

DataGridView in Windows Forms – Tips, Tricks and Frequently Asked Questions(FAQ)

DataGridView control is a Windows Forms control that gives you the ability to customize and edit tabular data. It gives you number of properties, methods and events to customize its appearance and behavior. In this article, we will discuss some frequently asked questions and their solutions. These questions have been collected from a variety of sources including some newsgroups, MSDN site and a few, answered by me at the MSDN forums.

Tip 1 – Populating a DataGridView /Wypełnienie DGV/

In this short snippet, we will populate a DataGridView using the LoadData() method. This method uses the SqlDataAdapter to populate a DataSet. The table 'Orders' in the DataSet is then bound to the BindingSource component which gives us the flexibility to choose/modify the data location.

C#

```
public partial class Form1 : Form
{
    private SqlDataAdapter da;
    private SqlConnection conn;
    BindingSource bsource = new BindingSource();
    DataSet ds = null;
    string sql;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnLoad_Click(object sender, EventArgs e)
    {
        LoadData();
    }

    private void LoadData()
    {
        string connectionString = "Data Source=localhost;Initial Catalog=Northwind;" + "Integrated Security=SSPI;";
        conn = new SqlConnection(connectionString);
        sql = "SELECT OrderID, CustomerID, EmployeeID, OrderDate, Freight," + "ShipName, ShipCountry FROM Orders";

        da = new SqlDataAdapter(sql, conn);
        conn.Open();
        ds = new DataSet();
        SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);
        da.Fill(ds, "Orders");
        bsource.DataSource = ds.Tables["Orders"];
        dgv.DataSource = bsource;
    }
}
```

VB.NET

```
Public Partial Class Form1
    Inherits Form
    Private da As SqlDataAdapter
    Private conn As SqlConnection
    Private bsource As BindingSource = New BindingSource()
    Private ds As DataSet = Nothing
    Private sql As String

    Public Sub New()
        InitializeComponent()
    End Sub

    Private Sub btnLoad_Click(ByVal sender As Object, ByVal e As EventArgs)
        LoadData()
    End Sub
```

```

        Private Sub LoadData()
Dim connectionString As String = "Data Source=localhost;Initial Catalog=Northwind;" &
"Integrated Security=SSPI;"
        conn = New SqlConnection(connectionString)
sql = "SELECT OrderID, CustomerID, EmployeeID, OrderDate, Freight," & "ShipName, ShipCountry
FROM Orders"

        da = New SqlDataAdapter(sql, conn)
        conn.Open()
        ds = New DataSet()
Dim commandBuilder As SqlCommandBuilder = New SqlCommandBuilder(da)
        da.Fill(ds, "Orders")
        bsource.DataSource = ds.Tables("Orders")
        dgv.DataSource = bsource
        End Sub
End Class

```

Tip 2 – Update the data in the DataGridView and save changes in the database / Aktualizacji danych w DataGridView i zapisać zmiany w bazie danych /

After editing the data in the cells, if you would like to update the changes permanently in the database, use the following code:

C#

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    DataTable dt = ds.Tables["Orders"];
    this.dgv.BindingContext[dt].EndCurrentEdit();
    this.da.Update(dt);
}

```

VB.NET

```

Private Sub btnUpdate_Click(ByVal sender As Object, ByVal e As EventArgs)
    Dim dt As DataTable = ds.Tables("Orders")
    Me.dgv.BindingContext(dt).EndCurrentEdit()
    Me.da.Update(dt)
End Sub

```

Tip 3 – Display a confirmation box before deleting a row in the DataGridView / Wyświetlić okno potwierdzenia przed usunięciem wiersza w DataGridView /

Handle the UserDeletingRow event to display a confirmation box to the user. If the user confirms the deletion, delete the row. If the user clicks cancel, set e.cancel = true which cancels the row deletion.

C#

```

private void dgv_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e)
{
    if (!e.Row.IsNewRow)
    {
        DialogResult res = MessageBox.Show("Are you sure you want to delete this
row?", "Delete confirmation",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (res == DialogResult.No)
            e.Cancel = true;
    }
}

```

VB.NET

```

Private Sub dgv_UserDeletingRow(ByVal sender As Object, ByVal e As
DataGridViewRowCancelEventArgs)
    If (Not e.Row.IsNewRow) Then
        Dim res As DialogResult = MessageBox.Show("Are you sure you want to
delete this row?", "Delete confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
        If res = DialogResult.No Then
            e.Cancel = True
        End If
    End If
End Sub

```

Tip 4 - How to autoresize column width in the DataGridView / Jak automatycznie dopasować szerokość kolumny w DataGridView /

The snippet shown below, first auto-resizes the columns to fit its content. Then the `AutoSizeColumnsMode` is set to the `'DataGridViewAutoSizeColumnsMode.AllCells'` enumeration value which automatically adjust the widths of the columns when the data changes.

