www.bytescout.com

How to generate barcode in crystalreports application vb with barcode sdk in Crystal Reports using ByteScout Premium Suite

Learn to generate barcode in crystalreports application vb with barcode sdk in Crystal Reports

The code displayed below will guide you to install an Crystal Reports app to generate barcode in crystalreports application vb with barcode sdk. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording and you can use it to generate barcode in crystalreports application vb with barcode sdk with Crystal Reports.

These Crystal Reports code samples for Crystal Reports guide developers to speed up coding of the application when using ByteScout Premium Suite. Follow the instructions from scratch to work and copy the Crystal Reports code. Applying Crystal Reports application mostly includes various stages of the software development so even if the functionality works please test it with your data and the production environment.

If you want to try other source code samples then the free trial version of ByteScout Premium Suite is available for download from our website. Just try other source code samples for Crystal Reports.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Premium Suite

Explore API Documentation

Get Free Training for ByteScout Premium Suite

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

CrystalReport1.vb

```
'------------------------------------------------------------------------------
' <auto-generated>
'     This code was generated by a tool.
'     Runtime Version:4.0.30319.34014
'
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </auto-generated>
'------------------------------------------------------------------------------

Option Strict Off
Option Explicit On

Imports CrystalDecisions.CrystalReports.Engine
Imports CrystalDecisions.ReportSource
Imports CrystalDecisions.Shared
Imports System
Imports System.ComponentModel


Public Class CrystalReport1
    Inherits ReportClass

    Public Sub New()
        MyBase.New
    End Sub

    Public Overrides Property ResourceName() As String
        Get
            Return "CrystalReport1.rpt"
        End Get
        Set
            'Do nothing
        End Set
    End Property

    Public Overrides Property NewGenerator() As Boolean
        Get
            Return true
        End Get
        Set
            'Do nothing
        End Set
    End Property

    Public Overrides Property FullResourceName() As String
        Get
            Return "BarcodeInCrystalReports.CrystalReport1.rpt"
        End Get
        Set
            'Do nothing
        End Set
    End Property

    <Browsable(false),  _
```

```
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public ReadOnly Property Section1() As CrystalDecisions.CrystalReports.Engine.Sect
        Get
            Return Me.ReportDefinition.Sections(0)
        End Get
    End Property

    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public ReadOnly Property Section2() As CrystalDecisions.CrystalReports.Engine.Sect
        Get
            Return Me.ReportDefinition.Sections(1)
        End Get
    End Property

    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public ReadOnly Property Section3() As CrystalDecisions.CrystalReports.Engine.Sect
        Get
            Return Me.ReportDefinition.Sections(2)
        End Get
    End Property

    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public ReadOnly Property Section4() As CrystalDecisions.CrystalReports.Engine.Sect
        Get
            Return Me.ReportDefinition.Sections(3)
        End Get
    End Property

    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public ReadOnly Property Section5() As CrystalDecisions.CrystalReports.Engine.Sect
        Get
            Return Me.ReportDefinition.Sections(4)
        End Get
    End Property
End Class

<System.Drawing.ToolboxBitmapAttribute(GetType(CrystalDecisions.[Shared].ExportOptions)
Public Class CachedCrystalReport1
    Inherits Component
    Implements ICachedReport

    Public Sub New()
        MyBase.New
    End Sub

    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public Overridable Property IsCacheable() As Boolean Implements CrystalDecisions.Re
        Get
            Return true
        End Get
        Set
            '
        End Set
    End Property
```

```
    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public Overridable Property ShareDBLogonInfo() As Boolean Implements CrystalDecisi
        Get
            Return false
        End Get
        Set
            '
        End Set
    End Property

    <Browsable(false),  _
     DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializat
    Public Overridable Property CacheTimeOut() As System.TimeSpan Implements CrystalDe
        Get
            Return CachedReportConstants.DEFAULT_TIMEOUT
        End Get
        Set
            '
        End Set
    End Property

    Public Overridable Function CreateReport() As CrystalDecisions.CrystalReports.Engi
        Dim rpt As CrystalReport1 = New CrystalReport1()
        rpt.Site = Me.Site
        Return rpt
    End Function

    Public Overridable Function GetCustomizedCacheKey(ByVal request As RequestContext)
        Dim key As [String] = Nothing
        '// The following is the code used to generate the default
        '// cache key for caching report jobs in the ASP.NET Cache.
        '// Feel free to modify this code to suit your needs.
        '// Returning key == null causes the default cache key to
        '// be generated.
        '
        'key = RequestContext.BuildCompleteCacheKey(
        '    request,
        '    null,       // sReportFilename
        '    this.GetType(),
        '    this.ShareDBLogonInfo );
        Return key
    End Function
End Class
```

