = _
            GetDivisionNameByIDFromCollection(obsStudents(pos).DivisionID)
        DatStudents.Commit()
    End If

End If

End Sub
```

## Count

Επιστρέφει τον αριθμό εγγραφών/αντικειμένων στη συλλογή CView.

**Σύνταξη:**

**VB :**

```
Public Function Count()
```

*Επιστρεφόμενη τιμή:*

Integer.

Το πλήθος των εγγραφών/αντικειμένων στη συλλογή CView.

**Παράδειγμα:**

**VB:**

```
Dim cntRecords As Integer = DatStudents.Count()
```

## CurrentAddItem

Επιστρέφει την υπό έγκριση νέα εγγραφή/αντικείμενο, όταν μία διαδικασία προσθήκης έχει ξεκινήσει (add transaction).

**Σύνταξη:**

**VB :**

```
Public Function CurrentAddItem()
```

*Επιστρεφόμενη τιμή:*

Τύπος: Object.

Το νέο αντικείμενο που είναι σε διαδικασία προσθήκης (add transaction) στη συλλογή CView.

**Παρατηρήσεις:**

Το καλύτερο σημείο για να έχουμε πρόσβαση στο νέο αντικείμενο (CurrentAddItem) που πρόκειται να προστεθεί στη συλλογή CView είναι στον AddNewStarted event handler, ο οποίος ενεργοποιείται αυτόματα όταν ο χρήστης πατήσει το button "AddNew" στο DataNavigator.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, λαμβάνουμε το CurrentAddItem μέσα στον AddNewStarted event handler και προ-συμπληρώνουμε κάποια στοιχεία.

**VB:**

```
Private Sub DatStudents_AddNewStarted(sender As Object, e As RoutedEventArgs)

    Dim student As Student = DatStudents.CurrentAddItem()

    With student
        .FirstName = "Χρήστος"
        .LastName = "Μουρατίδης"
        .RegistrationDate = #6/15/2019#
        .Grade = 20
        .IsActive = True
        .ClassID = 3 'Γ
        .DivisionID = 6 'Γ1
    End With

End Sub
```

## CurrentEditItem

Επιστρέφει την υπό έγκριση εγγραφή/αντικείμενο, όταν μία διαδικασία διόρθωσης έχει ξεκινήσει (edit transaction).

**Σύνταξη:**

**VB :**

```
Public Function CurrentEditItem()
```

*Επιστρεφόμενη τιμή:*

Τύπος: Object.

Το αντικείμενο που είναι σε διαδικασία διόρθωσης (edit transaction) στη συλλογή CView.

**Παρατηρήσεις:**

Το καλύτερο σημείο για να έχουμε πρόσβαση στο υπό διόρθωση αντικείμενο (CurrentEditItem) είναι στον EditItemStarted event handler, ο οποίος ενεργοποιείται αυτόματα όταν ο χρήστης πατήσει το button "Edit" στο DataNavigator.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, μέσα στον EditItemStarted event handler, καλούμε την ρουτίνα BackupStudent με παράμετρο το CurrentEditItem, ώστε να κρατήσει τα δεδομένα της εγγραφής σε ένα backup αντικείμενο Student, πριν ο χρήστης ξεκινήσει τη διόρθωση.

**VB:**

```
Private backupStudentData As Student
```

```
Private Sub DatStudents_EditItemStarted(sender As Object, e As RoutedEventArgs)
```

```
    BackupStudent(DatStudents.CurrentEditItem)
```

```
End Sub
```

```
Private Sub BackupStudent(editStudent As Student)
```

```
    backupStudentData = New Student
```

```
    With backupStudentData
```

```
        .LastName = editStudent.LastName
```

```
        .FirstName = editStudent.FirstName
```

```
        .RegistrationDate = editStudent.RegistrationDate
```

```
        .Grade = editStudent.Grade
```

```
        .ClassID = editStudent.ClassID
```

```
        .ClassName = editStudent.ClassName
```

```
.DivisionID = editStudent.DivisionID  
.DivisionName = editStudent.DivisionName  
.IsActive = editStudent.IsActive  
End With  
  
End Sub
```

## CurrentItem

Επιστρέφει την τρέχουσα εγγραφή/αντικείμενο.

**Σύνταξη:**

**VB :**

```
Public Function CurrentItem()
```

*Επιστρεφόμενη τιμή:*

Τύπος: Object.

Το τρέχον αντικείμενο της συλλογής CView.

**Παρατηρήσεις:**

Το CurrentItem πάντα επιστρέφει την εγγραφή/αντικείμενο όπου βρίσκεται ο cursor (ή *record pointer*. Στο *manual* χρησιμοποιούμε τον όρο *cursor*). Αν το control βρίσκεται σε διαδικασία προσθήκης επιστρέφει την υπό έγκριση νέα εγγραφή/αντικείμενο και σε διαδικασία διόρθωσης επιστρέφει την υπό έγκριση διορθωμένη εγγραφή/αντικείμενο.

**Παράδειγμα:**

**VB:**

```
Dim student As Student = DatStudents.CurrentItem()
```



## CurrentPosition

Επιστρέφει την θέση της τρέχουσας εγγραφής/αντικειμένου.

Σύνταξη:

**VB :**

```
Public Function CurrentPosition()
```

*Επιστρεφόμενη τιμή:*

Τύπος: Integer.

Το index της τρέχουσας εγγραφής/αντικειμένου στη συλλογή CView. Η πρώτη εγγραφή έχει index 0.

**Παρατηρήσεις:**

Το αντικείμενο CView μπορεί να το χειριστεί άμεσα ο προγραμματιστής και να εφαρμόσει Ταξινόμηση (Sort) και Φιλτράρισμα (Filter). Στην περίπτωση αυτή, η ιδιότητα CurrentPosition επιστρέφει τη θέση της τρέχουσας εγγραφής/αντικειμένου λαμβάνοντας υπόψη ενδεχόμενη ταξινόμηση ή/και φιλτράρισμα.

Σε μελλοντική έκδοση του DataNavigator θα μελετηθεί η δυνατότητα άμεσης παροχής Ταξινόμησης και Φιλτραρίσματος των εγγραφών του CView.

**Παράδειγμα:**

**VB:**

```
Dim pos As Integer = DatStudents.CurrentPosition
```

## EditItem

Ξεκινάει μία διαδικασία διόρθωσης του item (edit transaction). Ενεργοποιεί το συμβάν EditItemStarted.

### Σύνταξη:

#### VB :

```
Public Function EditItem(  
    item As Object  
)
```

Παράμετροι:

*item*

Τύπος: Object

Το αντικείμενο που τίθεται υπό διόρθωση.

*Επιστρεφόμενη τιμή:*

Τύπος: Boolean.

True, αν η εκκίνηση της διόρθωσης ήταν επιτυχής.

### Παράδειγμα:

Το συμβάν EditItemStarted ενεργοποιείται όταν ο χρήστης πατήσει το button "Edit" του DataNavigator. Αν επιθυμούμε να εκκινήσουμε τη διαδικασία διόρθωσης από άλλο σημείο (για παράδειγμα, από ένα δικό μας custom button "Edit"), τότε θα καλέσουμε από εκεί τη μέθοδο EditItem (περνώντας ως παράμετρο, συνήθως την τρέχουσα εγγραφή/αντικείμενο, δηλαδή αυτή που βλέπει εκείνη τη στιγμή ο χρήστης).

#### VB:

```
Private Sub btnEdit_Click(sender As Object, e As RoutedEventArgs)  
    With DatStudents  
        If Not .IsAddingNewOrIsEditing Then .EditItem(.CurrentItem)  
    End With  
End Sub
```

## HasItems

Επιστρέφει **True** αν η συλλογή **CView** έχει εγγραφές/αντικείμενα.

**Σύνταξη:**

**VB :**

```
Public Function HasItems()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η συλλογή **CView** έχει εγγραφές/αντικείμενα.

**Παρατηρήσεις:**

Αυτή η ιδιότητα παρέχεται για ευκολία αντί να χρησιμοποιούμε την σύγκριση `Count > 0`.

**Παράδειγμα:**

**VB:**

```
Dim hasRecords As Boolean = DatStudents.HasItems()
```

## MoveToFirst

Μετακινεί τον `cursor` στην πρώτη εγγραφή/αντικείμενο της συλλογής `CView`.

**Σύνταξη:**

**VB :**

```
Public Function MoveToFirst()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η μετακίνηση ήταν επιτυχής.

**Παρατηρήσεις:**

Η επιτυχής μετακίνηση ενεργοποιεί το συμβάν `MoveCompleted`.

**Παράδειγμα:**

**VB:**

```
DatStudents.MoveToFirst()
```

## MoveToLast

Μετακινεί τον `cursor` στην τελευταία εγγραφή/αντικείμενο της συλλογής `CView`.

**Σύνταξη:**

**VB :**

```
Public Function MoveToLast()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η μετακίνηση ήταν επιτυχής.

**Παρατηρήσεις:**

Η επιτυχής μετακίνηση ενεργοποιεί το συμβάν `MoveCompleted`.

**Παράδειγμα:**

**VB:**

```
DatStudents.MoveToLast()
```

## MoveToNext

Μετακινεί τον `cursor` στην επόμενη εγγραφή/αντικείμενο της συλλογής `CView`.

**Σύνταξη:**

**VB :**

```
Public Function MoveToNext()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η μετακίνηση ήταν επιτυχής.

**Παρατηρήσεις:**

Η επιτυχής μετακίνηση ενεργοποιεί το συμβάν `MoveCompleted`.

**Παράδειγμα:**

**VB:**

```
DatStudents.MoveToNext()
```

## MoveToPosition

Μετακινεί τον `cursor` στην συγκεκριμένη θέση προς της συλλογής `CView`.

**Σύνταξη:**

**VB :**

```
Public Function MoveToPosition(  
    pos As Integer  
)
```

Παράμετροι:

*pos*

Τύπος: Integer

Η θέση (index) στην συλλογή `CView`. Το index ξεκινάει από το 0.

Επιστρεφόμενη τιμή:

Boolean.

True, αν η μετακίνηση ήταν επιτυχής.

**Παρατηρήσεις:**

Η επιτυχής μετακίνηση ενεργοποιεί το συμβάν `MoveCompleted`.

**Παράδειγμα:**

**VB:**

```
DatStudents.MoveToPosition(5)
```

## MoveToPrevious

Μετακινεί τον cursor στην προηγούμενη εγγραφή/αντικείμενο της συλλογής CView.

**Σύνταξη:**

**VB :**

```
Public Function MoveToPrevious()
```

*Επιστρεφόμενη τιμή:*

Boolean.

True, αν η μετακίνηση ήταν επιτυχής.

**Παρατηρήσεις:**

Η επιτυχής μετακίνηση ενεργοποιεί το συμβάν MoveCompleted.

**Παράδειγμα:**

**VB:**

```
DatStudents.MoveToPrevious()
```



## Styles και Templates

- Parts και States
- To default Style και ControlTemplate

## Parts και States

Το default ControlTemplate περιλαμβάνει κάποια **part names** και **visual states**. Μπορείτε να τροποποιήσετε το default ControlTemplate ώστε να δώσετε στο control μία μοναδική εμφάνιση.

Ο παρακάτω πίνακας εμφανίζει τα **part names** του **DataNavigator**:

Part	Τύπος	Περιγραφή
PART_First	Button	Το button "MoveToFirst".
PART_Previous	Button	Το button "MoveToPrevious".
PART_Next	Button	Το button "MoveToNext".
PART_Last	Button	Το button "MoveToLast".
PART_InfoText	Label	Το πληροφοριακό Label μεταξύ των buttons πλοήγησης.
PART_AddNew	Button	Το button "AddNew".
PART_Edit	Button	Το button "Edit".
PART_Commit	Button	Το button "Commit".
PART_Remove	Button	Το button "Remove".

Ο παρακάτω πίνακας εμφανίζει τα **visual states** του **DataNavigator**:

VisualState	VisualStateGroup	Περιγραφή
Normal	CommonStates	Το default state.
MouseOver	CommonStates	Το state όταν ο δείκτης του ποντικιού είναι πάνω από το control.
Disabled	CommonStates	Το state όταν το control είναι disabled.
Focused	FocusStates	Το state όταν το control έχει το focus.
Unfocused	FocusStates	Το state όταν το control δεν έχει το focus.

*Το default ControlTemplate έχει καθορισμένο μόνο το Disabled state. Μπορείτε να δημιουργήσετε ένα custom ControlTemplate για να το αλλάξετε ή/και για να καθορίσετε τα υπόλοιπα.*

## To default Style και ControlTemplate

Ο XAML κώδικας για το **default Style** και **ControlTemplate** φαίνεται παρακάτω. Μπορείτε να βασιστείτε σε αυτόν για να δημιουργήσετε μία μικρή ή μεγάλη παραλλαγή του δικού σας custom Style και ControlTemplate:

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.DataControls">

<Style TargetType="{x:Type z:DataNavigator}">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type z:DataNavigator}">
                <ControlTemplate.Resources >
                    <z:BooleanToVisibilityConverter
                        x:Key="booleanToVisibilityConverter"/>
                </ControlTemplate.Resources>

                <!-- Root element -->
                <Border Name="border" Background="{TemplateBinding Background}"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}" >

                    <!-- Visual States -->
                    <VisualStateManager.VisualStateGroups >

                        <VisualStateGroup Name="CommonStates">

                            <VisualStateGroup.Transitions >
                                <VisualTransition GeneratedDuration="0:0:0.2">
                                </VisualTransition>
                            </VisualStateGroup.Transitions>

                            <VisualState Name="Normal"/>
                            <VisualState Name="MouseOver" />
                            <VisualState Name="Disabled">
                                <Storyboard >
                                    <DoubleAnimation
                                        Storyboard.TargetName="PART_First"
                                        Storyboard.TargetProperty="(Button.Opacity)"
                                        To="0.5" />
                                    <DoubleAnimation
                                        Storyboard.TargetName="PART_Previous"
                                        Storyboard.TargetProperty="(Button.Opacity)"
                                        To="0.5" />
                                    <DoubleAnimation
                                        Storyboard.TargetName="PART_Next"
                                        Storyboard.TargetProperty="(Button.Opacity)"
                                        To="0.5" />
                                    <DoubleAnimation
                                        Storyboard.TargetName="PART_Last"
                                        Storyboard.TargetProperty="(Button.Opacity)"
                                        To="0.5" />
                                    <DoubleAnimation
```

```

        Storyboard.TargetName="PART_AddNew"
        Storyboard.TargetProperty="(Button.Opacity)"
        To="0.5" />
    <DoubleAnimation
        Storyboard.TargetName="PART_Edit"
        Storyboard.TargetProperty="(Button.Opacity)"
        To="0.5" />
    <DoubleAnimation
        Storyboard.TargetName="PART_Commit"
        Storyboard.TargetProperty="(Button.Opacity)"
        To="0.5" />
    <DoubleAnimation
        Storyboard.TargetName="PART_Remove"
        Storyboard.TargetProperty="(Button.Opacity)"
        To="0.5" />
    <DoubleAnimation
        Storyboard.TargetName="PART_InfoText"
        Storyboard.TargetProperty="(Label.Opacity)"
        To="0.6" />
    </Storyboard>
</VisualState>
</VisualStateGroup>

<VisualStateGroup Name="FocusStates">

    <VisualStateGroup.Transitions >
        <VisualTransition GeneratedDuration="0:0:0.2">
        </VisualTransition>
    </VisualStateGroup.Transitions>

    <VisualState Name="Focused" />
    <VisualState Name="Unfocused"/>

</VisualStateGroup>

</VisualStateManager.VisualStateGroups>

<!-- Content -->
<Grid Name="grdContainer" Margin="{TemplateBinding Padding }"
    Background="Transparent"
    IsEnabled="{TemplateBinding IsEnabled }">

    <Grid.ColumnDefinitions >
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="*/>
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition Width="Auto" />
        <ColumnDefinition Width="Auto" />
    </Grid.ColumnDefinitions>

    <Button Name="PART_First"
        Width="{TemplateBinding ButtonsWidth}"
        Style="{TemplateBinding ButtonsStyle }">
        <Button.Content>
            <Viewbox Width="16" Height="16">
                <Rectangle Width="16" Height="16">
                    <Rectangle.Fill>

