

Forum:[Hack Forums](#)

Thread: [Visual studio 2013 help searching data grid](#)

iProgrammer x - November 25, 2019, 7:46 pm

hi all looking for some quick help, i have attached an image of the current issue im having so im building an epos sytem for a store however they need it to be able to search bar-codes in order to scan as oppose to clicking each item, so how would i search the bar code grid in the datagrid view?

i currently have something like this although this is not the correct code

```
Code Dim x As Integer
    x = ProductsDataGridView.Rows(1).Cells(3).Value

    If ProductsDataGridView.Columns.Contains(x) Then
        MsgBox("found")
    End If
```

[SCREENSHOT 1](#)

HFDone - November 26, 2019, 12:20 pm

you want to search Columns or Rows or both

itcorp - November 26, 2019, 3:06 pm

(Nov 25, 2019 08:46 PM)iProgrammer x Wrote: [\[CITE\]](#) hi all looking for some quick help, i have attached an image of the current issue im having so im building an epos sytem for a store however they need it to be able to search bar-codes in order to scan as oppose to clicking each item, so how would i search the bar code grid in the datagrid view?

i currently have something like this although this is not the correct code

```
Code Dim x As Integer
    x = ProductsDataGridView.Rows(1).Cells(3).Value

    If ProductsDataGridView.Columns.Contains(x) Then
        MsgBox("found")
    End If
```

[\[url=https://#/azc\]SCREENSHOT 1](#)

i see no reason why you are trying to do it this way,as far as I can see from the image you are storing data from the database in dataset or datatable therefore, it is easier to simply find information directly from the collection.

Here's a simple example with datatable.
Code Imports System.ComponentModel

```
Public Class Form1
    Private dt As DataTable
```

Private _source As BindingSource

Public Class Product

Public Property id As Integer

Public Property ProductName As String

Public Property Barcode As Integer

Public Property Price As Decimal

Public Property Description As String

End Class

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim products As List(Of Product) = New List(Of Product)() From {

New Product() With {

.id = 1,

.ProductName = "product1",

.Barcode = 1233333333,

.Price = 20,

.Description = ""

},

New Product() With {

.id = 2,

.ProductName = "product2",

.Barcode = 12547,

.Price = 20,

.Description = ""

},

New Product() With {

.id = 3,

.ProductName = "product3",

.Barcode = 124367,

.Price = 20,

.Description = ""

},

New Product() With {

.id = 4,

.ProductName = "product4",

.Barcode = 999874521,

.Price = 20,

.Description = ""

}

}

dt = ConvertToDataTable(Of Product)(products)

DataGridView1.DataSource = dt

End Sub

Private Sub btnSearch_Click(sender As Object, e As EventArgs) Handles btnSearch.Click

_source = New BindingSource()

_source.DataSource = DataGridView1.DataSource

_source.Filter = String.Format("CONVERT({0}, System.String) like '%{1}%'", "Barcode",
txtSearch.Text.Trim())

DataGridView1.DataSource = _source

End Sub

Public Function ConvertToDataTable(Of T)(ByVal data As IList(Of T)) As DataTable

```

Dim properties As PropertyDescriptorCollection = TypeDescriptor.GetProperties(GetType(T))
Dim table As DataTable = New DataTable()

For Each prop As PropertyDescriptor In properties
    table.Columns.Add(prop.Name, If(Nullable.GetUnderlyingType(prop.PropertyType),
prop.PropertyType))
Next

For Each item As T In data
    Dim row As DataRow = table.NewRow()

    For Each prop As PropertyDescriptor In properties
        row(prop.Name) = If(prop.GetValue(item), DBNull.Value)
    Next

    table.Rows.Add(row)
Next

Return table
End Function
End Class

```

iProgrammer x - November 26, 2019, 7:10 pm

Thanks guys this has all been fixed now and thanks for the input i really appreciate it @itcorp thanks for the +3 :D

Eternal Darkness - November 27, 2019, 7:03 pm

(Nov 26, 2019 04:06 PM)itcorp Wrote: [CITE](#) i see no reason why you are trying to do it this way,as far as I can see from the image you are storing data from the database in dataset or datatable therefore, it is easier to simply find information directly from the collection.

