

[HOME](#)[ARTICLES](#)[FORUMS](#)[COMMUNITY](#)[MEMBERS](#)[NEWSLETTER](#)[Database](#) [Ethical hacking Tips](#) [Game Programming](#) [Engineering Concepts](#) [Internet Marketing](#) [Gadgets Review & Analysis](#)[Home](#) > [Articles](#) > [Programming](#) > [Visual Basic \[VB\]](#) >

This site uses cookies. By continuing to use this site, you are agreeing to our use of cookies. [Learn More](#).

Introduction to DataGridView Control in VB.NET

Discussion in 'Visual Basic [VB]' started by akhtar727, Aug 19, 2014.



akhtar727
New Member

JOINED: Aug 2, 2014
MESSAGES: 3
LIKES RECEIVED: 5
TROPHY POINTS: 0

Working with the DataGridView control is easy in Visual Basic.NET. The useful features of this control makes database software development a breeze. The ready-made features such as the properties, methods, functions and events do most of the tasks.

What is DataGridView?

This is a control in Visual Basic .NET, which provides you with an interactive user-interface to show information graphically. The user-interface is interactive, because the end-user of the software can interact with it if the developer of the software allows. The end-user can add, edit and remove columns if the developer allows. And also, they can edit the content if the permission is given. The DataGridView control looks like the Excel spreadsheet, which can be added onto the form so that the end-user can use it for specific purposes.

The DataGridView control is basically made of rows and columns. Additionally, every column has its header, where the header text can be changed. The header text can be changed in the designer. It can also be changed programmatically. Vertical and horizontal scrollbars appear automatically whenever they are needed. The scrollbars are used to scroll through the pages to see more content. So a DataGridView control may contain information of multiple pages.

Show Data From Database into DataGridView

Although you can add data manually, the DataGridView control is generally used for showing data from the database. For example, if you have an Access database connected to the Visual Basic project, you can show data from it. The data from the database will be shown in rows and columns of the DataGridView control. Besides Access databases, you can also show data on the DataGridView control from other database management systems like Oracle, MS [SQL Server](#), MySQL, dBASE, etc.

Choosing Data Source for the DataGridView control

After adding the DataGridView control on the form, the first thing that you need to do is to choose the data source, either from the designer or in the code, so that the data is automatically shown from the database.

[Sign up now!](#)

MEMBERS ONLINE NOW

Total: 226 (members: 0, guests: 178, robots: 48)

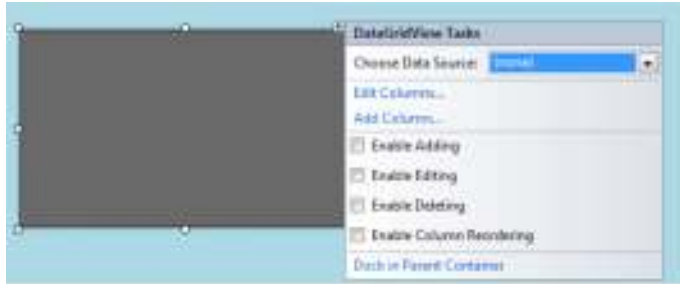
NEW PROFILE POSTS

- SAI HACKER** Hi am dr rann iam black hat hacker
Apr 10, 2017
- hypertuned** ► **Cleptography** Hello, I am new to programming. I get the following errors in your keylogger program
1. Application.Run() in main.
2. Application.ExecutablePath.ToString() in startup
3. Process[] ProcessList = Process.GetProcesses() in OnTimedEvent
4. using (Process curProcess = Process.GetCurrentProcess()) in IntPtr SetHook
5. Keys in IntPtr LowLevelKeyboardProc
Please let me know how to fix it
Mar 15, 2017
- SHK** Supported Dongle: HASP HL SRM Hardlock Marx CryptoBox Eutron Smartkey 3 Wibu BOX Dinkey 2 Keylok 2 Rockey 2/4/4ND Sentinel SuperPro/UltraPro
Mar 10, 2017
- SHK** To those who need help to emulate a dongle Communicate with me (dongledumper@outlook.com)
Mar 10, 2017
- Rick Davis** I want to thank shabbir and the administrators for allowing me into the group. I tried other sites but met mostly people selling services
Mar 10, 2017

FORUM STATISTICS

DISCUSSIONS:	24,684
MESSAGES:	96,145
MEMBERS:	55,557
LATEST MEMBER:	maxim1970

Choosing data source from the designer: The following screenshot demonstrates how to choose data source from the designer.



Choosing data source in the code: The following code shows you how to add a data source.