```

```

        <DrawingBrush Stretch="Uniform">
            <DrawingBrush.Drawing>
                <DrawingGroup>
                    <DrawingGroup.Children>
                        <GeometryDrawing Brush="#F
F1A0A9D" Geometry="F1 M 226.988,239.161L 227.478,13.6956L 28.7864,131.684L 226.988,239.161 Z
">
                            <GeometryDrawing.Pen>
                                <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
                            </GeometryDrawing.Pen>
                        </GeometryDrawing>
                    <GeometryDrawing Brush="#F
F1A0A9D" Geometry="F1 M 21.8821,12.9033L 21.8821,239.391">
                            <GeometryDrawing.Pen>
                                <Pen Thickness="15
" LineJoin="Round" Brush="#FF1A0A9D"/>
                            </GeometryDrawing.Pen>
                        </GeometryDrawing>
                    </DrawingGroup.Children>
                </DrawingGroup>
            </DrawingBrush.Drawing>
        </DrawingBrush>
    </Rectangle.Fill>
</Rectangle>
</Viewbox>
</Button.Content>
</Button>

<Button Name="PART_Previous" Grid.Column="1" Margin="2,0,0,0"
Width="{TemplateBinding ButtonsWidth}"
Style="{TemplateBinding ButtonsStyle }">
    <Button.Content >
        <Viewbox Width="16" Height="16">
            <Rectangle Width="16" Height="16">
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#F
F1A0A9D" Geometry="F1 M 198.701,225.966L 199.191,0.499995L 0.5,118.489L 198.701,225.966 Z ">
                                            <GeometryDrawing.Pen>
                                                <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
                                            </GeometryDrawing.Pen>
                                        </GeometryDrawing>
                                    </DrawingGroup.Children>
                                </DrawingGroup>
                            </DrawingBrush.Drawing>
                        </DrawingBrush>
                    </Rectangle.Fill>
                </Rectangle>
            </Viewbox>
        </Button.Content>
    </Button>

<Label Name="PART_InfoText" Grid.Column="2"
VerticalAlignment="Center"
HorizontalAlignment="Stretch"

```

```

        HorizontalContentAlignment="Center"
        VerticalContentAlignment="Center"
        Content="{TemplateBinding InfoText }"
        Style="{TemplateBinding InfoTextStyle }">

</Label>

<Button Name="PART_Next" Grid.Column ="3"
        Width="{TemplateBinding ButtonsWidth}"
        Style="{TemplateBinding ButtonsStyle }">
    <Button.Content >
        <Viewbox Width="16" Height="16">
            <Rectangle Width="16" Height="16">
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#F
F1A0A9D" Geometry="F1 M 0.990074,225.966L 0.499996,0.499995L 199.191,118.489L
0.990074,225.966 Z ">
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>
        </Viewbox>
    </Button.Content>
</Button>

<Button Name="PART_Last" Grid.Column="4" Margin="2,0,0,0"
        Width="{TemplateBinding ButtonsWidth}"
        Style="{TemplateBinding ButtonsStyle }">
    <Button.Content >
        <Viewbox Width="16" Height="16">
            <Rectangle Width="16" Height="16">
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#F
F1A0A9D" Geometry="F1 M 22.3722,239.161L 21.8821,13.6956L 220.573,131.685L 22.3722,239.161 Z
">
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>
        </Viewbox>
    </Button.Content>
</Button>

<Button Name="PART_Next" Grid.Column ="3"
        Width="{TemplateBinding ButtonsWidth}"
        Style="{TemplateBinding ButtonsStyle }">
    <Button.Content >
        <Viewbox Width="16" Height="16">
            <Rectangle Width="16" Height="16">
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#F
F1A0A9D" Geometry="F1 M 227.478,12.9033L 227.478,239.391">
                                        <GeometryDrawing.Pen>
                                            <Pen Thickness="15
" LineJoin="Round" Brush="#FF1A0A9D"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>
        </Viewbox>
    </Button.Content>
</Button>

```

```

        </GeometryDrawing.Pen>
        </GeometryDrawing>
        </DrawingGroup.Children>
    </DrawingGroup>
    </DrawingBrush.Drawing>
</DrawingBrush>
    </Rectangle.Fill>
</Rectangle>

    </Viewbox>
</Button.Content>

</Button>

<!-- Transactions -->

    <Button Name="PART_AddNew" Grid.Column="5" Margin="5,0,0,0"
        Width="{TemplateBinding ButtonsWidth}"
        Style="{TemplateBinding ButtonsStyle}"
        Visibility="{TemplateBinding ShowAddNew, Converter={StaticResource booleanToVisibilityConverter}}">
        <Button.Content >
            <Viewbox Width="16" Height="16">
                <Rectangle Width="16" Height="16">
                    <Rectangle.Fill>
                        <DrawingBrush Stretch="Uniform">
                            <DrawingBrush.Drawing>
                                <DrawingGroup>
                                    <DrawingGroup.Children>
                                        <GeometryDrawing Brush="#F0CA80C" Geometry="F1 M 96.8821,33.3033L 159.549,33.3033L 159.549,95.6181L 223.86,95.2348L 224.234,157.9L 159.549,158.286L 159.549,224.303L 96.8821,224.303L 96.8821,158.659L 33.2372,159.039L 32.8637,96.373L 96.8821,95.9915L 96.8821,33.3033 Z " >
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Round" Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

            </Viewbox>
        </Button.Content>

    </Button>

    <Button Name="PART_Edit" Grid.Column="6" Margin="2,0,0,0"
        Width="{TemplateBinding ButtonsWidth}"
        Style="{TemplateBinding ButtonsStyle}"
        Visibility="{TemplateBinding ShowEdit, Converter={StaticResource booleanToVisibilityConverter}}">
        <Button.Content >
            <Viewbox Width="16" Height="16">
                <Rectangle Width="16" Height="16">
                    <Rectangle.Fill>
                        <DrawingBrush Stretch="Uniform">
                            <DrawingBrush.Drawing>

```



```
<DrawingGroup>
  <DrawingGroup.Children>
    <GeometryDrawing Brush="#F
F0CA80C" Geometry="F1 M 225.329,72.0836L 182.621,28.3352L 64.5521,143.594L 107.259,187.343L
225.329,72.0836 Z ">
      <GeometryDrawing.Pen>
        <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
      </GeometryDrawing.Pen>
    </GeometryDrawing>
    <GeometryDrawing Brush="#F
F096909" Geometry="F1 M 64.8482,142.72L 45.5819,200.874L 107.215,186.97L 64.8482,142.72 Z ">
      <GeometryDrawing.Pen>
        <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
      </GeometryDrawing.Pen>
    </GeometryDrawing>
    <GeometryDrawing Geometry=
"F1 M 31.8821,210.637C 34.4125,209.394 37.2681,207.701 40.0299,208.303C 42.5489,208.853
44.9339,210.024 47.4987,210.303C 49.3017,210.5 51.1275,209.773 52.9305,209.97C
58.2215,210.547 63.5957,211.214 68.8866,210.637C 75.1376,209.955 81.2949,208.529
87.5586,207.97C 92.63,207.517 97.7643,208.423 102.836,207.97C 104.867,207.789
106.628,206.456 108.607,205.97C 112.849,204.929 117.157,204.064 121.508,203.637C
128.83,202.918 136.252,203.689 143.575,202.97C 145.502,202.781 147.414,203.501
149.346,203.637C 153.536,203.931 157.832,203.636 161.907,204.637C 165.422,205.499
168.808,207.303 172.431,207.303">
      <GeometryDrawing.Pen>
        <Pen Thickness="3"
LineJoin="Round" Brush="#FF075707"/>
      </GeometryDrawing.Pen>
    </GeometryDrawing>
    <GeometryDrawing Geometry=
"F1 M 30.5487,216.97C 35.2699,215.783 40.2666,216.303 45.1331,216.303C 50.8795,216.303
56.6578,215.332 62.3691,215.97C 64.9327,216.256 67.4262,217.045 69.9927,217.303C
75.3617,217.843 80.5487,213.63 85.9029,214.303C 90.8668,214.927 95.8548,216.594
100.819,215.97C 103.27,215.662 105.65,214.862 108.111,214.637C 112.983,214.191
118.171,214.843 122.695,212.97C 124.936,212.042 126.916,210.239 129.324,209.97C
132.955,209.564 136.619,209.565 140.263,209.303C 144.236,209.018 148.212,208.637
152.195,208.637C 153.636,208.637 155.098,207.989 156.504,208.303C 161.051,209.319
165.705,210.826 169.431,213.637">
      <GeometryDrawing.Pen>
        <Pen Thickness="3"
LineJoin="Round" Brush="#FF075707"/>
      </GeometryDrawing.Pen>
    </GeometryDrawing>
    <GeometryDrawing Geometry=
"F1 M 32.2154,224.637C 37.511,224.637 42.8109,226.268 48.0801,225.637C 50.912,225.298
53.3277,222.726 56.1516,222.303C 60.2984,221.683 64.5055,222.658 68.6764,222.97C
72.8689,223.284 77.033,221.927 81.2012,221.303C 83.2349,220.999 85.2833,221.748
87.3244,221.97C 89.1017,222.164 90.839,221.206 92.6127,220.97C 96.2135,220.491
99.8551,221.304 103.467,221.637C 108.741,222.122 114.065,221.397 119.332,221.97C
122.026,222.263 124.78,223.756 127.404,222.97C 130.48,222.049 133.383,220.431 136.31,218.97C
137.854,218.2 139.712,219.187 141.32,218.637C 142.039,218.391 142.587,217.674
143.268,217.303C 143.617,217.114 144.002,217.021 144.382,216.97C 148.217,216.46
151.964,218.73 155.793,219.303C 156.99,219.483 158.224,219.052 159.411,219.303C
162.762,220.012 166.029,221.637 169.431,221.637">
      <GeometryDrawing.Pen>
        <Pen Thickness="3"
LineJoin="Round" Brush="#FF075707"/>
      </GeometryDrawing.Pen>
    </GeometryDrawing>
  </DrawingGroup.Children>
</DrawingGroup>
```

```

        </DrawingGroup>
        </DrawingBrush.Drawing>
    </DrawingBrush>
    </Rectangle.Fill>
</Rectangle>

    </Viewbox>
</Button.Content>

</Button>

<Button Name="PART_Commit" Grid.Column="7" Margin="2,0,0,0"
    Width="{TemplateBinding ButtonsWidth}"
    Style="{TemplateBinding ButtonsStyle }"
    Visibility="{TemplateBinding ShowCommit, Converter={St
aticResource booleanToVisibilityConverter}}">
    <Button.Content >
        <Viewbox Width="16" Height="16">
            <Rectangle Width="16" Height="16">
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#F
F0CA80C" Geometry="F1 M 23.9679,153.275L 61.1001,104.459L 103.92,137.03L 183.821,33.6276L
232.353,71.1296L 144.023,185.44L 144.003,185.423L 115.497,222.898L 23.9679,153.275 Z " >
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Rou
nd" Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

        </Viewbox>
    </Button.Content>

</Button>

<Button Name="PART_Remove" Grid.Column="8" Margin="2,0,0,0"
    Width="{TemplateBinding ButtonsWidth}"
    Style="{TemplateBinding ButtonsStyle }"
    Visibility="{TemplateBinding ShowRemove, Converter={St
aticResource booleanToVisibilityConverter}}">
    <Button.Content >
        <Viewbox Width="16" Height="16">
            <Rectangle Width="16" Height="16">
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#F
FD54545" Geometry="F1 M 22.0573,77.8647L 64.1859,33.2894L 126.777,92.4449L 188.378,33.7767L
230.677,78.1901L 171.414,134.632L 227.235,187.389L 185.106,231.964L 126.939,176.99L
68.2197,232.913L 25.9206,188.5L 82.302,134.803L 22.0573,77.8647 Z " >
                                        <GeometryDrawing.Pen>

```

```
nd" Brush="#FF000000"/>
<Pen LineJoin="Rou
</GeometryDrawing.Pen>
</GeometryDrawing>
</DrawingGroup.Children>
</DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>
</Viewbox>
</Button.Content>
</Button>
</Grid>
</Border>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
```

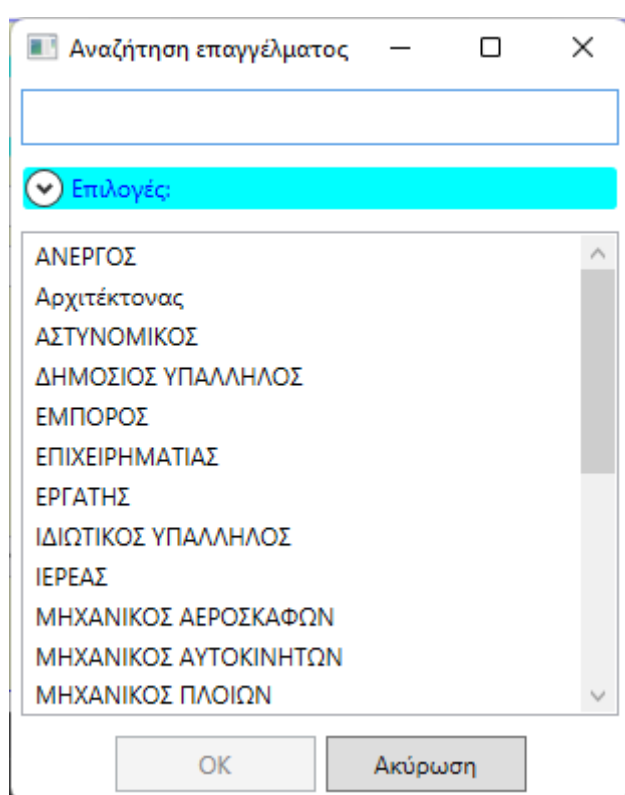
Για τη σχεδίαση των buttons χρησιμοποίησα το Microsoft Expression Design 4, το οποίο επιτρέπει να εξάγουμε το αποτέλεσμα σε XAML κώδικα.



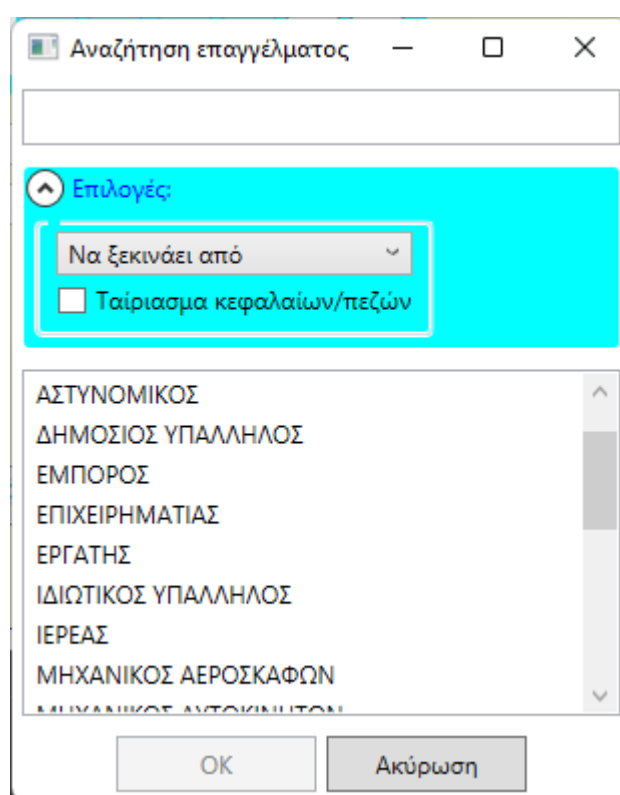
## FindXPress

Ένα **Control** που παρέχει μία φόρμα για τη γρήγορη αναζήτηση κι επιλογή κάποιου δεδομένου (κειμένου) σε μία λίστα με μοναδικές (unique) τιμές. Η αναζήτηση μπορεί να διαμορφωθεί με βάση κάποιες επιλογές όπως τρόπος (προσανατολισμός) αναζήτησης (π.χ. να ξεκινάει με, να περιέχει, να ισούται κλπ) καθώς και αν θα γίνεται διάκριση κεφαλαίων-πεζών (match case).