Form1.Designer.vb

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Form1
    Inherits System.Windows.Forms.Form
```

```
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    Private components As System.ComponentModel.IContainer

    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.CrystalReportViewer1 = New CrystalDecisions.Windows.Forms.CrystalReportViewe
        Me.CrystalReport11 = New BarcodeInCrystalReports.CrystalReport1()
        Me.SuspendLayout()
        '
        'CrystalReportViewer1
        '
        Me.CrystalReportViewer1.ActiveViewIndex = 0
        Me.CrystalReportViewer1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSi
        Me.CrystalReportViewer1.Cursor = System.Windows.Forms.Cursors.Default
        Me.CrystalReportViewer1.Dock = System.Windows.Forms.DockStyle.Fill
        Me.CrystalReportViewer1.Location = New System.Drawing.Point(0, 0)
        Me.CrystalReportViewer1.Name = "CrystalReportViewer1"
        Me.CrystalReportViewer1.ReportSource = Me.CrystalReport11
        Me.CrystalReportViewer1.Size = New System.Drawing.Size(792, 566)
        Me.CrystalReportViewer1.TabIndex = 0
        '
        'Form1
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.ClientSize = New System.Drawing.Size(792, 566)
        Me.Controls.Add(Me.CrystalReportViewer1)
        Me.Name = "Form1"
        Me.Text = "Form1"
        Me.ResumeLayout(False)

    End Sub
    Friend WithEvents CrystalReportViewer1 As CrystalDecisions.Windows.Forms.CrystalRep
    Friend WithEvents CrystalReport11 As CrystalReport1

End Class
```

Form1.vb

```
```

```
Imports System.Data.OleDb
Imports Bytescout.BarCode

Public Class Form1

    Private Sub CrystalReportViewer1_Load(sender As System.Object, e As System.EventArg

        Dim connection As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data So
        Dim dataAdapter As New OleDbDataAdapter("SELECT ID, ProductName, ProductDescrip

        ' fill dataset
        Dim dataSet As New DataSet()
        dataAdapter.Fill(dataSet)

        connection.Close()

        'add virtual column into the result table
        dataSet.Tables(0).Columns.Add(New DataColumn("BarcodeImage", GetType(System.Byt

        ' create barcode object
        Dim barcode As New Barcode(SymbologyType.Code128)
        barcode.DrawCaption = False

        ' Fill BarcodeImage column with generated barcode image bytes
        For Each row As DataRow In dataSet.Tables(0).Rows

            'set barcode value
            barcode.Value = Convert.ToString(row("ID"))

            ' retrieve generated image bytes
            Dim barcodeBytes As Byte() = barcode.GetImageBytesWMF()

            ' fill virtual column with generated image bytes
            row("BarcodeImage") = barcodeBytes

        Next

        ' set filled DataSet as report's data source
        CrystalReport11.SetDataSource(dataSet.Tables(0))

    End Sub
End Class
```

ProductsDataSet.Designer.vb

```
'------------------------------------------------------------------------------
' <auto-generated>
'     This code was generated by a tool.
'     Runtime Version:4.0.30319.34014
'
```

```
'     Changes to this file may cause incorrect behavior and will be lost if
'     the code is regenerated.
' </auto-generated>
'------------------------------------------------------------------------------

