## **Database Tutorials**

Hello everyone... welcome to database tutorials. These are going to be very basic tutorials about using the database to create simple applications, hope you enjoy it. If you have any notes about it, please send them to **notes@mka-soft.com**. Finally if you find these tutorials are useful, it would be nice from you to send a small donation via PayPal to **donation@mka-soft.com**.

The work with this tutorial started on 2010-SEPTEMBER-30.

## Validation

Today's tutorial is about Validation. The validation processes is needed to make sure that the data entered by the user are correct. For the sake of example, consider the following table:

0	alibri B <i>I</i> <u>U</u>    <u>A</u> +	+  11 +  [■ <u>}</u> - ] ⊞+] =	F 🖀 🗐 1	律計		efresh All *	Save	∑ Totals	
		Font	5	Ric	n Text		Records		
nt ir	the database has b	een disabled	Options	]					
	PName -	PAddress 👻	PTel	¥	PAge	+	PNotes	<ul> <li>POSTAL C</li> </ul>	OC -
	Michel	US	555456			22 1	N/A		11223
	Smith	UK	123456			30 1	N/A	1	7777
÷	ĸ								
			ß						

Now create a datasource in your project

		▲ [] [] & 레   I] & 관   []
Data Sources + 4 X	Form1.vb [Design] Wind	owsApplication15* Output
□	- Form1	
People -		

Drag and drop the table on the form as Details.

## http://www.mka-soft.com

Data Sources - 4 ×	Form1.vb [Design]* WindowsApplication15* Output
► EstDataSet	Form1       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PName:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PAddress:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PAddress:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PAddress:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PAge:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PNotes:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PNotes:       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)       Image: Control of (0)         PNotes:       Image: Control of (0)         PNotes:       Image: Control of (0)         PNotes:       Image: Control of (0)       Image: Control of (0)       Image: Control of

If you try to enter a new record, you can enter any wrong data.

i 🚺 🔺   3	of 3   ▶ →	+ × 🖬
PName:		
PAddress:	5	
PTel:	4	
PAge:	-3	li.
PNotes:	5 6	
POSTAL CODE:	1	

If you try to save the data, you will get an error similar to this:

Column 'PName' does not allow	nulls.	
Troubleshooting tips:	2	
Check if the value is DBNull befo	re adding it to the column.	
Set AllowDBNull to True.		
Get general help for this exceptio	n.	
Search for more Help Online		
Actions:		
View Detail		
C		

So now let us add validation to the controls. Double click the textbox control that stores the name of the person, and write the following in the validating event:

```
Private Sub PNameTextBox_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles PNameTextBox.Validating
' check if the name is not null
If PNameTextBox.Text = "" Then
        e.Cancel = True
        Exit Sub
End If
' Check the name is no more than 10 chars
If PNameTextBox.Text.Length > 10 Then
        e.Cancel = True
        Exit Sub
End If
End Sub
```

The validating event is triggered whenever the focus of the control is transferred to another one. In this event, you should check the input, if it is not correct, then set the cancel parameter to true to signal to the application that the input is wrong, and the user should correct it.

Now run the application, add a new record, and try not to enter data into the name textbox, you will find that you are unable to leave the textbox. Try to write a name more than 10 character in length, and again you will not be able to leave the control. Finally make the length of the name equal to 10 chars or less. You will find you are able to navigate into another control.

Now, the problem with this code is that the user can't figure out what is the problem with the data entered, so you need to display a message. To do that add an ErrorProvider control into the form, and modify the code to be similar to the following:

```
Private Sub PNameTextBox_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles PNameTextBox.Validating
    ' check if the name is not null
    If PNameTextBox.Text = "" Then
        e.Cancel = True
        ErrorProvider1.SetError(PNameTextBox, "A name should be entered")
```

```
Exit Sub
End If
' Check the name is no more than 10 chars
If PNameTextBox.Text.Length > 10 Then
    e.Cancel = True
    ErrorProvider1.SetError(PNameTextBox, "The name should not exceed 10 chars")
    Exit Sub
End If
' no error, hide the error message
ErrorProvider1.SetError(PNameTextBox, "")
End Sub
```

You should be able to see an error message now whenever you enter invalid data



Similarly, add validation to the other controls

```
Private Sub PAddressTextBox_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles PAddressTextBox.Validating
         the address can't be null
        If PAddressTextBox.Text = "" Then
            e.Cancel = True
           ErrorProvider1.SetError(PAddressTextBox, "You should provide an address")
            Exit Sub
       End If
       ErrorProvider1.SetError(PAddressTextBox, "")
   End Sub
   Private Sub PAgeTextBox_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles PAgeTextBox.Validating
        ' the age is +ve integer
       Try
           Dim I As Integer
            I = Integer.Parse(PAgeTextBox.Text)
            If I <= 0 Then
                Err.Raise(1)
           End If
       Catch ex As Exception
           e.Cancel = True
            ErrorProvider1.SetError(PAgeTextBox, "The age is a +ve integer value")
            Exit Sub
       End Try
```

```
ErrorProvider1.SetError(PAgeTextBox, "")
   End Sub
   Private Sub POSTAL_CODETextBox_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles POSTAL_CODETextBox.Validating
        ' the postal code is 5 digits
       Try
            If POSTAL_CODETextBox.Text.Length <> 5 Then
               Err.Raise(1)
           End If
           Dim I As Integer
            I = Integer.Parse(POSTAL_CODETextBox.Text)
            If I < 0 Then
               Err.Raise(1)
           End If
       Catch ex As Exception
            e.Cancel = True
            ErrorProvider1.SetError(POSTAL_CODETextBox, "The postal code is 5 digits")
           Exit Sub
        End Try
       ErrorProvider1.SetError(POSTAL_CODETextBox, "")
   End Sub
```

Now as you see, the validation for any field entered is done by writing the code into the Validating event of the control.

Finally, another method for validating data is to use the properties of the dataset.



When you double click to dataset, you get into the designer.

	People
3	PN/me
	PAddress
	PTel
	PAge
	PNotes
	POSTAL CODE
8	PeopleTableAdapter
Soul	Fill GetData ()

Click on any field, and check its properties

Properties		• + ×
PName DataColumn		1
Expression		
MaxLength	255	
Name	PName	
NullValue	(Throw exception)	
ReadOnly	False	
Source	PName	
Unique	True	

In the properties window you can specify a number of properties such as the length of the field, datatype, null values, and unique constraints.

So this will be all for today. If you have questions or notes, send them to notes@mka-soft.com.

Thank you.

mkaatr