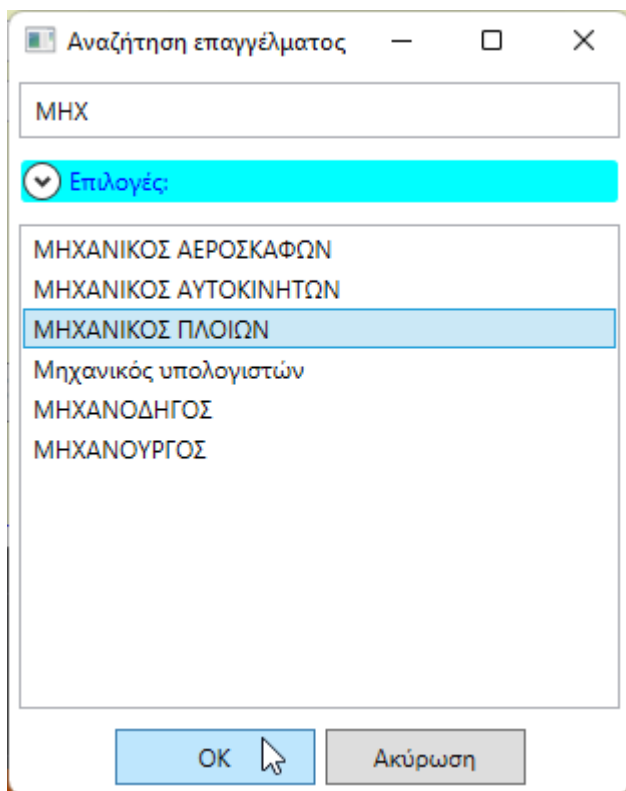
Το control κανονικά θα το προσθέσετε μόνο του σε ένα Window element, όπως φαίνεται και στις παρακάτω εικόνες:



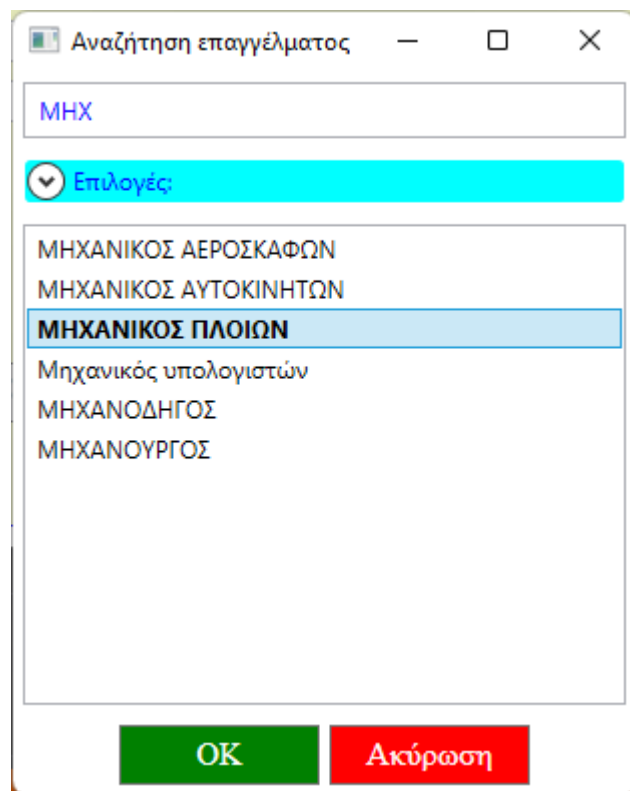
Η default φόρμα



Η default φόρμα με άνοιγμα των επιλογών αναζήτησης



Γράφοντας μία τιμή ή τα αρχικά μίας τιμής αναζήτησης η λίστα προσαρμόζεται ανάλογα. Αν επιλέξουμε κάποια τιμή από τη λίστα τότε ενεργοποιείται το OK.



**Σχεδόν όλα τα συστατικά του μπορούν να προσαρμοστούν με styles.** Π.χ. το TextBox αναζήτησης, η λίστα, τα buttons OK και Ακύρωση, ακόμα και το ComboBox και CheckBox του τμήματος Επιλογές. Επίσης, **μπορούν να αλλάξουν και τα λεκτικά του control.**

## Σύνταξη:

### VB:

```
<TemplateVisualState(Name:="Normal", GroupName:="CommonStates"),
TemplateVisualState(Name:="MouseOver", GroupName:="CommonStates"),
TemplateVisualState(Name:="Disabled", GroupName:="CommonStates"),
TemplateVisualState(Name:="Focused", GroupName:="FocusStates"),
TemplateVisualState(Name:="Unfocused", GroupName:="FocusStates"),
TemplatePart(Name:="PART_Value", Type:=GetType(TextBox)),
TemplatePart(Name:="PART_FindOrientation", Type:=GetType(ComboBox)),
TemplatePart(Name:="PART_FindMatchCase", Type:=GetType(CheckBox)),
TemplatePart(Name:="PART_ListResult", Type:=GetType(ListBox)),
TemplatePart(Name:="PART_OK", Type:=GetType(Button)),
TemplatePart(Name:="PART_Cancel", Type:=GetType(Button))>
Public Class FindXPress
    Inherits Control
```

### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

Χρήση:

```
<zeus:FindXPress ... />
```

## Παρατηρήσεις:

Πρόκειται για ένα πολύ ευέλικτο control που παρέχει στον χρήστη μία φόρμα για να κάνει γρήγορη αναζήτηση μίας τιμής κειμένου (ή τα αρχικά της) σε μία λίστα που περιέχει μοναδικές τιμές (unique).

Σχεδόν όλα τα γραφικά στοιχεία του control καθώς και τα λεκτικά μπορούν να παραμετροποιηθούν μέσω των αντίστοιχων ιδιοτήτων. Για παράδειγμα, τα λεκτικά των buttons OK και Ακύρωση μπορούν να αλλάξουν σε Select και Cancel. Επίσης, μπορούμε να αλλάξουμε και τα styles των buttons αυτών.

Η βασική ιδιότητα του FindXPress είναι η CView, τύπου CollectionView. Δηλαδή, δέχεται μία συλλογή δεδομένων, η οποία γεμίζει το ListBox του control. Σε αυτήν τη λίστα θα γίνει η αναζήτηση (φιλτράρισμα). Οι sub-classes αυτής είναι δύο: ListCollectionView όταν η συλλογή υλοποιεί το IList ή ICollection interface (για παράδειγμα μία ObservableCollection) και BindingListCollectionView όταν τα δεδομένα προέρχονται από ADO.NET objects (DataTable, DataView). Συνεπώς, σε όποια πηγή έχουμε δεδομένα θα πρέπει να τα μεταφέρουμε σε ένα αντικείμενο μίας από τις παραπάνω sub-classes, ώστε να γίνει η αναζήτηση και να το θέσουμε στην ιδιότητα CView.

Εκείνο που πρέπει να προσέξουμε είναι ότι η συλλογή πρέπει να έχει αντικείμενα τύπου String.

Όταν ο χρήστης επιλέξει μία τιμή από τη λίστα τότε ενεργοποιείται το button OK και η επιλεγμένη τιμή τίθεται στην ιδιότητα ReturnValue την οποία θα εκμεταλευτούμε περαιτέρω.

## Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε μέσα σε ένα παράθυρο (Window element) μόνο του ένα FindXPress. Στο τμήμα Window.Resources, θα δούμε διάφορα custom styles που θα εφαρμοστούν σε αντίστοιχες ιδιότητες του FindXpress, ώστε να το παραμετροποιήσουμε σε εμφάνιση.

```
<Window x:Class="FindValue"
  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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
  mc:Ignorable="d"
  Title="Αναζήτηση επαγγέλματος" Height="400" Width="300"
  Loaded="Window_Loaded" Closing="Window_Closing">
  <Window.Resources>
    <!--To style για το κουτί αναζήτησης-->
```

```
<Style x:Key="ValueTextBoxStyle" TargetType="{x:Type TextBox }">
    <Setter Property="Foreground" Value="Blue"/>
    <Setter Property="FontWeight" Value="Normal"/>
    <Style.Triggers >
        <Trigger Property="IsKeyboardFocused" Value="True">
            <Setter Property="Background" Value="Yellow"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!--To style για το ComboBoxFindOrientationItem-->
<Style x:Key="FindOrientationComboBoxItemStyle" TargetType="{x:Type ComboBoxItem }">
    <Style.Triggers >
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="12"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Blue"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!--To style θα εφαρμοστεί στο ComboBoxItem του FindOrientation ComboBox-->
<Style x:Key="FindOrientationComboBoxStyle" TargetType="{x:Type ComboBox }">
    <Setter Property="ItemContainerStyle"
        Value="{StaticResource FindOrientationComboBoxItemStyle }"/>
</Style>

<!--To style για το ListBoxResultItem-->
<Style x:Key="ResultListBoxItemStyle" TargetType="{x:Type ListBoxItem }">
    <Style.Triggers >
        <Trigger Property="IsSelected" Value="True">
            <Setter Property="FontWeight" Value="Bold"/>
        </Trigger>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="12"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Orange"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!--To style θα εφαρμοστεί στο ListBoxItem του Result ListBox-->
<Style x:Key="ResultListBoxStyle" TargetType="{x:Type ListBox }">
    <Setter Property="ItemContainerStyle"
        Value="{StaticResource ResultListBoxItemStyle }"/>
</Style>

<!--A style for the OK button-->
<Style x:Key="OKButtonStyle" TargetType="{x:Type Button}">
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Green"/>
    <Setter Property="FontSize" Value="16"/>
</Style>
```

```
        <Setter Property="FontFamily" Value ="Times New Roman"/>
    </Style>

    <!--A style for the Cancel button-->
    <Style x:Key="CancelButtonStyle" TargetType="{x:Type Button}">

        <Setter Property="Foreground" Value ="White"/>
        <Setter Property="Background" Value ="Red"/>
        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontFamily" Value ="Times New Roman"/>

    </Style>

</Window.Resources>

<Grid>
    <z:FindXPress Name="findXpress"
        TextBoxValueStyle="{StaticResource ValueTextBoxStyle}"
        ComboBoxFindOrientationStyle="{StaticResource
            FindOrientationComboBoxStyle}"
        ListBoxResultStyle="{StaticResource ResultListBoxStyle}"
        ButtonOKStyle="{StaticResource OKButtonStyle}"
        ButtonCancelStyle="{StaticResource CancelButtonStyle}"
        OKClicked="findXpress_OKClicked"
        CancelClicked="findXpress_CancelClicked"/>
</Grid>

</Window>
```

Στον VB κώδικα:

```
Imports Zeus.WPF.Controls.DataControls

Public Class FindValue

    Private professionsView As ListCollectionView

    Public Property ReturnValue As String

    Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)

        findXpress.CView = professionsView
        findXpress.CView.MoveCurrentTo(Nothing)

    End Sub

    Private Sub Window_Closing(sender As Object, _
        e As ComponentModel.CancelEventArgs)

        findXpress.CView.Filter = Nothing

    End Sub

    Private Sub findXpress_OKClicked(sender As Object, e As RoutedEventArgs)

        Dim fndXpress As FindXPress = TryCast(sender, FindXPress)

        If fndXpress IsNot Nothing Then
```



```
        ReturnValue = fndXpress.ReturnValue
        fndXpress.CView.Filter = Nothing
        Me.Close()

    End If

End Sub

Private Sub findXpress_CancelClicked(sender As Object, e As RoutedEventArgs)

    Dim fndXpress As FindXPress = TryCast(sender, FindXPress)

    If fndXpress IsNot Nothing Then

        fndXpress.CView.Filter = Nothing
        Me.Close()

    End If

End Sub

Public Sub New(CView As ListCollectionView)

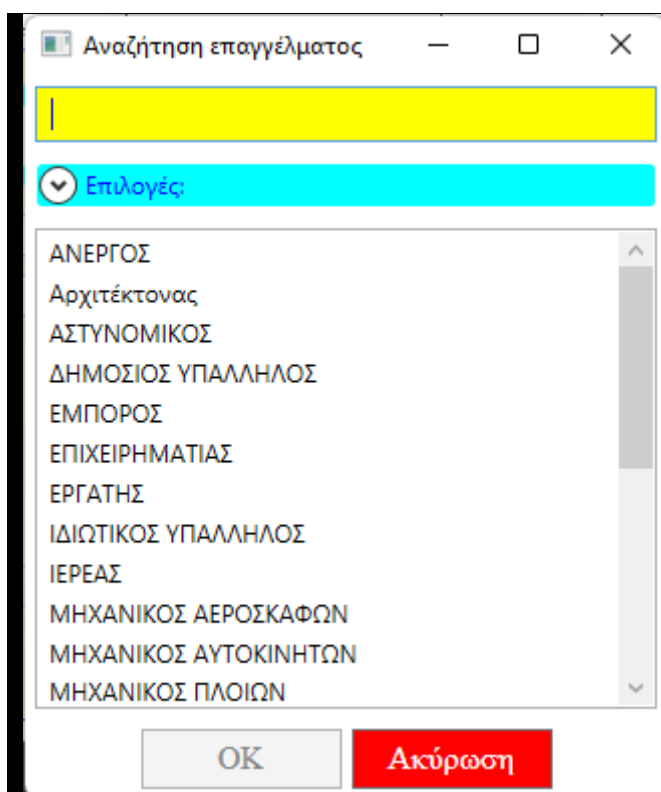
    ' This call is required by the designer.
    InitializeComponent()

    ' Add any initialization after the InitializeComponent() call.
    professionsView = CView

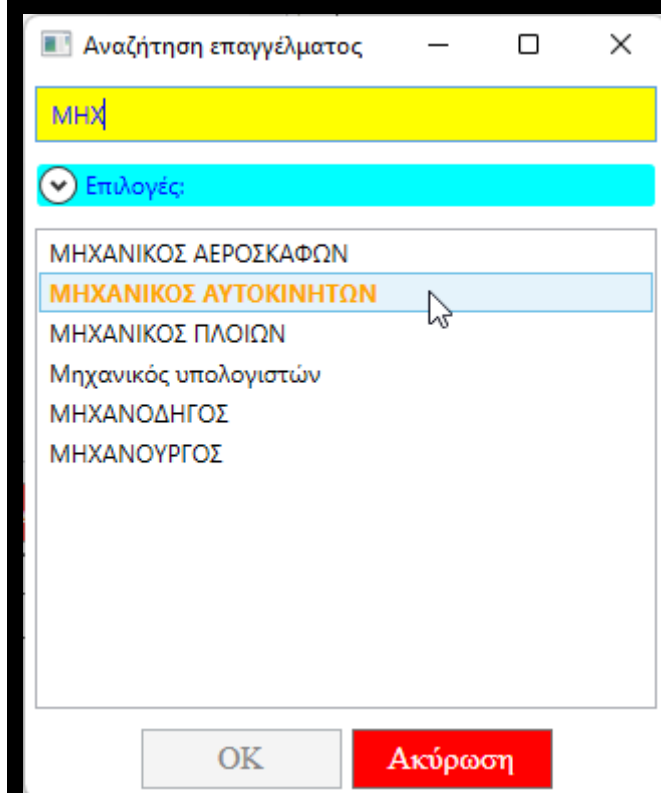
End Sub

End Class
```

Στις παρακάτω εικόνες φαίνονται δύο δείγματα:



Όταν ανοίγει το παράθυρο



Γράφοντας ΜΗΧ οι τιμές της λίστας περιορίζονται σε αυτές που ξεκινούν από την τιμή αυτή. Οι default επιλογές αναζήτησης είναι "Να ξεκινάει με" και να μην γίνεται διάκριση κεφαλαίων/πεζών.

Επίσης, φαίνεται και το style όταν το ποντίκι ίπταται πάνω από μία τιμή (πορτοκαλί έντονο).

## Απαριθμήσεις

Όνομα	Περιγραφή
FindOrientationEnum	Περιλαμβάνει τις τιμές Προσανατολισμού αναζήτησης (π.χ StartsWith).
LanguageForUIEnum	Περιλαμβάνει τις τιμές για την γλώσσα που θα χρησιμοποιηθεί στα κείμενα του UI (π.χ Greek, English).

## FindOrientationEnum

Περιλαμβάνει τιμές Προσανατολισμού αναζήτησης.

**Σύνταξη:**

**VB:**

```
Public Enum FindOrientationEnum
```

**Μέλη:**

Τιμή	Περιγραφή
StartsWith	Να ξεκινάει με το λεκτικό
EndsWith	Να τελειώνει με το λεκτικό
Contains	Να περιλαμβάνει το λεκτικό
Equal	Να ισούται με το λεκτικό

## LanguageForUIEnum

Περιλαμβάνει τις τιμές για την γλώσσα που θα χρησιμοποιηθεί στα κείμενα του UI.