Option Strict Off
Option Explicit On



'''<summary>
'''Represents a strongly typed in-memory cache of data.
'''</summary>
<Global.System.Serializable(),  _
 Global.System.ComponentModel.DesignerCategoryAttribute("code"),  _
 Global.System.ComponentModel.ToolboxItem(true),  _
 Global.System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedDataSetSchema"),
 Global.System.Xml.Serialization.XmlRootAttribute("ProductsDataSet"),  _
 Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.DataSet")>  _
Partial Public Class ProductsDataSet
    Inherits Global.System.Data.DataSet

    Private tableProducts As ProductsDataTable

    Private _schemaSerializationMode As Global.System.Data.SchemaSerializationMode = G

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Public Sub New()
        MyBase.New
        Me.BeginInit
        Me.InitClass
        Dim schemaChangedHandler As Global.System.ComponentModel.CollectionChangeEventH
        AddHandler MyBase.Tables.CollectionChanged, schemaChangedHandler
        AddHandler MyBase.Relations.CollectionChanged, schemaChangedHandler
        Me.EndInit
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Protected Sub New(ByVal info As Global.System.Runtime.Serialization.SerializationIn
        MyBase.New(info, context, false)
        If (Me.IsBinarySerialized(info, context) = true) Then
            Me.InitVars(false)
            Dim schemaChangedHandler1 As Global.System.ComponentModel.CollectionChangeE
            AddHandler Me.Tables.CollectionChanged, schemaChangedHandler1
            AddHandler Me.Relations.CollectionChanged, schemaChangedHandler1
            Return
        End If
        Dim strSchema As String = CType(info.GetValue("XmlSchema", GetType(String)),Str
        If (Me.DetermineSchemaSerializationMode(info, context) = Global.System.Data.Sch
            Dim ds As Global.System.Data.DataSet = New Global.System.Data.DataSet()
            ds.ReadXmlSchema(New Global.System.Xml.XmlTextReader(New Global.System.IO.S
            If (Not (ds.Tables("Products")) Is Nothing) Then
                MyBase.Tables.Add(New ProductsDataTable(ds.Tables("Products")))
            End If
            Me.DataSetName = ds.DataSetName
            Me.Prefix = ds.Prefix
            Me.Namespace = ds.Namespace
            Me.Locale = ds.Locale
```

```
            Me.CaseSensitive = ds.CaseSensitive
            Me.EnforceConstraints = ds.EnforceConstraints
            Me.Merge(ds, false, Global.System.Data.MissingSchemaAction.Add)
            Me.InitVars
        Else
            Me.ReadXmlSchema(New Global.System.Xml.XmlTextReader(New Global.System.IO.S
        End If
        Me.GetSerializationData(info, context)
        Dim schemaChangedHandler As Global.System.ComponentModel.CollectionChangeEventH
        AddHandler MyBase.Tables.CollectionChanged, schemaChangedHandler
        AddHandler Me.Relations.CollectionChanged, schemaChangedHandler
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
     Global.System.ComponentModel.Browsable(false),  _
     Global.System.ComponentModel.DesignerSerializationVisibility(Global.System.Compone
    Public ReadOnly Property Products() As ProductsDataTable
        Get
            Return Me.tableProducts
        End Get
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
     Global.System.ComponentModel.BrowsableAttribute(true),  _
     Global.System.ComponentModel.DesignerSerializationVisibilityAttribute(Global.Syste
    Public Overrides Property SchemaSerializationMode() As Global.System.Data.SchemaSer
        Get
            Return Me._schemaSerializationMode
        End Get
        Set
            Me._schemaSerializationMode = value
        End Set
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
     Global.System.ComponentModel.DesignerSerializationVisibilityAttribute(Global.Syste
    Public Shadows ReadOnly Property Tables() As Global.System.Data.DataTableCollectio
        Get
            Return MyBase.Tables
        End Get
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
     Global.System.ComponentModel.DesignerSerializationVisibilityAttribute(Global.Syste
    Public Shadows ReadOnly Property Relations() As Global.System.Data.DataRelationCol
        Get
            Return MyBase.Relations
        End Get
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
    Protected Overrides Sub InitializeDerivedDataSet()
        Me.BeginInit
        Me.InitClass
        Me.EndInit
```

