

# How to export to winforms datagrid in VB.NET using ByteScout Spreadsheet SDK

The tutorial below will demonstrate how to export to winforms datagrid in VB.NET

Sample source code below will show you how to cope with a difficult task like export to winforms datagrid in VB.NET. ByteScout Spreadsheet SDK is the SDK component for writing, reading, modifying and calculating Excel and CSV spreadsheets. Can calculate and recalculate formulas with Excel installed. You may import or export data to and from CSV, XML, JSON. Supports export to databases, arrays, streams and you can use it to export to winforms datagrid with VB.NET.

Fast application programming interfaces of ByteScout Spreadsheet SDK for VB.NET plus the instruction and the code below will help you quickly learn how to export to winforms datagrid. In order to implement the functionality, you should copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. You can use these VB.NET sample examples in one or many applications.

You can download free trial version of ByteScout Spreadsheet SDK from our website to see and try many others source code samples for VB.NET.

VB.NET - Form1.Designer.vb

```
Partial Class Form1
    '''
    ''' Required designer variable.
    '''

    Private components As System.ComponentModel.IContainer = Nothing

    '''
    ''' Clean up any resources being used.
    '''

    ''' true if managed resources should be disposed; otherwise, false.
    Protected Overrides Sub Dispose(disposing As Boolean)
        If disposing AndAlso (components IsNot Nothing) Then
            components.Dispose()
        End If
        MyBase.Dispose(disposing)
    End Sub

    #Region "Windows Form Designer generated code"
```

```

'''
''' Required method for Designer support - do not modify
''' the contents of this method with the code editor.
'''

Private Sub InitializeComponent()
    Me.dataGrid1 = New System.Windows.Forms.DataGrid()
    Me.dataGridTableStyle1 = New
System.Windows.Forms.DataGridTableStyle()
    Me.dataGridTextBoxColumn1 = New
System.Windows.Forms.DataGridTextBoxColumn()
    Me.btnImport = New System.Windows.Forms.Button()
    Me.btnExport = New System.Windows.Forms.Button()
    Me.openFileDialog1 = New System.Windows.Forms.OpenFileDialog()
    DirectCast(Me.dataGrid1,
System.ComponentModel.ISupportInitialize).BeginInit()
    Me.SuspendLayout()
    '
    ' dataGrid1
    '
    Me.dataGrid1.Anchor = CType((((System.Windows.Forms.AnchorStyles.Top
Or System.Windows.Forms.AnchorStyles.Bottom) Or
System.Windows.Forms.AnchorStyles.Left) Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)
    Me.dataGrid1.DataMember = ""
    Me.dataGrid1.HeaderForeColor =
System.Drawing.SystemColors.ControlText
    Me.dataGrid1.Location = New System.Drawing.Point(12, 12)
    Me.dataGrid1.Name = "dataGrid1"
    Me.dataGrid1.Size = New System.Drawing.Size(438, 308)
    Me.dataGrid1.TabIndex = 0
    Me.dataGrid1.TableStyles.AddRange(New
System.Windows.Forms.DataGridTableStyle() {Me.dataGridTableStyle1})
    '
    ' dataGridTableStyle1
    '
    Me.dataGridTableStyle1.DataGrid = Me.dataGrid1
    Me.dataGridTableStyle1.GridColumnStyles.AddRange(New
System.Windows.Forms.DataGridColumnStyle() {Me.dataGridTextBoxColumn1})
    Me.dataGridTableStyle1.HeaderForeColor =
System.Drawing.SystemColors.ControlText
    '
    ' dataGridTextBoxColumn1
    '
    Me.dataGridTextBoxColumn1.Format = ""
    Me.dataGridTextBoxColumn1.FormatInfo = Nothing
    Me.dataGridTextBoxColumn1.NullText = ""
    Me.dataGridTextBoxColumn1.Width = 75
    '
    ' btnImport
    '
    Me.btnImport.Anchor = CType((System.Windows.Forms.AnchorStyles.Bottom
Or System.Windows.Forms.AnchorStyles.Right), System.Windows.Forms.AnchorStyles)
    Me.btnImport.Location = New System.Drawing.Point(238, 326)
    Me.btnImport.Name = "btnImport"
    Me.btnImport.Size = New System.Drawing.Size(212, 23)
    Me.btnImport.TabIndex = 1
    Me.btnImport.Text = "Import from DataGrid to spreadsheet"
    Me.btnImport.UseVisualStyleBackColor = True