### Σύνταξη:

**VB:**

```
Public Enum LanguageForUIEnum
```

**Μέλη:**

Τιμή	Περιγραφή
Greek	Τα κείμενα του UI θα είναι στα Ελληνικά
English	Τα κείμενα του UI θα είναι στα Αγγλικά

## Ιδιότητες

Όνομα	Περιγραφή
ButtonCancelStyle	Καθορίζει το <b>στυλ</b> για το <b>button "Ακύρωση"</b> , τύπου <b>Style</b> .
ButtonCancelText	Καθορίζει το <b>κείμενο (λεκτικό)</b> για το <b>button "Ακύρωση"</b> , τύπου <b>String</b> .
ButtonOKStyle	Καθορίζει το <b>στυλ</b> για το <b>button "OK"</b> , τύπου <b>Style</b> .
ButtonOKText	Καθορίζει το <b>κείμενο (λεκτικό)</b> για το <b>button "OK"</b> , τύπου <b>String</b> .
ButtonsOKCancelWidth	Καθορίζει το <b>πλάτος, σε pixels, των buttons "OK" και "Ακύρωση"</b> . Είναι τύπου <b>Double</b> .
CheckBoxFindMatchCaseStyle	Καθορίζει το <b>στυλ</b> για το <b>checkbox "Match Case"</b> , τύπου <b>Style</b> .
ComboBoxFindOrientationStyle	Καθορίζει το <b>στυλ</b> για το <b>combobox "Find Orientation"</b> , τύπου <b>Style</b> .
CView	Καθορίζει την <b>πηγή δεδομένων</b> για τη <b>λίστα</b> του control, τύπου <b>CollectionView</b> .
FindMatchCase	Καθορίζει <b>αν θα γίνεται διάκριση κεφαλαίων/πεζών κατά την αναζήτηση</b> . Είναι τύπου <b>Boolean</b> .
FindMatchCaseCaption	Καθορίζει το <b>κείμενο-λεκτικό (caption)</b> για το <b>checkbox "Match Case"</b> , τύπου <b>String</b> .
FindOptionsCaption	Καθορίζει το <b>κείμενο (caption)</b> για την <b>επικεφαλίδα "Επιλογές"</b> . Είναι τύπου <b>String</b> .
FindOrientation	Καθορίζει τον <b>προσανατολισμό της αναζήτησης</b> , τύπου <b>FindOrientationEnum</b> .

FindOrientationTextContains	Καθορίζει το κείμενο για την τιμή "Contains" στο combobox "Find Orientation". Είναι τύπου String.
FindOrientationTextEndsWith	Καθορίζει το κείμενο για την τιμή "Ends with" στο combobox "Find Orientation". Είναι τύπου String.
FindOrientationTextEqualTo	Καθορίζει το κείμενο για την τιμή "Equal to" στο combobox "Find Orientation". Είναι τύπου String.
FindOrientationTextStartsWith	Καθορίζει το κείμενο για την τιμή "Starts with" στο combobox "Find Orientation". Είναι τύπου String.
LanguageForUI	Καθορίζει το τη γλώσσα που θα χρησιμοποιηθεί στα κείμενα του UI. Είναι τύπου LanguageForUIEnum.
ListBoxResultStyle	Καθορίζει το στυλ για το listbox του αποτελέσματος της αναζήτησης (λίστα), τύπου Style.
ReturnValue	Επιστρέφει την τιμή όταν ο χρήστης επιλέξει μία από τη λίστα αποτελέσματος και πατήσει το OK. Είναι τύπου String.
TextBoxValueStyle	Καθορίζει το στυλ για το textbox όπου ο χρήστης πληκτρολογεί ολόκληρη ή ένα μέρος της τιμής αναζήτησης. Είναι τύπου Style.
ToolTipTextForValue	Καθορίζει το κείμενο του tooltip για το textbox αναζήτησης. Είναι τύπου String.
ShowToolTipForValueProperty	Καθορίζει αν θα εμφανίζεται το tooltip για το textbox αναζήτησης. Είναι τύπου Boolean.

## ButtonCancelStyle

Καθορίζει το **στυλ για το button "Ακύρωση"**, τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property ButtonCancelStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Ακύρωση". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonCancelStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"OKButtonStyle"** και το εφαρμόζουμε στο **button "OK"** και ένα δεύτερο **resource style** με όνομα **"CancelButtonStyle"** και το εφαρμόζουμε στο **button "Ακύρωση"**.

#### XAML:

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300"
    Loaded="Window_Loaded" Closing="Window_Closing">

    <Window.Resources>

        ...

        <!-- A style for the OK button-->
        <Style x:Key="OKButtonStyle" TargetType="{x:Type Button}">

            <Setter Property="Foreground" Value="White"/>
            <Setter Property="Background" Value="Green"/>
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontFamily" Value="Times New Roman"/>
```



```
</Style>

<!-- A style for the Cancel button-->
<Style x:Key="CancelButtonStyle" TargetType="{x:Type Button}">

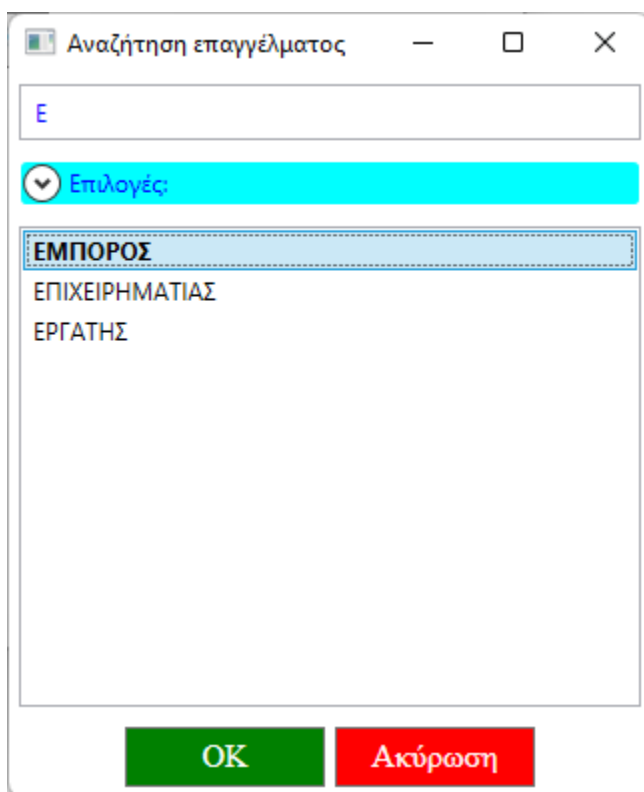
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

</Window.Resources>

<Grid>
    <z:FindXpress Name="findXpress"
        ButtonOKStyle="{StaticResource OKButtonStyle}"
        ButtonCancelStyle="{StaticResource CancelButtonStyle}"
        ... />
</Grid>

</Window>
```



## ButtonCancelText

Καθορίζει το κείμενο (λεκτικό) για το **button "Ακύρωση"**, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ButtonCancelText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button "Ακύρωση"**. Η default τιμή είναι "Ακύρωση".

### Dependency Property Information:

**Identifier field:** ButtonCancelTextProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

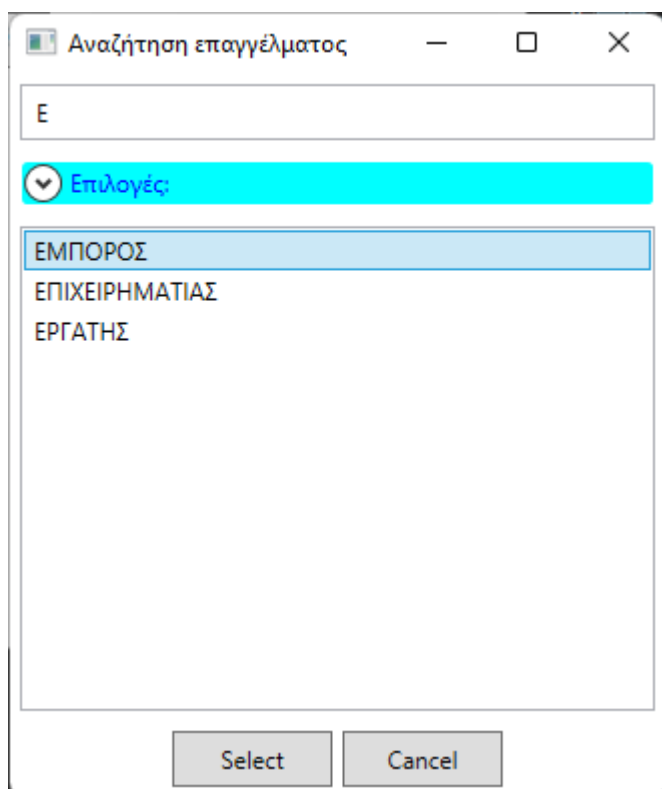
Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το **button "Ακύρωση"**, το λεκτικό **"Select"** για το **button "OK"** και θέτουμε το **πλάτος των buttons** αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση **Button**.

#### XAML:

```
<z:FindXPress Name="findXpress"
    ButtonCancelText="Cancel"
    ButtonOKText="Select"
    ButtonsOKCancelWidth="80" />
```

#### VB:

```
findXPress.ButtonOKText = "Select"
findXPress.ButtonCancelText = "Cancel"
findXPress.ButtonsOKCancelWidth = 80
```



## ButtonOKStyle

Καθορίζει το **στυλ για το button "OK"**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ButtonOKStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "OK". Το default style είναι αυτό ενός Button element.

### Dependency Property Information:

**Identifier field:** ButtonOKStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"OKButtonStyle"** και το εφαρμόζουμε στο **button "OK"** και ένα δεύτερο **resource style** με όνομα **"CancelButtonStyle"** και το εφαρμόζουμε στο **button "Ακύρωση"**.

**XAML:**

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

        <!-- A style for the OK button-->
        <Style x:Key="OKButtonStyle" TargetType="{x:Type Button}">

            <Setter Property="Foreground" Value="White"/>
            <Setter Property="Background" Value="Green"/>
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontFamily" Value="Times New Roman"/>

        </Style>
```

```
<!-- A style for the Cancel button-->
<Style x:Key="CancelButtonStyle" TargetType="{x:Type Button}">

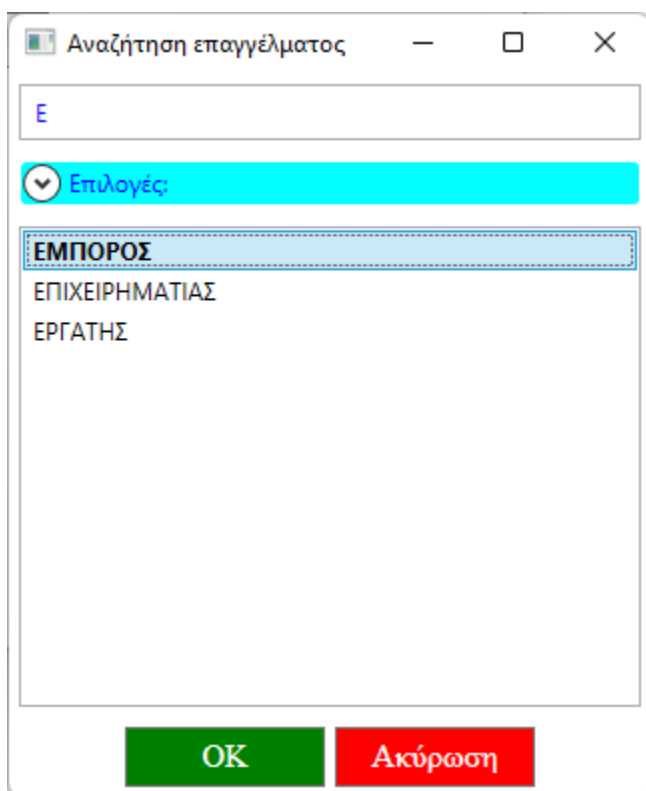
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="Background" Value="Red"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontFamily" Value="Times New Roman"/>

</Style>

</Window.Resources>

<Grid>
    <z:FindXpress Name="findXpress"
        ButtonOKStyle="{StaticResource OKButtonStyle}"
        ButtonCancelStyle="{StaticResource CancelButtonStyle}"
        ... />
</Grid>

</Window>
```



## ButtonOKText

Καθορίζει το κείμενο (λεκτικό) για το **button "OK"**, τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property ButtonOKText As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή **String** που θα αποτελεί το λεκτικό για το **button "OK"**. Η default τιμή είναι **"OK"**.

### Dependency Property Information:

**Identifier field:** `ButtonOKTextProperty`

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό **"Cancel"** για το **button "Ακύρωση"**, το λεκτικό **"Select"** για το **button "OK"** και θέτουμε το **πλάτος των buttons** αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση **Button**.

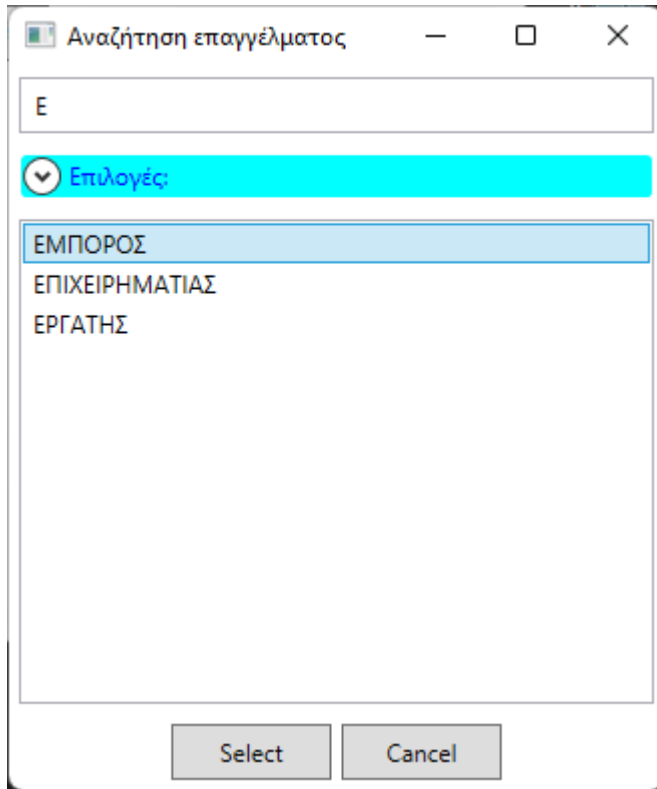
**XAML:**

```
<z:FindXPress Name="findXpress"  
    ButtonCancelText="Cancel"  
    ButtonOKText="Select"  
    ButtonsOKCancelWidth="80" />
```

**VB:**

```
findXPress.ButtonOKText = "Select"
```

```
findXPress.ButtonCancelText = "Cancel"  
findXPress.ButtonsOKCancelWidth = 80
```



## ButtonsOKCancelWidth

Καθορίζει το πλάτος, σε pixels, των buttons "OK" και "Ακύρωση". Είναι τύπου **Double**.

### Σύνταξη:

**VB:**

```
Public Property ButtonsOKCancelWidth As Double
```

**Τύπος: System.Double**

Προσδιορίζουμε μία τιμή Double που καθορίζει το πλάτος για τα buttons "OK" και "Ακύρωση". Η default τιμή είναι 100.

### Dependency Property Information:

**Identifier field:** ButtonsOKCancelWidthProperty

### Παρατηρήσεις:

Ο λόγος που υπάρχει αυτή η ιδιότητα είναι για να καθορίσουμε γρήγορα ένα ομοιόμορφο πλάτος για τα buttons "OK" και "Ακύρωση".

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "Cancel" για το button "Ακύρωση", το λεκτικό "Select" για το button "OK" και θέτουμε το πλάτος των buttons αυτών σε **80 pixels**. Να σημειώσουμε ότι έχουμε κρατήσει το default style, που προέρχεται από την κλάση Button.

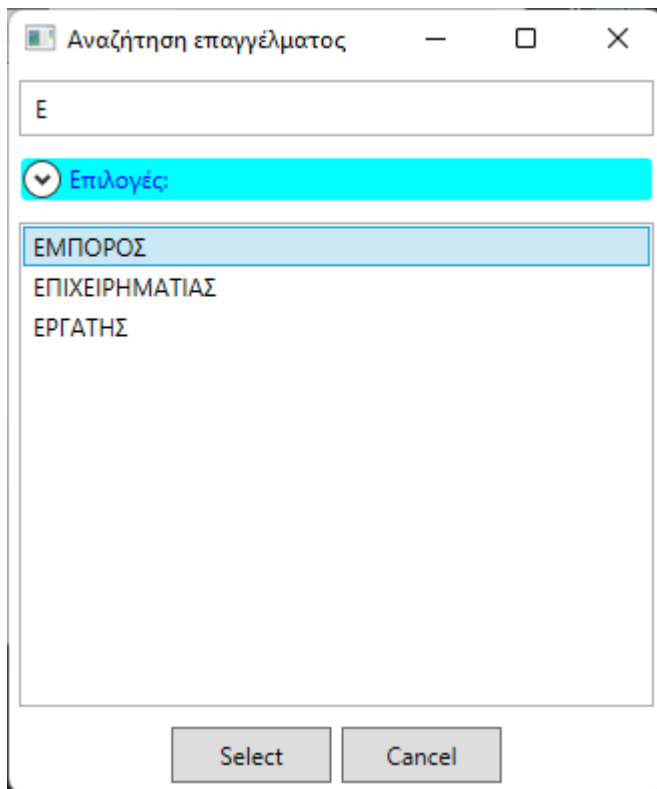
**XAML:**

```
<z:FindXPress Name="findXpress"
    ButtonCancelText="Cancel"
    ButtonOKText="Select"
    ButtonsOKCancelWidth="80" />
```

**VB:**

```
findXPress.ButtonOKText = "Select"
findXPress.ButtonCancelText = "Cancel"
findXPress.ButtonsOKCancelWidth = 80
```





## CheckBoxFindMatchCaseStyle

Καθορίζει το **στυλ για το checkbox "Match Case", τύπου Style**.