```
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Public Overrides Function Clone() As Global.System.Data.DataSet
        Dim cln As ProductsDataSet = CType(MyBase.Clone,ProductsDataSet)
        cln.InitVars
        cln.SchemaSerializationMode = Me.SchemaSerializationMode
        Return cln
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Protected Overrides Function ShouldSerializeTables() As Boolean
        Return false
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Protected Overrides Function ShouldSerializeRelations() As Boolean
        Return false
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Protected Overrides Sub ReadXmlSerializable(ByVal reader As Global.System.Xml.XmlRe
        If (Me.DetermineSchemaSerializationMode(reader) = Global.System.Data.SchemaSer
            Me.Reset
            Dim ds As Global.System.Data.DataSet = New Global.System.Data.DataSet()
            ds.ReadXml(reader)
            If (Not (ds.Tables("Products")) Is Nothing) Then
                MyBase.Tables.Add(New ProductsDataTable(ds.Tables("Products")))
            End If
            Me.DataSetName = ds.DataSetName
            Me.Prefix = ds.Prefix
            Me.Namespace = ds.Namespace
            Me.Locale = ds.Locale
            Me.CaseSensitive = ds.CaseSensitive
            Me.EnforceConstraints = ds.EnforceConstraints
            Me.Merge(ds, false, Global.System.Data.MissingSchemaAction.Add)
            Me.InitVars
        Else
            Me.ReadXml(reader)
            Me.InitVars
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Protected Overrides Function GetSchemaSerializable() As Global.System.Xml.Schema.Xm
        Dim stream As Global.System.IO.MemoryStream = New Global.System.IO.MemoryStrea
        Me.WriteXmlSchema(New Global.System.Xml.XmlTextWriter(stream, Nothing))
        stream.Position = 0
        Return Global.System.Xml.Schema.XmlSchema.Read(New Global.System.Xml.XmlTextRe
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Friend Overloads Sub InitVars()
        Me.InitVars(true)
```

```
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Friend Overloads Sub InitVars(ByVal initTable As Boolean)
        Me.tableProducts = CType(MyBase.Tables("Products"),ProductsDataTable)
        If (initTable = true) Then
            If (Not (Me.tableProducts) Is Nothing) Then
                Me.tableProducts.InitVars
            End If
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Private Sub InitClass()
        Me.DataSetName = "ProductsDataSet"
        Me.Prefix = ""
        Me.Namespace = "http://tempuri.org/ProductsDataSet.xsd"
        Me.EnforceConstraints = true
        Me.SchemaSerializationMode = Global.System.Data.SchemaSerializationMode.Include
        Me.tableProducts = New ProductsDataTable()
        MyBase.Tables.Add(Me.tableProducts)
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Private Function ShouldSerializeProducts() As Boolean
        Return false
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Private Sub SchemaChanged(ByVal sender As Object, ByVal e As Global.System.Componer
        If (e.Action = Global.System.ComponentModel.CollectionChangeAction.Remove) Ther
            Me.InitVars
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDa
    Public Shared Function GetTypedDataSetSchema(ByVal xs As Global.System.Xml.Schema.X
        Dim ds As ProductsDataSet = New ProductsDataSet()
        Dim type As Global.System.Xml.Schema.XmlSchemaComplexType = New Global.System.X
        Dim sequence As Global.System.Xml.Schema.XmlSchemaSequence = New Global.System
        Dim any As Global.System.Xml.Schema.XmlSchemaAny = New Global.System.Xml.Schema
        any.Namespace = ds.Namespace
        sequence.Items.Add(any)
        type.Particle = sequence
        Dim dsSchema As Global.System.Xml.Schema.XmlSchema = ds.GetSchemaSerializable
        If xs.Contains(dsSchema.TargetNamespace) Then
            Dim s1 As Global.System.IO.MemoryStream = New Global.System.IO.MemoryStrea
            Dim s2 As Global.System.IO.MemoryStream = New Global.System.IO.MemoryStrea
            Try
                Dim schema As Global.System.Xml.Schema.XmlSchema = Nothing
                dsSchema.Write(s1)
                Dim schemas As Global.System.Collections.IEnumerator = xs.Schemas(dsSc
                Do While schemas.MoveNext
                    schema = CType(schemas.Current,Global.System.Xml.Schema.XmlSchema)
                    s2.SetLength(0)
```