```
C#
private void btnResize_Click(object sender, EventArgs e)
{
    dgv.AutoSizeColumns();
    dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
}

VB.NET
Private Sub btnResize_Click(ByVal sender As Object, ByVal e As EventArgs)
    dgv.AutoSizeColumns()
    dgv.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells
End Sub
```

Tip 5 - Select and Highlight an entire row in DataGridView / Wybierz i zaznacz całą wiersz w DataGridView /

```
C#
int rowToBeSelected = 3; // third row
if (dgv.Rows.Count >= rowToBeSelected)
{
    // Since index is zero based, you have to subtract 1
    dgv.Rows[rowToBeSelected - 1].Selected = true;
}

VB.NET
Dim rowToBeSelected As Integer = 3 ' third row
If dgv.Rows.Count >= rowToBeSelected Then
    ' Since index is zero based, you have to subtract 1
    dgv.Rows(rowToBeSelected - 1).Selected = True
End If
```

Tip 6 - How to scroll programmatically to a row in the DataGridView / Jak programowo przewinąć do zadanego wiersza w DataGridView /

The `DataGridView` has a property called `FirstDisplayedScrollingRowIndex` that can be used in order to scroll to a row programmatically.

```
C#
int jumpToRow = 20;
if (dgv.Rows.Count >= jumpToRow && jumpToRow >= 1)
{
    dgv.FirstDisplayedScrollingRowIndex = jumpToRow;
    dgv.Rows[jumpToRow].Selected = true;
}

VB.NET
Dim jumpToRow As Integer = 20
If dgv.Rows.Count >= jumpToRow AndAlso jumpToRow >= 1 Then
    dgv.FirstDisplayedScrollingRowIndex = jumpToRow
    dgv.Rows(jumpToRow).Selected = True
End If
```

Tip 7 - Calculate a column total in the DataGridView and display in a textbox / Obliczyć łączną kolumny w DataGridView i wyświetlać w polu tekstowym /

A common requirement is to calculate the total of a currency field and display it in a textbox. In the snippet below, we will be calculating the total of the 'Freight' field. We will then display the data in a textbox by formatting the result (observe the `ToString("c")`) while displaying the data, which displays the culture-specific currency.

```
C#
```

```

private void btnTotal_Click(object sender, EventArgs e)
{
    if (dgv.Rows.Count > 0)
        txtTotal.Text = Total().ToString("c");
}
private double Total()
{
    double tot = 0;
    int i = 0;
    for (i = 0; i < dgv.Rows.Count; i++)
    {
        tot = tot + Convert.ToDouble(dgv.Rows[i].Cells["Freight"].Value);
    }
    return tot;
}

```

VB.NET

```

Private Sub btnTotal_Click(ByVal sender As Object, ByVal e As EventArgs)
    If dgv.Rows.Count > 0 Then
        txtTotal.Text = Total().ToString("c")
    End If
End Sub

```

```

Private Function Total() As Double
    Dim tot As Double = 0
    Dim i As Integer = 0
    For i = 0 To dgv.Rows.Count - 1
        tot = tot + Convert.ToDouble(dgv.Rows(i).Cells("Freight").Value)
    Next i
    Return tot
End Function

```

Tip 8 - Change the Header Names in the DataGridView / Zmiana nazw nagłówek w DataGridView /

If the columns being retrieved from the database do not have meaningful names, we always have the option of changing the header names as shown in this snippet:

C#

```

private void btnChange_Click(object sender, EventArgs e)
{
    dgv.Columns[0].HeaderText = "MyHeader1";
    dgv.Columns[1].HeaderText = "MyHeader2";
}

```

VB.NET

```

Private Sub btnChange_Click(ByVal sender As Object, ByVal e As EventArgs)
    dgv.Columns(0).HeaderText = "MyHeader1"
    dgv.Columns(1).HeaderText = "MyHeader2"
End Sub

```

Tip 9 - Change the Color of Cells, Rows and Border in the DataGridView / Zmień kolor komórek, wierszy i Granicznej w DataGridView /

C#

```

private void btnCellRow_Click(object sender, EventArgs e)
{
    // Change ForeColor of each Cell
    this.dgv.DefaultCellStyle.ForeColor = Color.Coral;
    // Change back color of each row
    this.dgv.RowsDefaultCellStyle.BackColor = Color.AliceBlue;
    // Change GridLine Color
    this.dgv.GridColor = Color.Blue;
    // Change Grid Border Style
    this.dgv.BorderStyle = BorderStyle.Fixed3D;
}

```

VB.NET

```

Private Sub btnCellRow_Click(ByVal sender As Object, ByVal e As EventArgs)
    ' Change ForeColor of each Cell
    Me.dgv.DefaultCellStyle.ForeColor = Color.Coral
    ' Change back color of each row
    Me.dgv.RowsDefaultCellStyle.BackColor = Color.AliceBlue

```

```

' Change GridLine Color
Me.dgv.GridColor = Color.Blue
' Change Grid Border Style
Me.dgv.BorderStyle = BorderStyle.Fixed3D

```

End Sub

Tip 10 - Hide a Column in the DataGridView / Ukrywanie kolumny w DataGridView /

If you would like to hide a column based on a certain condition, here's a snippet for that.

C#