### Σύνταξη:

#### VB:

```
Public Property CheckBoxFindMatchCaseStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο checkbox "Match case". Το default style είναι αυτό ενός CheckBox element.

### Dependency Property Information:

**Identifier field:** CheckBoxFindMatchCaseStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"FindMatchCaseCheckBoxStyle"**, που κάνει τα γράμματα πλάγια, και το εφαρμόζουμε στο **checkbox "Match case"**.

#### XAML:

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

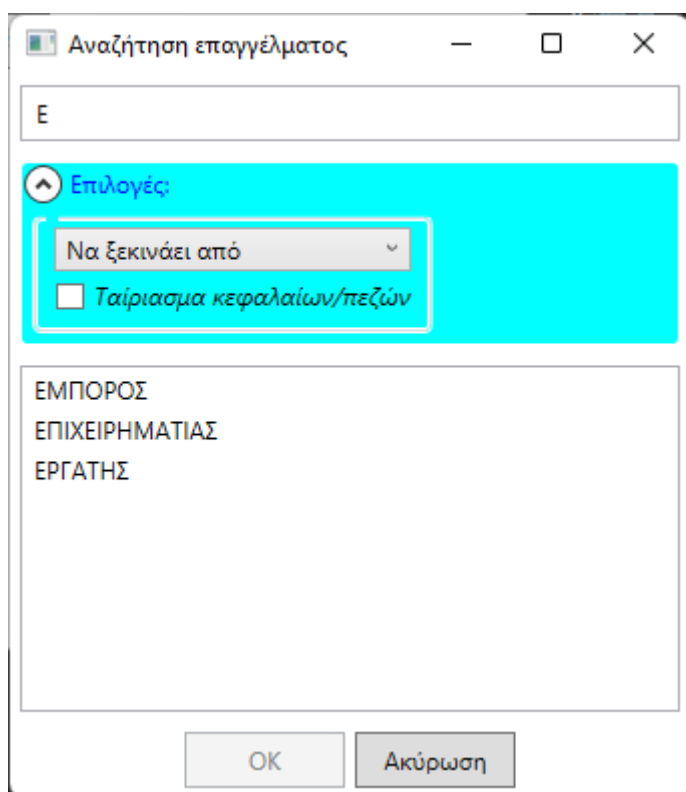
        <!-- To style θα εφαρμοστεί στο CheckBox "Match case"-->
        <Style x:Key="FindMatchCaseCheckBoxStyle" TargetType="{x:Type CheckBox}">

            <Setter Property="TextElement.FontStyle" Value="Italic"/>

        </Style>

    </Window.Resources>
```

```
<Grid>
  <z:FindXPress Name="findXpress"
    CheckBoxFindMatchCaseStyle="{StaticResource
      FindMatchCaseCheckBoxStyle }
    ... />
</Grid>
</Window>
```



## ComboBoxFindOrientationStyle

Καθορίζει το **στυλ για το combobox "Find Orientation", τύπου Style**.

### Σύνταξη:

#### VB:

```
Public Property ComboBoxFindOrientationStyle As Style
```

**Τύπος:** System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο combobox "Find Orientation". Το default style είναι αυτό ενός ComboBox element.

### Dependency Property Information:

**Identifier field:** ComboBoxFindOrientationStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"FindOrientationComboBoxItemStyle"**, που εφαρμόζει σε ένα combobox item μέγεθος 12, έντονη γραφή και χρώμα μπλε και το εφαρμόζουμε στο άλλο **resource style** με όνομα **"FindOrientationComboBoxStyle"** στην **ιδιότητα ItemContainerStyle**. Το τελευταίο style το θέτουμε στην **ιδιότητα ComboBoxFindOrientationStyle** του FindXPress control.

#### XAML:

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

        <!-- To style για το ComboBoxFindOrientationItem -->
        <Style x:Key="FindOrientationComboBoxItemStyle"
            TargetType="{x:Type ComboBoxItem}">

            <Style.Triggers >
```

```

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="12"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Blue"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!-- To style θα εφαρμοστεί στο ComboBoxItem του FindOrientation ComboBox-->
<Style x:Key="FindOrientationComboBoxStyle"
    TargetType="{x:Type ComboBox }">

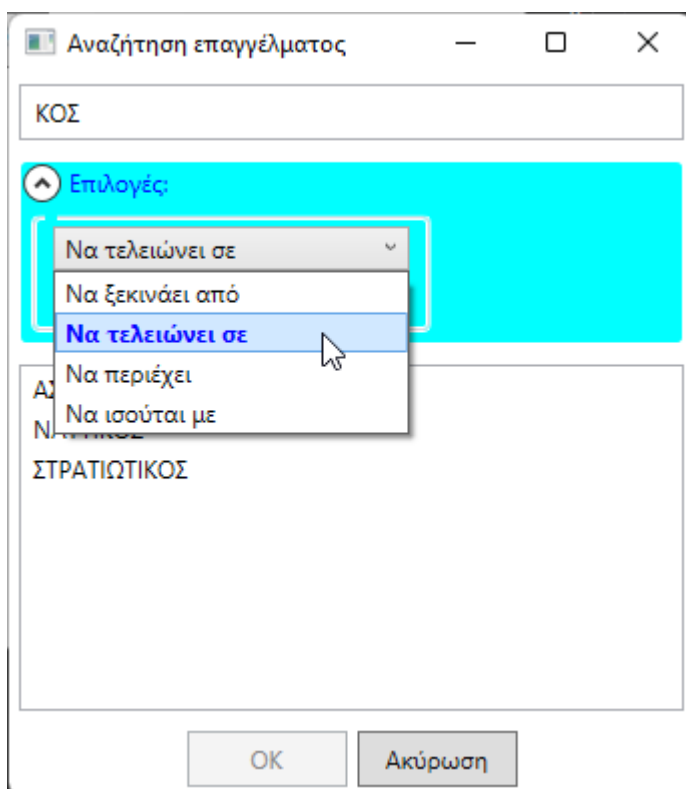
    <Setter Property="ItemContainerStyle"
        Value="{ StaticResource FindOrientationComboBoxItemStyle }"/>

</Style>

</Window.Resources>

<Grid>
    <z:FindXPress Name="findXpress"
        ComboBoxFindOrientationStyle="{StaticResource
            FindOrientationComboBoxStyle}"
        ... />
</Grid>
</Window>

```



## CView

Καθορίζει την πηγή δεδομένων για τη λίστα του control, τύπου **CollectionView**.

### Σύνταξη:

**VB:**

```
Public Property CView As CollectionView
```

**Τύπος:** **System.Windows.Data**

Προσδιορίζουμε ως πηγή δεδομένων ένα αντικείμενο **CollectionView** πάνω στο οποίο θα γίνει η αναζήτηση της τιμής.

### Dependency Property Information:

**Identifier field:** `CViewProperty`

### Παρατηρήσεις:

Ένα αντικείμενο **CollectionView** χρησιμοποιείται ως μία "άποψη" (view) για μία υποκείμενη πηγή δεδομένων. Αυτή η άποψη μπορεί να αφορά συγκεκριμένη ταξινόμηση, ομαδοποίηση και φιλτράρισμα των δεδομένων.

Συνήθως η sub-class του **CollectionView** που χρησιμοποιούμε είναι η **ListCollectionView** για συλλογές που υλοποιούν το **IIList** (custom collections). Η άλλη sub-class είναι η **BindingListCollectionView** για συλλογές που υλοποιούν το **IBindingList** και αφορούν υποκείμενα **DataTables** και **DataViews**.

**Κάτι σημαντικό:** Η υποκείμενη πηγή δεδομένων του **CView** (δηλαδή, το **ListCollectionView** ή το **BindingListCollectionView**) πρέπει να είναι συλλογή που περιέχει μοναδικές **String** τιμές (π.χ. **List(Of String)**). Έτσι, αν έχουμε μία συλλογή δεδομένων με σύνθετα αντικείμενα (π.χ. αντικείμενα που έχουν **ID** και **Description**, δείτε το παράδειγμα παρακάτω) πρέπει να απομονώσουμε μόνο την **String** ιδιότητα δημιουργώντας μία νέα λίστα τύπου **String**. Δείτε το παράδειγμα παρακάτω.

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε μία συλλογή **ObservableCollection(Of Profession)** με όνομα **obsProfessions**. Το κάθε αντικείμενο **Profession** αναπαριστά ένα επάγγελμα και περιέχει δύο ιδιότητες **ID** και **Description** (το πώς γεμίζει αυτή η συλλογή με data από μία Βάση Δεδομένων δεν μας ενδιαφέρει εδώ). Στη συνέχεια, προκειμένου να χρησιμοποιηθεί από την ιδιότητα **CView** του **FindXPress** control πρέπει να δημιουργηθεί μία άλλη λίστα που περιέχει μόνο τις περιγραφές (**Descriptions**) των επαγγελμάτων (χρησιμοποιούμε για αυτό

μία lambda έκφραση). Βάση αυτής της λίστας δημιουργείται το **ListCollectionView**, με όνομα **ProfessionsView**, και τίθεται στην ιδιότητα **CView**. Όλα αυτά γίνονται στον VB κώδικα.

#### XAML:

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

    </Window.Resources>

    <Grid>
        <z:FindXpress Name="findXpress"
            ... />
    </Grid>

</Window>
```

#### VB:

```
'Η lambda έκφραση για να πάρουμε μία νέα λίστα που περιέχει μόνο τα Descriptions από
'τη συλλογή obsProfessions.
Dim professions As List(Of String) = _
    obsProfessions.Select(Of String)(Function(x) x.Description).ToList

'Δημιούργησε το αντικείμενο ListCollectionView με βάση το υποκείμενο professions list.
Dim professionsView As ListCollectionView = _
    CType(CollectionViewSource.GetDefaultView(professions), ListCollectionView)

'Θέσε στην ιδιότητα CView το αντικείμενο ListCollectionView (που περιέχει μόνο τις
'μοναδικές περιγραφές των επαγγελμάτων).
findXpress.CView = professionsView
```

## FindMatchCase

Καθορίζει αν θα γίνεται διάκριση κεφαλαίων/πεζών κατά την αναζήτηση. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property FindMatchCase As Boolean
```

#### Τύπος: **System.Boolean**

Προσδιορίζουμε μία τιμή True/False που καθορίζει αν θα γίνεται διάκριση κεφαλαίων/πεζών κατά την αναζήτηση της τιμής. Η default τιμή είναι False.

### Dependency Property Information:

**Identifier field:** FindMatchCaseProperty

### Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε την ιδιότητα **FindMatchCase** σε **False**. Να σημειώσουμε ότι έχουμε θέσει ένα συγκεκριμένο custom style στην ιδιότητα **CheckBoxFindMatchCaseStyle** (πλάγια γράμματα).

#### XAML:

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

        <!-- To style θα εφαρμοστεί στο CheckBox "Match case"-->
        <Style x:Key="FindMatchCaseCheckBoxStyle" TargetType="{x:Type CheckBox}">

            <Setter Property="TextElement.FontStyle" Value="Italic"/>

        </Style>

    </Window.Resources>
```



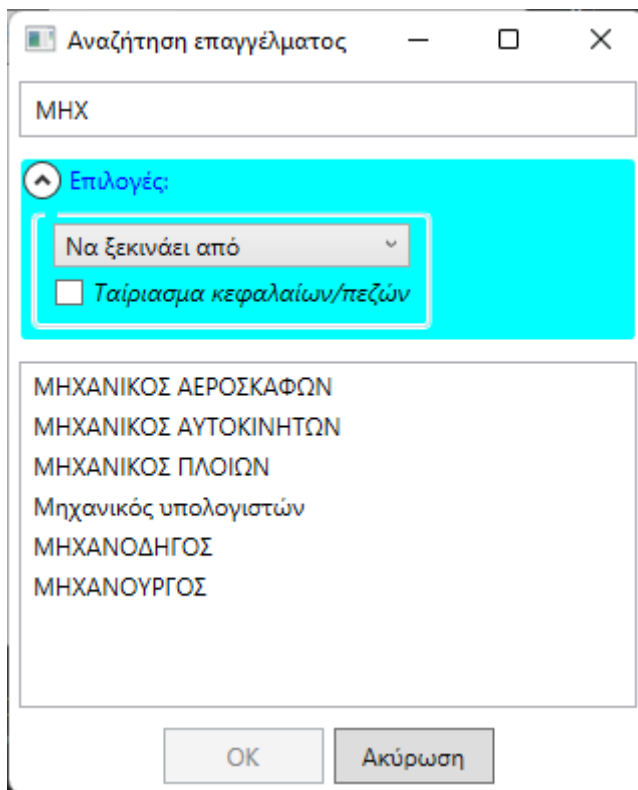
```

<Grid>
  <z:FindXpress Name="findXpress" ButtonsOKCancelWidth="80"
    FindMatchCase="False"
    CheckBoxFindMatchCaseStyle="{StaticResource
      FindMatchCaseCheckBoxStyle }
    ... />
</Grid>
</Window>

```

**VB:**

```
findXpress.FindMatchCase = False
```

**Παράδειγμα 2:**

Στο επόμενο παράδειγμα, καθορίζουμε την ιδιότητα **FindMatchCase** σε **True**. Να σημειώσουμε ότι έχουμε θέσει ένα συγκεκριμένο custom style στην ιδιότητα **CheckBoxFindMatchCaseStyle** (πλάγια γράμματα).

**XAML:**

```

<Window x:Class="FindValue"
  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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
xmlns:p="clr-namespace:DataControlsTestProject"
mc:Ignorable="d"
Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
Closing="Window_Closing">

<Window.Resources>

    ...

    <!-- Το style θα εφαρμοστεί στο CheckBox "Match case"-->
    <Style x:Key="FindMatchCaseCheckBoxStyle" TargetType="{x:Type CheckBox}">

        <Setter Property="TextElement.FontStyle" Value="Italic"/>

    </Style>

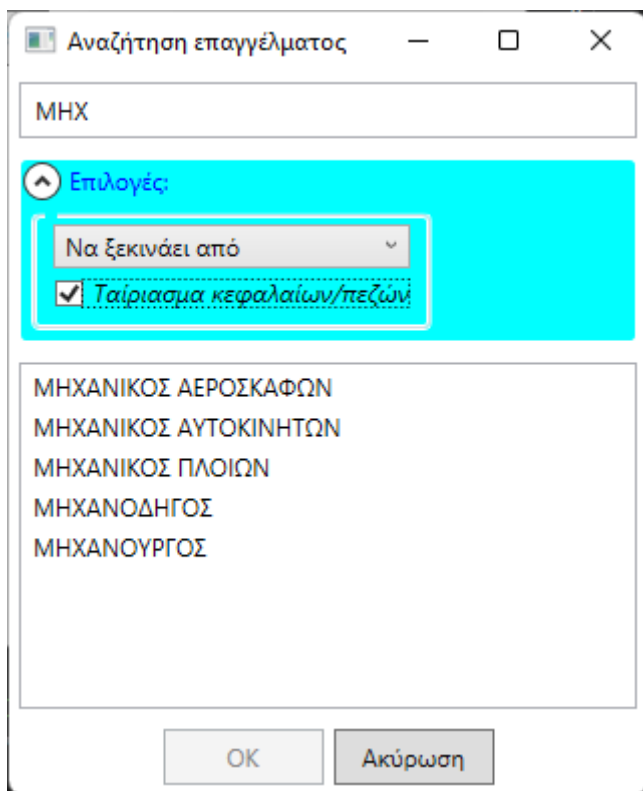
</Window.Resources>

<Grid>
    <z:FindXpress Name="findXpress" ButtonsOKCancelWidth="80"
        FindMatchCase="True"
        CheckBoxFindMatchCaseStyle="{StaticResource
            FindMatchCaseCheckBoxStyle }
        ... />
</Grid>

</Window>
```

**VB:**

```
findXpress.FindMatchCase = True
```



## FindMatchCaseCaption

Καθορίζει το κείμενο-λεκτικό (caption) για το checkbox "Match Case", τύπου String.