```
                    schema.Write(s2)
                    If (s1.Length = s2.Length) Then
                        s1.Position = 0
                        s2.Position = 0

                        Do While ((s1.Position <> s1.Length)  _
                                    AndAlso (s1.ReadByte = s2.ReadByte))


                        Loop
                        If (s1.Position = s1.Length) Then
                            Return type
                        End If
                    End If

                Loop
            Finally
                If (Not (s1) Is Nothing) Then
                    s1.Close
                End If
                If (Not (s2) Is Nothing) Then
                    s2.Close
                End If
            End Try
        End If
        xs.Add(dsSchema)
        Return type
    End Function

    <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
    Public Delegate Sub ProductsRowChangeEventHandler(ByVal sender As Object, ByVal e A

    '''<summary>
    '''Represents the strongly named DataTable class.
    '''</summary>
    <Global.System.Serializable(),  _
     Global.System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")>
    Partial Public Class ProductsDataTable
        Inherits Global.System.Data.DataTable
        Implements Global.System.Collections.IEnumerable

        Private columnID As Global.System.Data.DataColumn

        Private columnProductName As Global.System.Data.DataColumn

        Private columnProductDescription As Global.System.Data.DataColumn

        Private columnProductPrice As Global.System.Data.DataColumn

        Private columnBarcodeImage As Global.System.Data.DataColumn

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Sub New()
            MyBase.New
            Me.TableName = "Products"
            Me.BeginInit
            Me.InitClass
            Me.EndInit
        End Sub
```

```
        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Friend Sub New(ByVal table As Global.System.Data.DataTable)
            MyBase.New
            Me.TableName = table.TableName
            If (table.CaseSensitive <> table.DataSet.CaseSensitive) Then
                Me.CaseSensitive = table.CaseSensitive
            End If
            If (table.Locale.ToString <> table.DataSet.Locale.ToString) Then
                Me.Locale = table.Locale
            End If
            If (table.Namespace <> table.DataSet.Namespace) Then
                Me.Namespace = table.Namespace
            End If
            Me.Prefix = table.Prefix
            Me.MinimumCapacity = table.MinimumCapacity
        End Sub

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Protected Sub New(ByVal info As Global.System.Runtime.Serialization.Serializat
            MyBase.New(info, context)
            Me.InitVars
        End Sub

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public ReadOnly Property IDColumn() As Global.System.Data.DataColumn
            Get
                Return Me.columnID
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public ReadOnly Property ProductNameColumn() As Global.System.Data.DataColumn
            Get
                Return Me.columnProductName
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public ReadOnly Property ProductDescriptionColumn() As Global.System.Data.Data
            Get
                Return Me.columnProductDescription
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public ReadOnly Property ProductPriceColumn() As Global.System.Data.DataColumn
            Get
                Return Me.columnProductPrice
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
```

```
        Public ReadOnly Property BarcodeImageColumn() As Global.System.Data.DataColumn
            Get
                Return Me.columnBarcodeImage
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
         Global.System.ComponentModel.Browsable(false)>  _
        Public ReadOnly Property Count() As Integer
            Get
                Return Me.Rows.Count
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Default ReadOnly Property Item(ByVal index As Integer) As ProductsRow
            Get
                Return CType(Me.Rows(index),ProductsRow)
            End Get
        End Property

        <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Event ProductsRowChanging As ProductsRowChangeEventHandler

        <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Event ProductsRowChanged As ProductsRowChangeEventHandler

        <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Event ProductsRowDeleting As ProductsRowChangeEventHandler

        <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Event ProductsRowDeleted As ProductsRowChangeEventHandler

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Overloads Sub AddProductsRow(ByVal row As ProductsRow)
            Me.Rows.Add(row)
        End Sub