```

private void btnHide_Click(object sender, EventArgs e)
{
    this.dgv.Columns["EmployeeID"].Visible = false;
}

```

VB.NET

```

Private Sub btnHide_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.dgv.Columns("EmployeeID").Visible = False

```

End Sub

Tip 11 - Handle SelectedIndexChanged of a ComboBox in the DataGridView / Uchwył SelectedIndexChanged z ComboBox w DataGridView /

To handle the SelectedIndexChanged event of a DataGridViewComboBox, you need to use the DataGridView.EditingControlShowing event as shown below. You can then retrieve the selected index or the selected text of the combobox.

C#

```

private void dataGridView1_EditingControlShowing(object sender,
DataGridViewEditingControlShowingEventArgs e)
{
    ComboBox editingComboBox = (ComboBox)e.Control;
    if(editingComboBox != null)
        editingComboBox.SelectedIndexChanged += new
System.EventHandler(this.editingComboBox_SelectedIndexChanged);
}
private void editingComboBox_SelectedIndexChanged(object sender, System.EventArgs e)
{
    ComboBox comboBox1 = (ComboBox)sender;
    // Display index
    MessageBox.Show(comboBox1.SelectedIndex.ToString());
    // Display value
    MessageBox.Show(comboBox1.Text);
}

```

VB.NET

```

Private Sub dataGridView1_EditingControlShowing(ByVal sender As Object, ByVal e As
DataGridViewEditingControlShowingEventArgs)
    Dim editingComboBox As ComboBox = CType(e.Control, ComboBox)
    If Not editingComboBox Is Nothing Then
        AddHandler editingComboBox.SelectedIndexChanged, AddressOf
editingComboBox_SelectedIndexChanged
    End If

```

End Sub

```

Private Sub editingComboBox_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs)
    Dim comboBox1 As ComboBox = CType(sender, ComboBox)
    ' Display index
    MessageBox.Show(comboBox1.SelectedIndex.ToString())
    ' Display value
    MessageBox.Show(comboBox1.Text)

```

End Sub

Tip 12 - Change Color of Alternate Rows in the DataGridView / Zmień kolor co drugiego wiersza w DataGridView /

C#

```

private void btnAlternate_Click(object sender, EventArgs e)
{
    this.dgv.RowsDefaultCellStyle.BackColor = Color.White;
}

```

```

        this.dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Aquamarine;
    }
VB.NET
Private Sub btnAlternate_Click(ByVal sender As Object, ByVal e As EventArgs)
    Me.dgv.RowsDefaultCellStyle.BackColor = Color.White
    Me.dgv.AlternatingRowsDefaultCellStyle.BackColor = Color.Aquamarine
End Sub

```

Tip 13 - Formatting Data in the DataGridView / Formatowanie danych w DataGridView /

The DataGridView exposes properties that enable you to format data such as displaying a currency column in the culture specific currency or displaying nulls in a desired format and so on.

```

C#
private void btnFormat_Click(object sender, EventArgs e)
{
    // display currency in culture-specific currency for
    this.dgv.Columns["Freight"].DefaultCellStyle.Format = "c";
    // display nulls as 'NA'
    this.dgv.DefaultCellStyle.NullValue = "NA";
}

```

```

VB.NET
Private Sub btnFormat_Click(ByVal sender As Object, ByVal e As EventArgs)
    ' display currency in culture-specific currency for
    Me.dgv.Columns("Freight").DefaultCellStyle.Format = "c"
    ' display nulls as 'NA'
    Me.dgv.DefaultCellStyle.NullValue = "NA"
End Sub

```

Tip 14 - Change the order of columns in the DataGridView / Zmienić kolejność kolumn w DataGridView /

In order to change the order of columns, just set the DisplayIndex property of the DataGridView to the desired value. Remember that the index is zero based.

```

C#
private void btnReorder_Click(object sender, EventArgs e)
{
    dgv.Columns["CustomerID"].DisplayIndex = 5;
    dgv.Columns["OrderID"].DisplayIndex = 3;
    dgv.Columns["EmployeeID"].DisplayIndex = 1;
    dgv.Columns["OrderDate"].DisplayIndex = 2;
    dgv.Columns["Freight"].DisplayIndex = 6;
    dgv.Columns["ShipCountry"].DisplayIndex = 0;
    dgv.Columns["ShipName"].DisplayIndex = 4;
}

```

```

VB.NET
Private Sub btnReorder_Click(ByVal sender As Object, ByVal e As EventArgs)
    dgv.Columns("CustomerID").DisplayIndex = 5
    dgv.Columns("OrderID").DisplayIndex = 3
    dgv.Columns("EmployeeID").DisplayIndex = 1
    dgv.Columns("OrderDate").DisplayIndex = 2
    dgv.Columns("Freight").DisplayIndex = 6
    dgv.Columns("ShipCountry").DisplayIndex = 0
    dgv.Columns("ShipName").DisplayIndex = 4
End Sub

```

I hope this article was useful and I thank you for viewing it.