### Σύνταξη:

**VB:**

```
Public Property FindMatchCaseCaption As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για το checkbox "FindMatchCase". Η default τιμή είναι "Ταίριασμα κεφαλαίων/πεζών".

### Dependency Property Information:

**Identifier field:** FindMatchCaseCaptionProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την ιδιότητα **LanguageForUI** όπου θέτοντας την τιμή **LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

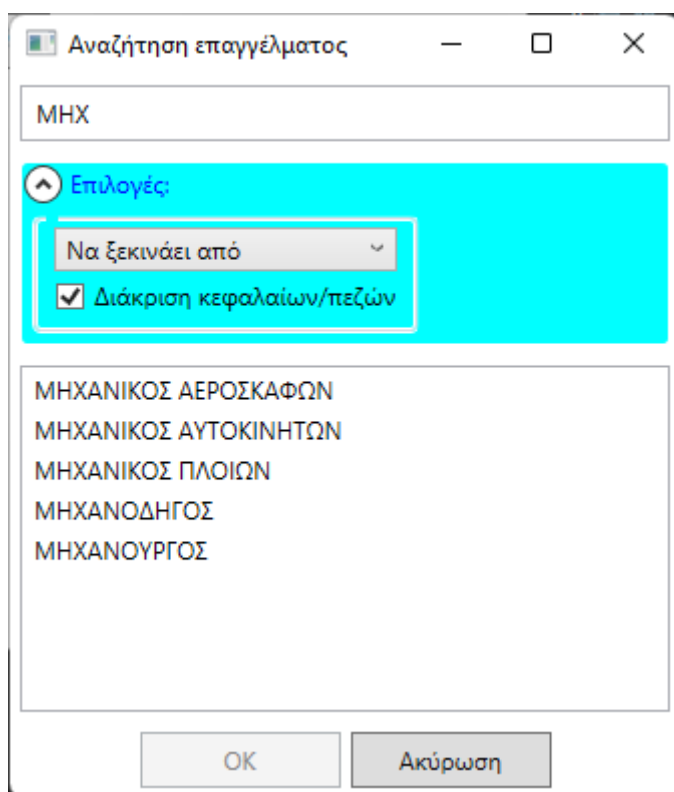
Στο επόμενο παράδειγμα, καθορίζουμε το λεκτικό "Διάκριση κεφαλαίων/πεζών" για το checkbox "FindMatchCase".

**XAML:**

```
<z:FindXPress Name="findXpress"  
    FindMatchCaseCaption="Διάκριση κεφαλαίων/πεζών" />
```

**VB:**

```
findXpress.FindMatchCaseCaption = "Διάκριση κεφαλαίων/πεζών"
```



## FindOptionsCaption

Καθορίζει το κείμενο (caption) για την επικεφαλίδα "Επιλογές". Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FindOptionsCaption As String
```

**Τύπος: System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για την επικεφαλίδα "Επιλογές". Η default τιμή είναι "Επιλογές".

### Dependency Property Information:

**Identifier field:** FindOptionsCaptionProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

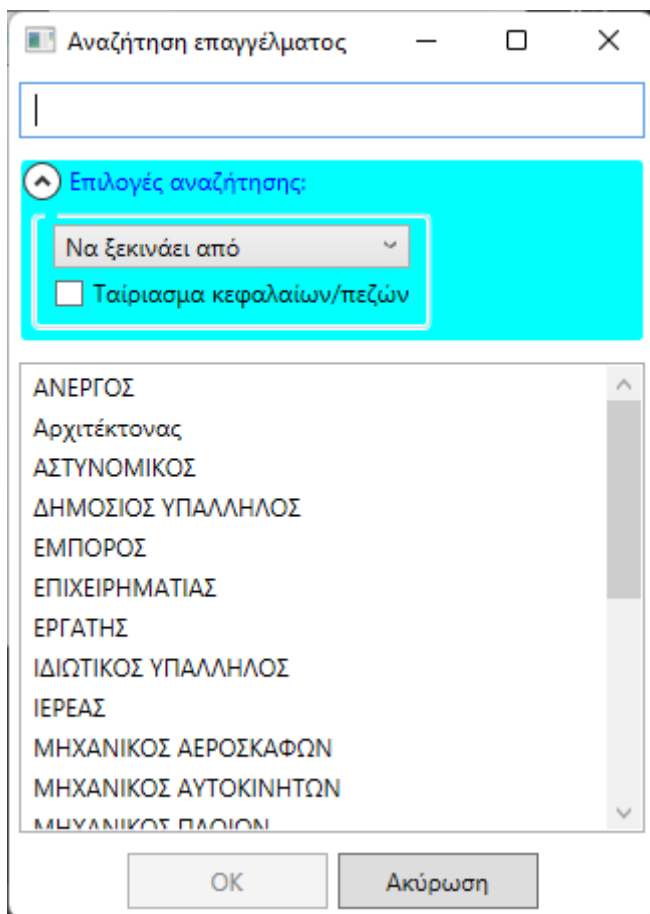
Στο επόμενο παράδειγμα, καθορίζουμε το **λεκτικό "Επιλογές αναζήτησης:"** για το την επικεφαλίδα του group "Επιλογές".

**XAML:**

```
<z:FindXPress Name="findXpress"  
    FindOptionsCaption="Επιλογές αναζήτησης:" />
```

**VB:**

```
findXpress.FindOptionsCaption = "Επιλογές αναζήτησης:"
```



## FindOrientation

Καθορίζει τον προσανατολισμό της αναζήτησης, τύπου **FindOrientationEnum**.

### Σύνταξη:

#### VB:

```
Public Property FindOrientation As FindOrientationEnum
```

**Τύπος:** **Zeus.WPF.Controls.DataControls.FindXPress.FindOrientationEnum**

Προσδιορίζουμε μία τιμή από την απαρίθμηση **FindOrientationEnum**, η οποία καθορίζει τον τύπο της αναζήτησης. Η default τιμή είναι **FindOrientationEnum.StartsWith**.

### Dependency Property Information:

**Identifier field:** **FindOrientationProperty**

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον προσανατολισμό της αναζήτησης στο "να περιέχει" ("contains").

#### XAML:

```
<z:FindXPress Name="findXpress"  
             FindOrientation="Contains" />
```

#### VB:

```
Imports Zeus.WPF.Controls.DataControls  
Imports Zeus.WPF.Controls.DataControls.FindXPress
```

```
findXpress.FindOrientation = FindOrientationEnum.Contains
```



Αναζήτηση επαγγέλματος

οδ

Επιλογές:

Να περιέχει

Ταίριασμα κεφαλαίων/πεζών

ΜΗΧΑΝΟΔΗΓΟΣ  
ΟΔΗΓΟΣ

OK Ακύρωση

## FindOrientationTextContains

Καθορίζει το κείμενο για την τιμή "Contains" στο combobox "Find Orientation". Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property FindOrientationTextContains As String
```

**Τύπος:** System.String

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για την τιμή "Contains" στο combobox "Find Orientation". Η default τιμή είναι "Να περιέχει".

### Dependency Property Information:

**Identifier field:** FindOrientationTextContainsProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά **"Starts with"**, **"Ends with"**, **"Contains"** και **"Equal to"** για τις τιμές του combobox **"Find Orientation"**.

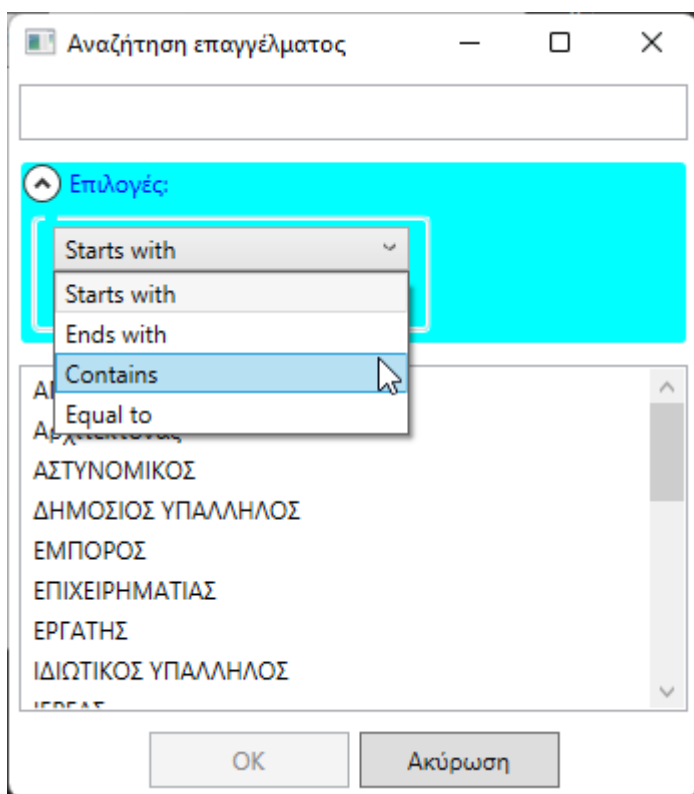
#### XAML:

```
<z:FindXPress Name="findXpress"  
    FindOrientationTextStartsWith="Starts with"  
    FindOrientationTextEndsWith="Ends with"  
    FindOrientationTextContains="Contains"  
    FindOrientationTextEqualTo="Equal to"  
>
```

#### VB:

```
findXpress.FindOrientationTextStartsWith = "Starts with"  
findXpress.FindOrientationTextEndsWith = "Ends with"  
findXpress.FindOrientationTextContains = "Contains"
```

findXpress.FindOrientationTextEqualTo = "Equal to"



## FindOrientationTextEndsWith

Καθορίζει το κείμενο για την τιμή "Ends with" στο combobox "Find Orientation". Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property FindOrientationTexEndsWith As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για την τιμή "Ends with" στο combobox "Find Orientation". Η default τιμή είναι "Να τελειώνει σε".

### Dependency Property Information:

**Identifier field:** FindOrientationTexEndsWithProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά **"Starts with"**, **"Ends with"**, **"Contains"** και **"Equal to"** για τις τιμές του combobox **"Find Orientation"**.

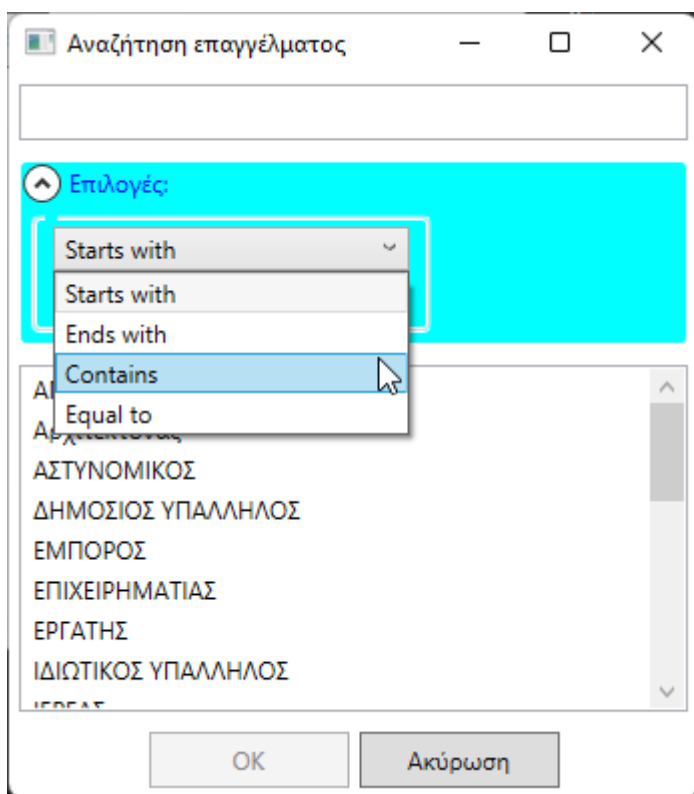
#### XAML:

```
<z:FindXPress Name="findXpress"  
    FindOrientationTextStartsWith="Starts with"  
    FindOrientationTextEndsWith="Ends with"  
    FindOrientationTextContains="Contains"  
    FindOrientationTextEqualTo="Equal to"  
/>
```

#### VB:

```
findXpress.FindOrientationTextStartsWith = "Starts with"  
findXpress.FindOrientationTextEndsWith = "Ends with"  
findXpress.FindOrientationTextContains = "Contains"
```

findXpress.FindOrientationTextEqualTo = "Equal to"



## FindOrientationTextEqualTo

Καθορίζει το κείμενο για την τιμή "Equal to" στο combobox "Find Orientation". Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FindOrientationTexEqualTo As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για την τιμή "Equal to" στο combobox "Find Orientation". Η default τιμή είναι "Να ισούται με".

### Dependency Property Information:

**Identifier field:** FindOrientationTexEqualToProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά **"Starts with"**, **"Ends with"**, **"Contains"** και **"Equal to"** για τις τιμές του combobox **"Find Orientation"**.

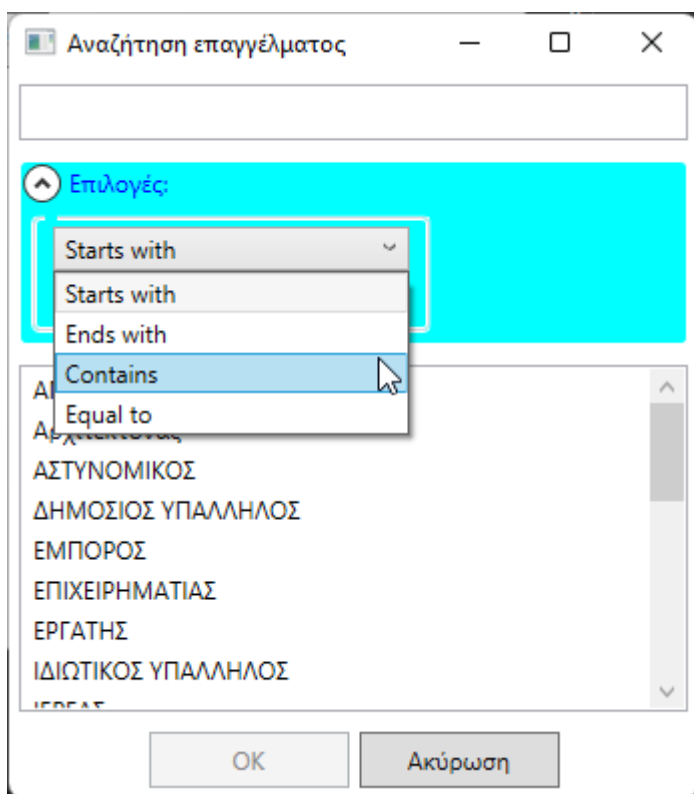
**XAML:**

```
<z:FindXPress Name="findXpress"  
    FindOrientationTextStartsWith="Starts with"  
    FindOrientationTextEndsWith="Ends with"  
    FindOrientationTextContains="Contains"  
    FindOrientationTextEqualTo="Equal to"  
>/>
```

**VB:**

```
findXpress.FindOrientationTextStartsWith = "Starts with"  
findXpress.FindOrientationTextEndsWith = "Ends with"  
findXpress.FindOrientationTextContains = "Contains"
```

findXpress.FindOrientationTextEqualTo = "Equal to"



## FindOrientationTextStartsWith

Καθορίζει το κείμενο για την τιμή "Starts with" στο combobox "Find Orientation". Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property FindOrientationTexStartsWith As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το λεκτικό για την τιμή "Starts with" στο combobox "Find Orientation". Η default τιμή είναι "Να ξεκινάει με".

### Dependency Property Information:

**Identifier field:** FindOrientationTexStartsWithProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τα λεκτικά **"Starts with"**, **"Ends with"**, **"Contains"** και **"Equal to"** για τις τιμές του combobox **"Find Orientation"**.

**XAML:**

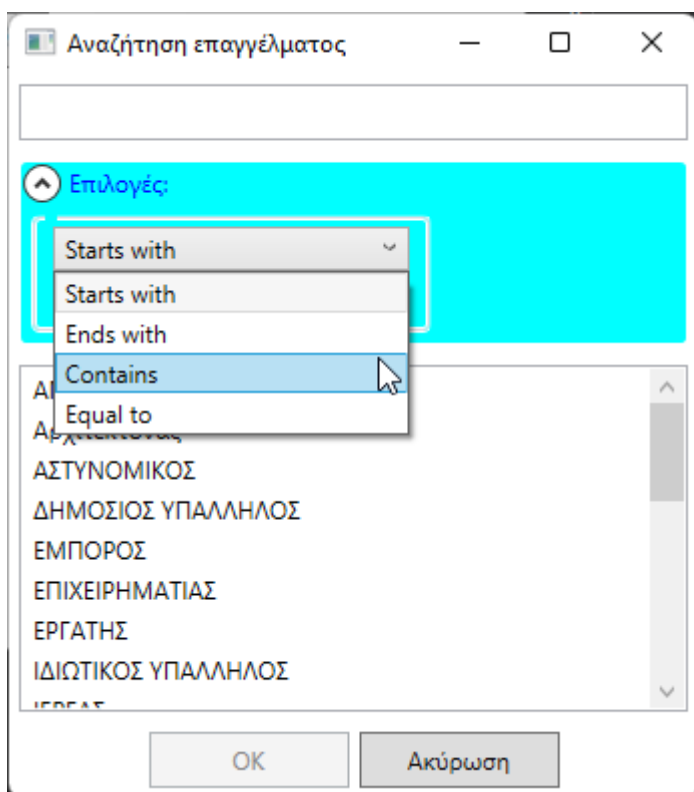
```
<z:FindXPress Name="findXpress"  
    FindOrientationTextStartsWith="Starts with"  
    FindOrientationTextEndsWith="Ends with"  
    FindOrientationTextContains="Contains"  
    FindOrientationTextEqualTo="Equal to"  
/>
```

**VB:**