Here's a simple example with datatable.
Code Imports System.ComponentModel

```

Public Class Form1
    Private dt As DataTable
    Private _source As BindingSource

    Public Class Product
        Public Property id As Integer
        Public Property ProductName As String
        Public Property Barcode As Integer
        Public Property Price As Decimal
        Public Property Description As String
    End Class

```

```

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Dim products As List(Of Product) = New List(Of Product)() From {
        New Product() With {
            .id = 1,
            .ProductName = "product1",
            .Barcode = 1233333333,
            .Price = 20,

```

```

        .Description = ""
    },
    New Product() With {
        .id = 2,
        .ProductName = "product2",
        .Barcode = 12547,
        .Price = 20,
        .Description = ""
    },
    New Product() With {
        .id = 3,
        .ProductName = "product3",
        .Barcode = 124367,
        .Price = 20,
        .Description = ""
    },
    New Product() With {
        .id = 4,
        .ProductName = "product4",
        .Barcode = 999874521,
        .Price = 20,
        .Description = ""
    }
}
dt = ConvertToDataTable(Of Product)(products)
DataGridView1.DataSource = dt
End Sub

```

```

Private Sub btnSearch_Click(sender As Object, e As EventArgs) Handles btnSearch.Click
    _source = New BindingSource()
    _source.DataSource = DataGridView1.DataSource
    _source.Filter = String.Format("CONVERT({0}, System.String) like '%{1}%'", "Barcode",
txtSearch.Text.Trim())
    DataGridView1.DataSource = _source
End Sub

Public Function ConvertToDataTable(Of T)(ByVal data As IList(Of T)) As DataTable
    Dim properties As PropertyDescriptorCollection = TypeDescriptor.GetProperties(GetType(T))
    Dim table As DataTable = New DataTable()

    For Each prop As PropertyDescriptor In properties
        table.Columns.Add(prop.Name, If(Nullable.GetUnderlyingType(prop.PropertyType),
prop.PropertyType))
    Next

    For Each item As T In data
        Dim row As DataRow = table.NewRow()

        For Each prop As PropertyDescriptor In properties
            row(prop.Name) = If(prop.GetValue(item), DBNull.Value)
        Next

        table.Rows.Add(row)
    Next

```

Return table
End Function
End Class

Amazed 0xDeadzo hasn't started flaming the thread yet for posting a non-excessive ("spoonfeeding") amount of code and firing off a bunch of personal attacks... Apparently re-posing a single method is considered spoonfeeding these days... Seems like a simple relevant example, good job.

inb4 "I called this code perfect". Don't like that string literal in the String.Format, and some other "minor stuff", but understand its just an example. "Community" has gotten so nit-picky... ?

itcorp - December 1, 2019, 1:44 pm

(Nov 27, 2019 08:03 PM)Eternal Darkness Wrote: [\[CITE\]](#) Amazed 0xDeadzo hasn't started flaming the thread yet for posting a non-excessive ("spoonfeeding") amount of code and firing off a bunch of personal attacks... Apparently re-posing a single method is considered spoonfeeding these days... Seems like a simple relevant example, good job.

inb4 "I called this code perfect". Don't like that string literal in the String.Format, and some other "minor stuff", but understand its just an example. "Community" has gotten so nit-picky... ?

0xDeadzo can always feel free to comment on my work he simply has the knowledge and i respect him because of that and i will take his opinion into account when the debate comes up at any time.

As for my solution,i'm glad you recognize the flaws and possible improvements in this example but as you just said this is spoonfeeding because I avoid giving complete solutions as much as I can for my own reasons but also in order to learn something in the whole process in order to improve themselves.

(Nov 26, 2019 08:10 PM)iProgrammer x Wrote: [\[CITE\]](#) Thanks guys this has all been fixed now and thanks for the input i really appreciate it @itcorp thanks for the +3 :D

You're welcome we are here to help if we can,keep working and don't hesitate to ask a question if something bothers you.All the best

0xDEAD10CC - December 2, 2019, 9:29 am

(Nov 27, 2019 08:03 PM)Eternal Darkness Wrote: [\[CITE\]](#) Amazed 0xDeadzo hasn't started flaming the thread yet for posting a non-excessive ("spoonfeeding") amount of code and firing off a bunch of personal attacks... Apparently re-posing a single method is considered spoonfeeding these days... Seems like a simple relevant example, good job.

inb4 "I called this code perfect". Don't like that string literal in the String.Format, and some other "minor stuff", but understand its just an example. "Community" has gotten so nit-picky... ?

Nice to know you're still butthurt and have to re-post your complaints entirely out of context due to your lack of comprehension skills for what was (or should've been) clearly stated back then.

Obviously you're still too naive to understand.

1. You posted an exact solution for the OP in the other thread, itcorp is posting an example of what should be done here (with better information/reasoning than what you provided in the other thread). It likely can't be used for a direct copy and paste because datasets differ.

2. "a single method" in your case was the Main() function... I don't see how this is even comparable, and how or why you had to bring up OO in the other discussion as a requirement for an example to be "complete" -- since it makes absolutely no sense.

itcorp's example is perfectly fine; he gave an indirect example with his suggestion and explained it accordingly. None of which you did at all, in any way whatsoever. I was also going to mention that you don't search through the controls for the data, you always search through the datasource that

you use to bind to or populate the control. These are basic fundamentals for UI driven programs, and all you have to do is look up the goals behind a pattern like MVC to understand.

It's quite clear that OP has intentions to learn/understand the solutions given to him too. At least we know some members aren't as gullible as you Eternal. ?