Code:

```
Private Sub FormMailBook_Load(ByVal sender As Object, ByVal e As EventArgs) Handles MyBase.Load
    DataGridView1.DataSource = db.data()
End Sub
```

The code is written inside the load event procedure. So the data source is selected when the form loads. Here, data() is a user-defined function and tablemailbook is the name of the database table.

The advantage of loading data from an existing database is that you don't need to manually put the content onto the DataGridView control. All the data is automatically loaded row-wise and column-wise. So this one line of code can save development time and effort significantly.

Layout of DataGridView control

You may design the DataGridView control according to your requirements. For example, you can change the background color. You may want to change the fonts, sizes, etc., or you may customize the rows and columns so that the DataGridView control meets your needs. There are several useful properties that can be used to design the DataGridView control. You may change the properties from the designer, or you may even set properties in the code when it's necessary to do so. As this control is very customizable, you should customize it to fulfill all of your design related requirements.

Important Properties

The DataGridView control has many useful properties. The properties, when used correctly, can save development time and

effort significantly. The properties make working with the DataGridView control almost effortless.

You can set the properties in design time or in the code. According to your specific needs, choose whether some properties need to be set in the code or whether some of them have to be set using the designer. There are some run-time properties which you cannot set using the designer. There are some ReadOnly properties. The ReadOnly properties are used only to get values, not to set.

There are several properties of the DataGridView control. This section lists only the most important properties that are frequently used while working with the DataGridView control.

Getting rows: the Rows property

'Rows' is a read-only property, which is used to get a collection of all the rows.

Getting columns: the Columns property

'Columns' is a read-only property, which is used to get a collection of all the columns.

Selecting rows: the Selected property

The Selected property is used in combination with the 'Rows' property to select a row, or to know whether the row is selected or not. This gives Boolean values.

```
Code:
For i = 0 To DataGridView1.Rows.Count - 1
    DataGridView1.Rows(i).Selected = True
Next
```

DataGridView1.Rows.Count returns the number of rows. So this code selects all the rows in the DataGridView control.

Counting rows/columns: the Count property

The 'Count' property is used in combination of 'Rows' and 'Columns' properties. It gives the number of the rows or columns.

```
Code:
Label1.Text = DataGridView1.Rows.Count
```

The 'Count' property counts the number of rows and assigns the value to the Text property of the Label1 control, so that the value is shown in the Label control.

Getting selected rows: the SelectedRows property

SelectedRows is a ReadOnly property that is used to get the rows selected by the user.

```
Code:
For i = 0 To DataGridView1.SelectedRows.Count - 1
    db.data("delete from tablem")
Next i
```

This code deletes values from the rows that are selected by the user. If the user selects 5 rows, all the values from 5 rows will be erased in this case. The above code uses SQL query to delete values from the database. 'Tablemailbook' is the name of the table in the database. Data() is a user-defined function here.

SelectedColumns

SelectedColumns is a ReadOnly property that is used to get the columns selected by the user.

SelectedCells

SelectedCells is a ReadOnly property that is used to get the cells selected by the user.

Changing the header text: the HeaderText property

You can easily change the header text using the HeaderText property.

```
Code:
DataGridView1.Columns.Item(1).HeaderText =
```

This code changes the header text of the second column to "Email Address". The index starts from 0, so Item(1) is the second item in the columns collection. You can change the header texts of all the columns using a loop.

Getting or setting cell values: the Value property

You can get or set the value of a particular cell. Consider the following code to understand it.

```
Code:
TextBoxName.Text = DataGridView1.SelectedRo
```

This code assigns the value from a cell to the Text property of the TextBox control.

Changing the background color: the BackgroundColor property

You can change the background color of the DataGridView control using the BackgroundColor property.

```
Code:
DataGridView1.BackgroundColor = Color.Aqua
```

This code sets the background color of the DataGridView control. You generally don't need to set the background color in the code, because this can be done using the designer. In some cases, you may need to write the code, though.

Number of columns: the ColumnCount property

The ColumnCount property is used to get or set the number of columns in the DataGridView control.

Row at a specified index: the Item property

Item is a ReadOnly property that is used to get an item such as a row at a specified index. The index value 0 indicates the first item.

```
Code:
TextBoxName.Text = DataGridView1.SelectedRo
```

In this code, Item(6) refers to the 7th row among the selected rows.

Multiple selections: the MultiSelect property

MultiSelect is a Boolean property, which is used to get or set a value that indicates whether the user can select more than one cell, column or row.

```
Code:
DataGridView1.MultiSelect = True
```

This code sets the MultiSelect property to True. So the user will be able to perform multiple selections on cells, columns and rows.