```
findXpress.FindOrientationTextStartsWith = "Starts with"  
findXpress.FindOrientationTextEndsWith = "Ends with"  
findXpress.FindOrientationTextContains = "Contains"
```



findXpress.FindOrientationTextEqualTo = "Equal to"



## LanguageForUI

Καθορίζει το τη γλώσσα που θα χρησιμοποιηθεί στα κείμενα/captions του UI. Είναι τύπου **LanguageForUIEnum**.

### Σύνταξη:

#### VB:

```
Public Property LanguageForUI As LanguageForUIEnum
```

**Τύπος:** Zeus.WPF.Controls.DataControls.FindXPress.LanguageForUIEnum

Προσδιορίζουμε μία τιμή από την απαρίθμηση LanguageForUIEnum, η οποία καθορίζει τη γλώσσα που θα χρησιμοποιηθεί στα κείμενα/captions του UI. Η default τιμή είναι LanguageForUIEnum.Greek.

### Dependency Property Information:

**Identifier field:** LanguageForUIProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα (π.χ . FindOptionsCaption, FindMatchCaseCaption, FindOrientationTextStartsWith κλπ).

### Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε ότι τα **κείμενα/captions του UI** θα είναι στην **Αγγλική γλώσσα**.

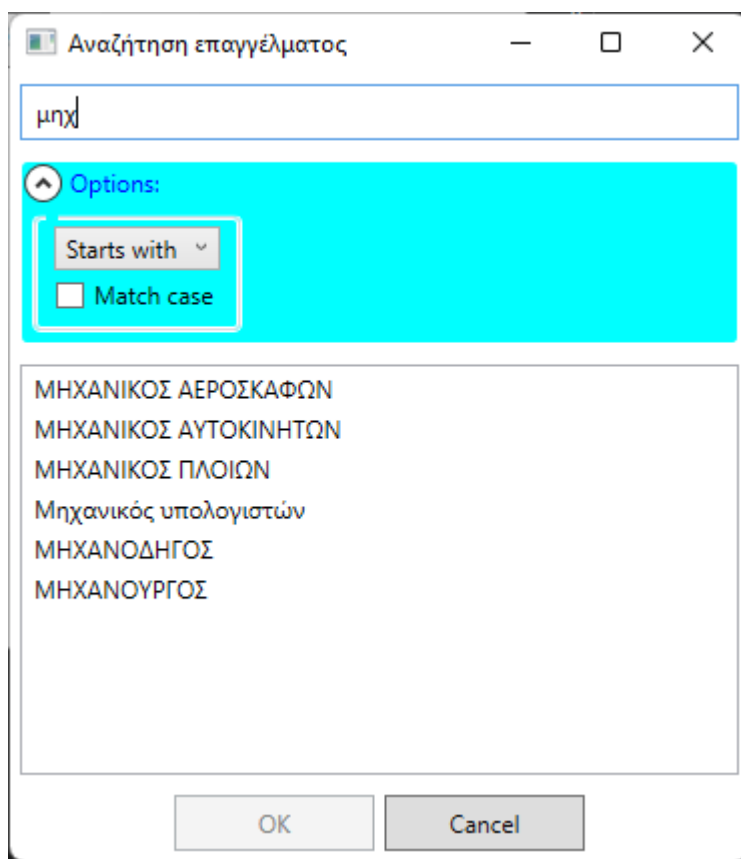
#### XAML:

```
<z:FindXPress Name="findXpress"  
             LanguageForUI="English" />
```

#### VB:

```
Imports Zeus.WPF.Controls.DataControls  
Imports Zeus.WPF.Controls.DataControls.FindXPress
```

```
findXpress.LanguageForUI = LanguageForUIEnum.English
```



## Παράδειγμα 2:

Στο επόμενο παράδειγμα, καθορίζουμε ότι τα **κείμενα/captions του UI** θα είναι στην **Ελληνική γλώσσα**.

### XAML:

```
<z:FindXpress Name="findXpress"  
              LanguageForUI="Greek" />
```

### VB:

```
Imports Zeus.WPF.Controls.DataControls  
Imports Zeus.WPF.Controls.DataControls.FindXPress  
  
findXpress.LanguageForUI = LanguageForUIEnum.Greek
```

Αναζήτηση επαγγέλματος

μηχ

Επιλογές:

Να ξεκινάει με

Ταίριασμα κεφαλαίων/πεζών

ΜΗΧΑΝΙΚΟΣ ΑΕΡΟΣΚΑΦΩΝ  
ΜΗΧΑΝΙΚΟΣ ΑΥΤΟΚΙΝΗΤΩΝ  
ΜΗΧΑΝΙΚΟΣ ΠΛΟΙΩΝ  
Μηχανικός υπολογιστών  
ΜΗΧΑΝΟΔΗΓΟΣ  
ΜΗΧΑΝΟΥΡΓΟΣ

OK Ακύρωση

## ListBoxResultStyle

Καθορίζει το **στυλ για το listbox του αποτελέσματος της αναζήτησης (λίστα)**, τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property ListBoxResultStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο listbox "Results". Το default style είναι αυτό ενός ListBox element.

### Dependency Property Information:

**Identifier field:** ListBoxResultStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"ResultListBoxItemStyle"** που θα αφορά στα **items του ListBox του αποτελέσματος**. Περιλαμβάνει δύο style triggers. Το πρώτο αφορά όταν ένα item είναι επιλεγμένο, το οποίο θα έχει έντονη γραφή. Το δεύτερο όταν το ποντίκι ίπταται πάνω από ένα item, το οποίο θα έχει μέγεθος 12, έντονη γραφή και χρώμα πορτοκαλί. Στη συνέχεια αυτό το resource style το θέτουμε στην **ιδιότητα ItemContainerStyle** του **ListBox του αποτελέσματος**.

**XAML:**

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

        <!-- To style για το ListBoxResultItem-->
        <Style x:Key="ResultListBoxItemStyle" TargetType="{x:Type ListBoxItem }">

            <Style.Triggers >
```

```
        <Trigger Property="IsSelected" Value="True">
            <Setter Property="FontWeight" Value="Bold"/>
        </Trigger>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="FontSize" Value="12"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Foreground" Value="Orange"/>
        </Trigger>
    </Style.Triggers>
</Style>

<!-- Το style θα εφαρμοστεί στο ListBoxItem του Result ListBox-->
<Style x:Key="ResultListBoxStyle" TargetType="{x:Type ListBoxItem}">
    <Setter Property="ItemContainerStyle"
        Value="{StaticResource ResultListBoxItemStyle }"/>
</Style>

</Window.Resources>

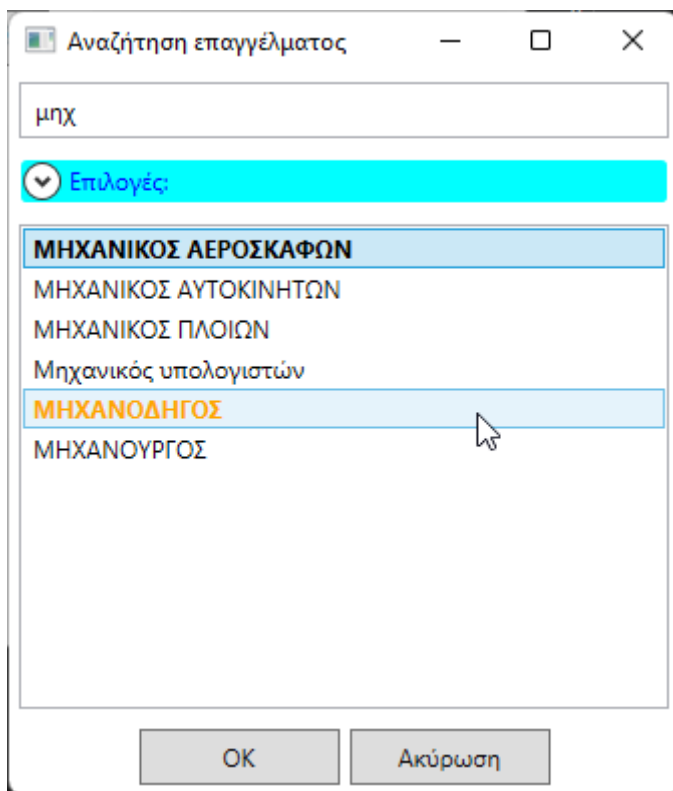
<Grid>
    <z:FindXPress Name="findXpress"
        ListBoxResultStyle="{StaticResource ResultListBoxStyle}"
    />
</Grid>

</Window>
```

**VB:**

```
findXpress.ListBoxResultStyle = FindResource("ResultListBoxStyle")
```

Στην παρακάτω εικόνα, το πρώτο item είναι επιλεγμένο ενώ το ποντίκι ίπταται στο προτελευταίο item.



## ReturnValue

Επιστρέφει την τιμή όταν ο χρήστης επιλέξει μία από τη λίστα αποτελέσματος και πατήσει το OK. Είναι τύπου `String`.

### Σύνταξη:

#### VB:

```
Public ReadOnly Property ReturnValue As String
```

#### Τύπος: `System.String`

Επιστρέφει μία τιμή `String` που διαλέγει ο χρήστης από τη λίστα αποτελέσματος. Η default τιμή είναι `String.Empty`.

### Dependency Property Information:

**Identifier field:** `ReturnValueProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, **ιδιότητα `ReturnValue`** αξιοποιείται για να επιλεγεί η αντίστοιχη τιμή σε ένα `ComboBox` με όνομα `cboProfessions` που βρίσκεται σε ένα παράθυρο. Να σημειωθεί ότι η πηγή δεδομένων του `cboProfessions` είναι μία συλλογή από αντικείμενα `Profession` (με ιδιότητες `ID` και `Description`).

#### VB:

```
If findXpress.ReturnValue IsNot Nothing AndAlso findXpress.ReturnValue.Length > 0 Then  
    cboProfessions.SelectedItem = _  
        professions.Where(Function(x) x.Description = findXpress.ReturnValue).Single  
End If
```



## TextBoxValueStyle

Καθορίζει το **στυλ για το textbox όπου ο χρήστης πληκτρολογεί ολόκληρη ή ένα μέρος της τιμής αναζήτησης**. Είναι τύπου **Style**.

### Σύνταξη:

#### VB:

```
Public Property TextBoxValueStyle As Style
```

#### Τύπος: **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο textbox"Value". Το default style είναι αυτό ενός TextBox element.

### Dependency Property Information:

**Identifier field:** TextBoxValueStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε, μεταξύ άλλων, ένα συγκεκριμένο **resource style** με όνομα **"ValueTextBoxStyle"** και το εφαρμόζουμε στο στην **ιδιότητα TextBoxValueStyle** του **FindXPress** control.

#### XAML:

```
<Window x:Class="FindValue"
    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:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
    xmlns:p="clr-namespace:DataControlsTestProject"
    mc:Ignorable="d"
    Title="Αναζήτηση επαγγέλματος" Height="400" Width="300" Loaded="Window_Loaded"
    Closing="Window_Closing">

    <Window.Resources>

        ...

        <!-- A style for the value TextBox -->
        <Style x:Key="ValueTextBoxStyle" TargetType="{x:Type TextBox}">

            <Setter Property="Foreground" Value="Blue"/>
            <Setter Property="FontWeight" Value="Normal"/>

            <Style.Triggers>
```

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

</Style.Triggers>

</Style>

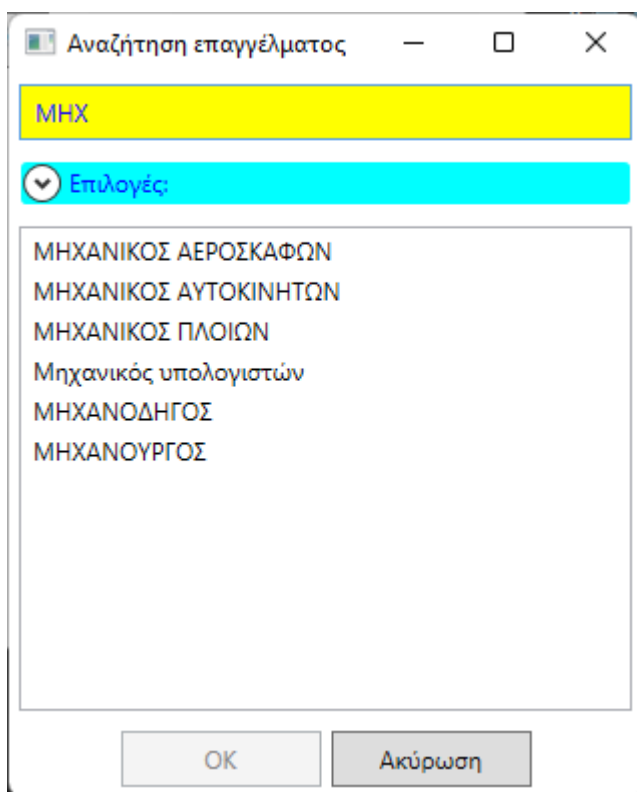
</Window.Resources>

<Grid>
    <z:FindXPress Name="findXpress"
        TextBoxValueStyle="{StaticResource ValueTextBoxStyle}" />
</Grid>

</Window>
```

**VB:**

```
findXpress.TextBoxValueStyle = FindResource("ValueTextBoxStyle")
```



## ToolTipTextForValue

Καθορίζει το κείμενο του tooltip για το textbox αναζήτησης. Είναι τύπου **String**.

### Σύνταξη:

**VB:**

```
Public Property ToolTipTextForValue As String
```

**Τύπος:** **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το κείμενο για το textbox "Value". Η default τιμή είναι "Παρακαλώ, δώστε την τιμή αναζήτησης".

### Dependency Property Information:

**Identifier field:** ToolTipTextForValueProperty

### Παρατηρήσεις:

Για τη διευκόλυνση του προγραμματιστή έχουμε δημιουργήσει την **ιδιότητα LanguageForUI** όπου θέτοντας την **τιμή LanguageForUIEnum.Greek** ή **LanguageForUIEnum.English**, τα λεκτικά του UI (buttons, checkboxes, comboboxitems κλπ) προσαρμόζονται αυτόματα στα Ελληνικά ή Αγγλικά, αντίστοιχα. Φυσικά, μπορούμε να θέσουμε συγκεκριμένη τιμή λεκτικού που θέλουμε στην αντίστοιχη ιδιότητα.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το κείμενο "**Παρακαλώ, δώστε ολόκληρο ή ένα μέρος της τιμής αναζήτησης**" για το Tooltip του TextBox αναζήτησης.

**XAML:**

```
<z:FindXPress Name="findXpress"
    ToolTipTextForValue="Παρακαλώ, δώστε ολόκληρο ή ένα μέρος της τιμής
αναζήτησης" />
```

**VB:**

```
findXpress.ToolTipTextForValue = _
    "Παρακαλώ, δώστε ολόκληρο ή ένα μέρος της τιμής αναζήτησης"
```

## ShowToolTipForValue

Καθορίζει αν θα εμφανίζεται το **tooltip για το textbox αναζήτησης**. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowToolTipForValue As Boolean
```

#### Τύπος: System.Boolean

Προσδιορίζουμε μία τιμή True/False που καθορίζει αν θα εμφανίζεται το tooltip για το textbox "Value". Η default τιμή είναι True.

### Dependency Property Information:

**Identifier field:** ShowToolTipForValueProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε να **μην εμφανίζεται το tooltip για το textbox αναζήτησης**.