```

```

        AddHandler Me.btnImport.Click, New System.EventHandler(AddressOf
Me.btnImport_Click)
    ,
    , btnExport
    ,
        Me.btnExport.Anchor = CType((System.Windows.Forms.AnchorStyles.Bottom
Or System.Windows.Forms.AnchorStyles.Right), System.Windows.Forms.AnchorStyles)
        Me.btnExport.Location = New System.Drawing.Point(238, 355)
        Me.btnExport.Name = "btnExport"
        Me.btnExport.Size = New System.Drawing.Size(212, 23)
        Me.btnExport.TabIndex = 2
        Me.btnExport.Text = "Export from spreadsheet to DataGrid"
        Me.btnExport.UseVisualStyleBackColor = True
        AddHandler Me.btnExport.Click, New System.EventHandler(AddressOf
Me.btnExport_Click)
    ,
    , openFileDialog1
    ,
        Me.openFileDialog1.FileName = "openFileDialog1"
        Me.openFileDialog1.Filter = "Excel files
(*.xls;*.xlsx)|*.xls;*.xlsx|All files|*.*"
    ,
    , Form1
    ,
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6F, 13F)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleModeMode.Font
        Me.ClientSize = New System.Drawing.Size(462, 390)
        Me.Controls.Add(Me.btnExport)
        Me.Controls.Add(Me.btnImport)
        Me.Controls.Add(Me.dataGrid1)
        Me.Name = "Form1"
        Me.Text = "ImportFromWinFormsDataGrid"
        DirectCast(Me.dataGrid1,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)

    End Sub

#End Region

Private dataGrid1 As System.Windows.Forms.DataGrid
Private btnImport As System.Windows.Forms.Button
Private btnExport As System.Windows.Forms.Button
Private openFileDialog1 As System.Windows.Forms.OpenFileDialog
Private dataGridTableStyle1 As System.Windows.Forms.DataGridTableStyle
Private dataGridTextBoxColumn1 As System.Windows.Forms.DataGridTextBoxColumn

End Class

```

```

Imports System.Data
Imports System.Diagnostics
Imports System.IO
Imports System.Windows.Forms
Imports Bytescout.Spreadsheet

Public Partial Class Form1
    Inherits Form
    Public Sub New()
        InitializeComponent()

        dataGrid1.DataSource = GetDemoDataTable()
    End Sub

    Private Sub btnImport_Click(sender As Object, e As EventArgs)
        Cursor = Cursors.WaitCursor

        Try
            Const fileName As String = "CSharpImportFromDataTable.xls"

            ' retrieve DataTable from DataGridView
            Dim dataTable As DataTable = DirectCast(dataGrid1.DataSource,
DataTable)

            ' Create spreadsheet
            Dim spreadsheet As New Spreadsheet()

            ' Import data from DataTable into spreadsheet
            spreadsheet.ImportFromDataTable(dataTable)

            ' Insert row with column captions
            Dim worksheet As Worksheet = spreadsheet.Worksheets(0)
            worksheet.Rows.Insert(0)
            For colIndex As Integer = 0 To dataTable.Columns.Count - 1
                worksheet.Cell(0, colIndex).Value =
dataTable.Columns(colIndex).Caption
            Next

            ' Save the spreadsheet
            If File.Exists(fileName) Then
                File.Delete(fileName)
            End If
            spreadsheet.SaveAs(fileName)

            ' Close spreadsheet
            spreadsheet.Close()

            ' Open the spreadsheet
            Process.Start(fileName)
        Finally
            Cursor = Cursors.[Default]
        End Try
    End Sub

    Private Sub btnExport_Click(sender As Object, e As EventArgs)
        If openFileDialog1.ShowDialog() = DialogResult.OK Then
            Cursor = Cursors.WaitCursor

            Try

```

```

        ' Create spreadsheet
        Dim spreadsheet As New Spreadsheet()

        ' Load spreadsheet from file
        spreadsheet.LoadFromFile(openFileDialog1.FileName)

        ' Export to DataGridView
        Dim dataTable As DataTable =
spreadsheet.ExportToDataTable()
        dataGrid1.DataSource = dataTable
        Finally
            Cursor = Cursors.[Default]
        End Try
    End If
End Sub

'''
''' Fills a data table of the periodic table of elements.
'''

Private Function GetDemoDataTable() As DataTable
    Dim periodicTable As New DataTable("PeriodicTable")

    periodicTable.Columns.Add("Name", GetType(String))
    periodicTable.Columns.Add("Symbol", GetType(String))
    periodicTable.Columns.Add("AtomicNumber", GetType(Integer))

    Dim dr As DataRow = periodicTable.Rows.Add()
    dr(0) = "Hydrogen"
    dr(1) = "H"
    dr(2) = "1"

    dr = periodicTable.Rows.Add()
    dr(0) = "Helium"
    dr(1) = "He"
    dr(2) = "2"

    dr = periodicTable.Rows.Add()
    dr(0) = "Lithium"
    dr(1) = "Li"
    dr(2) = "3"

    dr = periodicTable.Rows.Add()
    dr(0) = "Beryllium"
    dr(1) = "Be"
    dr(2) = "4"

    dr = periodicTable.Rows.Add()
    dr(0) = "Boron"
    dr(1) = "B"
    dr(2) = "5"

    dr = periodicTable.Rows.Add()
    dr(0) = "Carbon"
    dr(1) = "C"
    dr(2) = "6"

    Return periodicTable
End Function

```

End Class

VB.NET - Program.vb

```
Imports System.Windows.Forms

Class Program
    Friend Shared Sub Main(args As String())
        Application.EnableVisualStyles()
        Application.SetCompatibleTextRenderingDefault(False)
        Application.Run(New Form1())
    End Sub
End Class
```

---

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Spreadsheet SDK](#)

[Explore documentation](#)

[Visit www.ByteScout.com](http://www.ByteScout.com)

or

[Get Your Free API Key for www.PDF.co Web API](#)