The ReadOnly property

The ReadOnly property is used to get or set a value that indicates whether the user of the program can edit the cells of the control.

```
Code:
DataGridView1.ReadOnly = True
```

This code makes the DataGridView control read-only. That means, the user of the program cannot make any changes in the DataGridView control.

Important Methods

There are several methods exposed by the DataGridView control. Here, only the most important methods are discussed, which are frequently used by the programmers. The methods perform some operations on the objects. So use the methods exposed by the DataGridView control when you want the DataGridView control, rows or the columns to perform some actions.

Selecting cells: the SelectAll method

Use the SelectAll method to select all the cells in the DataGridView control.

```
Code:
DataGridView1.Rows.SelectAll()
```

Adding rows/columns: the Add method

The Add method is used in combination with the Rows or Columns properties to add rows and columns. Consider the following example.

```
Code:
```

```
DataGridView1.Rows.Add("Car", "bike", "train")
```

GetFirstRow

The GetFirstRow method returns the index of the first row. This method is used in combination with the Rows property.

GetLastRow

The GetLastRow method returns the index of the last row. This method is used in combination with the Rows property.

GetNextRow

The GetNextRow method returns the index of the next row. This method is used in combination with the Rows property.

GetPreviousRow

The GetPreviousRow method returns the index of the previous row. This method is used in combination with the Rows property.

GetRowCount

The GetRowCount method returns the number of rows. This method is used in combination with the Rows property.

Important Events

The DataGridView control raises many events according to the specific actions performed by the user. This section covers only few of them.

CellMouseClick

The CellMouseClick event fires when the user clicks on the cell.

CellMouseDoubleClick

The CellMouseDoubleClick event fires when the user double-clicks on the cell.

Code:

```
Private Sub DataGridView1_CellMouseDoubleClick()  
    If e.RowIndex <> -1 Then  
        ButtonEdit.PerformClick()  
    End If  
End Sub
```

The code inside the CellMouseDoubleClick event procedure is executed when the user double-clicks on the cell.

KeyDown

The KeyDown event is fired when the user presses a key from the keyboard.

Code:

```
Private Sub DataGridView1_KeyDown(ByVal sender As Object, ByVal e As KeyEventArgs)  
    If e.KeyCode = Keys.Delete Then
```

```
        ButtonDelete.PerformClick()  
    End If  
End Sub
```

The KeyDown event handler is executed when the user presses a key. The DataGridView control should be in focus in order that the event for the DataGridView control is triggered.

KeyPress

The KeyPress event handler is executed when the user releases a key from the keyboard.

MouseClicked

The MouseClick event fires when the user clicks on the DataGridView control with a mouse.

CellClick

The CellClick event handler is invoked when the user clicks on any portion of the cell.

CellContentClick

The CellContentClick event handler is invoked when the user clicks on the content of the cell.

CellContentDoubleClick

The CellContentDoubleClick event handler is invoked when the user double-clicks on the content of the cell.

CellDoubleClick

The CellDoubleClick event fires when the user double-clicks on the cell.

CellMouseClick

The CellMouseClick event fires when the user clicks on the cell with a mouse.

Code:

```
Private Sub DataGridView1_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) As Void  
    If e.Button = Windows.Forms.MouseButtons.Left Then  
        ContextMenuStrip1.Show(MousePosition, e.Button)  
    End If  
    For i = 0 To DataGridView1.Rows.Count - 1  
        DataGridView1.Rows(i).Selected = True  
    Next  
    If e.RowIndex <> -1 Then  
        DataGridView1.Rows(e.RowIndex).Selected = True  
    End If  
End Sub
```

The code inside the CellMouseClick event procedure is executed when the cell is clicked with a mouse.

Click

The Click event occurs when the user clicks on the DataGridViewControl.

Last edited by a moderator: Jan 21, 2017

akhtar727, Aug 19, 2014

SHARE #1

SunilC likes this.



SunilC
New Member

JOINED: Sep 25, 2005
MESSAGES: 5
LIKES RECEIVED: 0
TROPHY POINTS: 0

Thanks for sharing such a wonderful writeup.

SunilC, Aug 20, 2014

SHARE #2

(You must log in or sign up to reply here.)

Share This Page

Tweet



Be the first of your friends to like this.

[Home](#) > [Articles](#) > [Programming](#) > [Visual Basic \[VB\]](#) >

[CONTACT US](#) [HELP](#) [HOME](#) [TOP](#)

User Contribution Licensed Under Creative Commons with Attribution Required. Site Design and Logo Copyright © Go4Expert™ 2004 - 2017.

Forum software by XenForo™ ©2010-2017 XenForo Ltd.
Some XenForo functionality crafted by ThemeHouse.

[TERMS AND RULES](#) [PRIVACY POLICY](#)