#### XAML:

```
<z:FindXPress Name="findXpress"  
             ShowToolTipForValue="False" />
```

#### VB:

```
findXpress.ShowToolTipForValue = False
```

## Συμβάντα

Όνομα	Περιγραφή
OKClicked	Ενεργοποιείται όταν ο χρήστης πατήσει το button "OK".
CancelClicked	Ενεργοποιείται όταν ο χρήστης πατήσει το button "Ακύρωση".

## OKClicked

Ενεργοποιείται όταν ο χρήστης πατήσει το button "OK".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event OKClicked As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:FindXPress OKClicked="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε τοποθετήσει αυτόνομα το FindXpress σε ένα παράθυρο (Window). Όταν ο χρήστης πατήσει το button "OK", το περιεχόμενο της ιδιότητας ReturnValue τοποθετείται σε μία public ιδιότητα ReturnValue του παραθύρου και το παράθυρο κλείνει.

#### XAML:

```
<zeus:FindXPress Name="findXpress" ...  
    OKClicked="findXpress_OKClicked"  
    CancelClicked="findXpress_CancelClicked" />
```

#### VB:

```
Private Sub findXpress_OKClicked(sender As Object, e As RoutedEventArgs)  
  
    Dim fndXpress As FindXPress = TryCast(sender, FindXPress)  
  
    If fndXpress IsNot Nothing Then  
  
        ReturnValue = fndXpress.ReturnValue  
        fndXpress.CView.Filter = Nothing  
        Me.Close()  
  
    End If  
  
End Sub
```

## CancelClicked

Ενεργοποιείται όταν ο χρήστης πατήσει το button "Ακύρωση".

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event CancelClicked As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:FindXPress CancelClicked="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, έχουμε τοποθετήσει αυτόνομα το FindXPress σε ένα παράθυρο (Window). Όταν ο χρήστης πατήσει το button "Ακύρωση", το παράθυρο κλείνει.

#### XAML:

```
<zeus:FindXPress Name="findXpress" ...  
    OKClicked="findXpress_OKClicked"  
    CancelClicked="findXpress_CancelClicked" />
```

#### VB:

```
Private Sub findXpress_CancelClicked(sender As Object, e As RoutedEventArgs)  
    Me.Close()  
End Sub
```

## Styles και Templates

- Parts και States
- Το default Style και ControlTemplate
- Παράδειγμα ενός Custom Style and ControlTemplate



## Parts και States

Το default `ControlTemplate` περιλαμβάνει κάποια **part names** και **visual states**. Μπορείτε να τροποποιήσετε το default `ControlTemplate` ώστε να δώσετε στο control μία μοναδική εμφάνιση.

Ο παρακάτω πίνακας εμφανίζει τα **part names** του **FindXPress**:

Part	Τύπος	Περιγραφή
PART_Value	TextBox	Το textbox για την εισαγωγή από τον χρήστη της τιμής αναζήτησης.
PART_FindOrientation	ComboBox	Το combobox για τον τύπο της αναζήτησης (π.χ. να ξεκινάει με, περιέχει κλπ).
PART_FindMatchCase	CheckBox	Το checkbox για το ταίριασμα κεφαλαίων/πεζών.
PART_ListResult	ListBox	Το listbox για τα αποτελέσματα αναζήτησης.
PART_OK	Button	Το button επιβεβαίωσης ("OK").
PART_Cancel	Button	Το button ακύρωσης ("Ακύρωση").

Ο παρακάτω πίνακας εμφανίζει τα **visual states** του **FindXPress**:

VisualState	VisualStateGroup	Περιγραφή
Normal	CommonStates	Το default state.
MouseOver	CommonStates	Το state όταν ο δείκτης του ποντικιού είναι πάνω από το control.
Disabled	CommonStates	Το state όταν το control είναι disabled.
Focused	FocusStates	Το state όταν το control έχει το focus.
Unfocused	FocusStates	Το state όταν το control δεν έχει το focus.

*Το default `ControlTemplate` έχει καθορισμένο μόνο το `Disabled` state. Μπορείτε να δημιουργήσετε ένα custom `ControlTemplate` για να το αλλάξετε ή/και για να καθορίσετε τα υπόλοιπα.*

## To default Style και ControlTemplate

Ο XAML κώδικας για το **default Style** και **ControlTemplate** φαίνεται παρακάτω. Μπορείτε να βασιστείτε σε αυτόν για να δημιουργήσετε μία μικρή ή μεγάλη παραλλαγή του δικού σας custom Style και ControlTemplate:

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.DataControls">

<Style TargetType="{x:Type z:FindXPress}">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type z:FindXPress}">
                <!-- Root element -->
                <Border Background="{TemplateBinding Background}"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding BorderThickness}" >

                <!-- Visual States -->
                <VisualStateManager.VisualStateGroups >

                    <VisualStateGroup Name="CommonStates">

                        <VisualStateGroup.Transitions >
                            <VisualTransition
                                GeneratedDuration="0:0:0.2">
                            </VisualTransition>
                        </VisualStateGroup.Transitions>

                        <VisualState Name="Normal"/>
                        <VisualState Name="MouseOver" />
                        <VisualState Name="Disabled">
                            <Storyboard >
                                <DoubleAnimation Storyboard.TargetName="PART_OK"
                                    Storyboard.TargetProperty="(Button.Opacity)"
                                    To="0.5" />
                                <DoubleAnimation
                                    Storyboard.TargetName="PART_Cancel"
                                    Storyboard.TargetProperty="(Button.Opacity)"
                                    To="0.5" />
                            </Storyboard>
                        </VisualState>
                    </VisualStateGroup>

                    <VisualStateGroup Name="FocusStates">

                        <VisualStateGroup.Transitions >
                            <VisualTransition
                                GeneratedDuration="0:0:0.2">
                            </VisualTransition>
                        </VisualStateGroup.Transitions>

                        <VisualState Name="Focused" />
                        <VisualState Name="Unfocused"/>
                    </VisualStateGroup>
                </VisualStateManager.VisualStateGroups >
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

```
</VisualStateManager.VisualStateGroups>

<!-- Content -->
<Grid Name="grdMain" IsEnabled="{TemplateBinding IsEnabled}"
      Background="Transparent" Margin="{TemplateBinding Padding}">

  <Grid.ColumnDefinitions>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>
  <RowDefinition Height="Auto"/>
  <RowDefinition Height="Auto"/>
  <RowDefinition Height="*/>
  <RowDefinition Height="Auto"/>
</Grid.RowDefinitions>

<!-- The TextBox for value->
<TextBox Name="PART_Value" Margin="5" Padding="5"
         Style="{TemplateBinding TextBoxValueStyle}"/>

<!--Find options-->
<Expander Grid.Row="1" Name="OptionsExpander"
          Header="{TemplateBinding FindOptionsCaption}"
          Foreground="Blue" Background="Cyan"
          Margin="5" >

  <GroupBox Margin="5" HorizontalAlignment="Left"
           Padding="5" >

    <StackPanel>
      <ComboBox Name="PART_FindOrientation"
               Style="{TemplateBinding ComboBoxFindOrientationStyle}">
        <ComboBoxItem
          Content="{TemplateBinding FindOrientationTextStartsWith}"/>
        <ComboBoxItem
          Content="{TemplateBinding FindOrientationTextEndsWith}"/>
        <ComboBoxItem
          Content="{TemplateBinding FindOrientationTextContains}"/>
        <ComboBoxItem
          Content="{TemplateBinding FindOrientationTextEqualTo}"/>
      </ComboBox>

      <CheckBox Name="PART_FindMatchCase"
               Style="{TemplateBinding CheckBoxFindMatchCaseStyle}"
               Content="{TemplateBinding FindMatchCaseCaption}"
               IsThreeState="False" Margin="0,5,0,0"/>
    </StackPanel>

  </GroupBox>

</Expander>

<!-- Results in List-->
<ListBox Grid.Row="2" Name="PART_Result" Margin="5"
         Style="{TemplateBinding ListBoxResultStyle}"
         IsSynchronizedWithCurrentItem="True"
         ItemsSource="{TemplateBinding CView}"/>
```

```
<!-- Dialog buttons-->
<StackPanel Name="stkDialogButtons" Grid.Row="3"
  Orientation="Horizontal" Margin="5"
  HorizontalAlignment="Center">

  <Button Name="PART_OK"
    Content="{TemplateBinding ButtonOKText}" Padding="5"
    Width="{TemplateBinding ButtonsOKCancelWidth}"
    Style="{TemplateBinding ButtonOKStyle}"/>

  <Button Name="PART_Cancel"
    Content="{TemplateBinding ButtonCancelText}"
    Padding="5" Margin="5,0,0,0"
    Width="{TemplateBinding ButtonsOKCancelWidth}"
    Style="{TemplateBinding ButtonCancelStyle}"/>
</StackPanel>

</Grid>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>
```

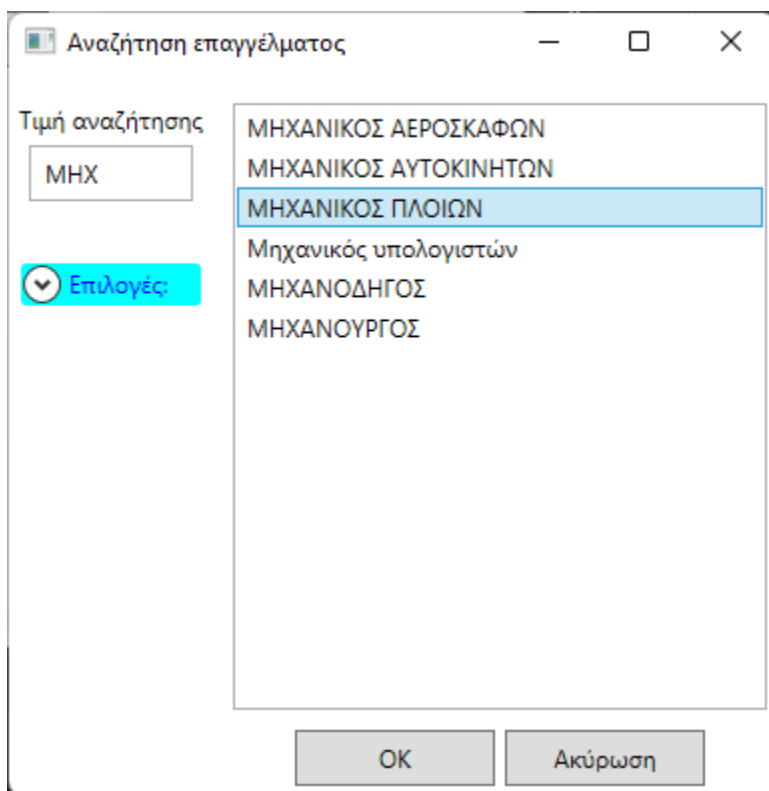
## Custom Style and ControlTemplate

Στο παρακάτω παράδειγμα, δημιουργούμε ένα **custom Style και ControlTemplate** στο οποίο:

- Το textbox αναζήτησης και οι επιλογές βρίσκονται αριστερά.
- Το listbox με τα αποτελέσματα αναζήτησης και τα dialog buttons OK και Cancel βρίσκονται δεξιά.

Το μόνο που προσέχουμε είναι **πού θα τοποθετήσουμε τα elements που το όνομά τους ξεκινάει με "PART\_"** π.χ. PART\_Value, PART\_ListResult κλπ. Η λειτουργικότητα τους είναι ενσωματωμένη.

Ένα δείγμα του control φαίνεται στην παρακάτω εικόνα:



Ο παρακάτω XAML κώδικας μπορεί να τοποθετηθεί στο τμήμα Window.Resources ενός Window element.

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.DataControls;assembly=ZeusDataControls"
```

```
<Style x:Key="findXPressCustomStyle" TargetType="{x:Type z:FindXPress}">
```

```
    <Setter Property="Template">
```

```
        <Setter.Value>
```

```
            <ControlTemplate TargetType="{x:Type z:FindXPress}">
```

```
<!-- Root element -->
<Border Background="{TemplateBinding Background}"
        BorderBrush="{TemplateBinding BorderBrush}"
        BorderThickness="{TemplateBinding BorderThickness}"
        CornerRadius="5"
    >

<!-- Visual States -->
<VisualStateManager.VisualStateGroups >

    <VisualStateGroup Name="CommonStates">

        <VisualStateGroup.Transitions >
            <VisualTransition
                GeneratedDuration="0:0:0.2">
            </VisualTransition>
        </VisualStateGroup.Transitions>

        <VisualState Name="Normal"/>
        <VisualState Name="MouseOver" />
        <VisualState Name="Disabled">
            <Storyboard >
                <DoubleAnimation
                    Storyboard.TargetName="PART_OK"
                    Storyboard.TargetProperty="(Button.Opacity)"
                    To="0.5" />
                <DoubleAnimation
                    Storyboard.TargetName="PART_Cancel"
                    Storyboard.TargetProperty="(Button.Opacity)"
                    To="0.5" />
            </Storyboard>
        </VisualState>
    </VisualStateGroup>

    <VisualStateGroup Name="FocusStates">

        <VisualStateGroup.Transitions >
            <VisualTransition
                GeneratedDuration="0:0:0.2">
            </VisualTransition>
        </VisualStateGroup.Transitions>

        <VisualState Name="Focused" />
        <VisualState Name="Unfocused"/>
    </VisualStateGroup>
</VisualStateManager.VisualStateGroups>

<!-- Content -->
<Grid Name="grdMain" IsEnabled="{TemplateBinding IsEnabled }"
        Background="Transparent"
        Margin="{TemplateBinding Padding}">

    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>

<!-- Left Column: TextBoxValue and FindOptions-->
<Grid Margin="0,10,0,0">
```

```
<Grid.RowDefinitions>
  <RowDefinition Height="Auto"/>
  <RowDefinition Height="Auto" />
</Grid.RowDefinitions>

<!-- The TextBox for value -->
<StackPanel Margin="5">
  <TextBlock Text="Τιμή αναζήτησης" />
  <TextBox Name="PART_Value" Margin="5" Padding="5"
    Style="{TemplateBinding TextBoxValueStyle}"/>
</StackPanel>

<!--Find options-->
<Expander Grid.Row="1" Name="OptionsExpander"
  Header="{TemplateBinding FindOptionsCaption}"
  Foreground="Blue" Background="Cyan"
  Margin="5,20,5,5" >

  <GroupBox Margin="5" HorizontalAlignment="Left"
    Padding="5" >

    <StackPanel>
      <ComboBox Name="PART_FindOrientation"
        Style="{TemplateBinding ComboBoxFindOrientationStyle}">
        <ComboBoxItem
Content="{TemplateBinding FindOrientationTextStartsWith}"/>
        <ComboBoxItem
Content="{TemplateBinding FindOrientationTextEndsWith}"/>
        <ComboBoxItem
Content="{TemplateBinding FindOrientationTextContains}"/>
        <ComboBoxItem
Content="{TemplateBinding FindOrientationTextEqualTo}"/>
      </ComboBox>
      <CheckBox Name="PART_FindMatchCase"
        Style="{TemplateBinding CheckBoxFindMatchCaseStyle}"
        Content="{TemplateBinding FindMatchCaseCaption}"
        IsThreeState="False" Margin="0,5,0,0"/>
    </StackPanel>

  </GroupBox>

</Expander>

</Grid>

<!-- Right Column: ListBoxResults and Dialog buttons-->
<Grid Grid.Column="1" Margin="5,10,0,0">

  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>

  <!-- Results in List-->
  <ListBox Name="PART_Result" Margin="5"
    Style="{TemplateBinding ListBoxResultStyle}"
    IsSynchronizedWithCurrentItem="True"
    ItemsSource="{TemplateBinding CView}"/>

  <!-- Dialog buttons-->
```

```
<StackPanel Name="stkDialogButtons" Grid.Row="1"
            Orientation="Horizontal" Margin="5"
            HorizontalAlignment="Center">
    <Button Name="PART_OK"
            Content="{TemplateBinding ButtonOKText}"
            Padding="5"
            Width="{TemplateBinding ButtonsOKCancelWidth}"
            Style="{TemplateBinding ButtonOKStyle}"/>
    <Button Name="PART_Cancel"
            Content="{TemplateBinding ButtonCancelText}"
            Padding="5" Margin="5,0,0,0"
            Width="{TemplateBinding ButtonsOKCancelWidth}"
            Style="{TemplateBinding ButtonCancelStyle}"/>
</StackPanel>
</Grid>
</Grid>
</Border>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
```

Και στο περιεχόμενο του Window element:

```
<z:FindXPress Name="findXpress" Style="{StaticResource findXpressCustomStyle}"
            ... />
```



# Τέλος Manual