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Overloads Function AddProductsRow(ByVal ProductName As String, ByVal Pro
            Dim rowProductsRow As ProductsRow = CType(Me.NewRow,ProductsRow)
            Dim columnValuesArray() As Object = New Object() {Nothing, ProductName, Pro
            rowProductsRow.ItemArray = columnValuesArray
            Me.Rows.Add(rowProductsRow)
            Return rowProductsRow
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Function FindByID(ByVal ID As Integer) As ProductsRow
            Return CType(Me.Rows.Find(New Object() {ID}),ProductsRow)
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Overridable Function GetEnumerator() As Global.System.Collections.IEnume
```

```
        Return Me.Rows.GetEnumerator
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Overrides Function Clone() As Global.System.Data.DataTable
        Dim cln As ProductsDataTable = CType(MyBase.Clone,ProductsDataTable)
        cln.InitVars
        Return cln
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Function CreateInstance() As Global.System.Data.DataTable
        Return New ProductsDataTable()
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Friend Sub InitVars()
        Me.columnID = MyBase.Columns("ID")
        Me.columnProductName = MyBase.Columns("ProductName")
        Me.columnProductDescription = MyBase.Columns("ProductDescription")
        Me.columnProductPrice = MyBase.Columns("ProductPrice")
        Me.columnBarcodeImage = MyBase.Columns("BarcodeImage")
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Private Sub InitClass()
        Me.columnID = New Global.System.Data.DataColumn("ID", GetType(Integer), Not
        MyBase.Columns.Add(Me.columnID)
        Me.columnProductName = New Global.System.Data.DataColumn("ProductName", Get
        MyBase.Columns.Add(Me.columnProductName)
        Me.columnProductDescription = New Global.System.Data.DataColumn("ProductDes
        MyBase.Columns.Add(Me.columnProductDescription)
        Me.columnProductPrice = New Global.System.Data.DataColumn("ProductPrice", C
        MyBase.Columns.Add(Me.columnProductPrice)
        Me.columnBarcodeImage = New Global.System.Data.DataColumn("BarcodeImage", C
        MyBase.Columns.Add(Me.columnBarcodeImage)
        Me.Constraints.Add(New Global.System.Data.UniqueConstraint("Constraint1", N
        Me.columnID.AutoIncrement = true
        Me.columnID.AutoIncrementSeed = -1
        Me.columnID.AutoIncrementStep = -1
        Me.columnID.AllowDBNull = false
        Me.columnID.Unique = true
        Me.columnProductName.MaxLength = 255
        Me.columnProductDescription.MaxLength = 255
        Me.columnProductPrice.MaxLength = 255
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Function NewProductsRow() As ProductsRow
        Return CType(Me.NewRow,ProductsRow)
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Function NewRowFromBuilder(ByVal builder As Global.System.D
```

```
        Return New ProductsRow(builder)
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Function GetRowType() As Global.System.Type
        Return GetType(ProductsRow)
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Sub OnRowChanged(ByVal e As Global.System.Data.DataRowChang
        MyBase.OnRowChanged(e)
        If (Not (Me.ProductsRowChangedEvent) Is Nothing) Then
            RaiseEvent ProductsRowChanged(Me, New ProductsRowChangeEvent(CType(e.Ro
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Sub OnRowChanging(ByVal e As Global.System.Data.DataRowChan
        MyBase.OnRowChanging(e)
        If (Not (Me.ProductsRowChangingEvent) Is Nothing) Then
            RaiseEvent ProductsRowChanging(Me, New ProductsRowChangeEvent(CType(e.R
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Sub OnRowDeleted(ByVal e As Global.System.Data.DataRowChang
        MyBase.OnRowDeleted(e)
        If (Not (Me.ProductsRowDeletedEvent) Is Nothing) Then
            RaiseEvent ProductsRowDeleted(Me, New ProductsRowChangeEvent(CType(e.Ro
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Protected Overrides Sub OnRowDeleting(ByVal e As Global.System.Data.DataRowChan
        MyBase.OnRowDeleting(e)
        If (Not (Me.ProductsRowDeletingEvent) Is Nothing) Then
            RaiseEvent ProductsRowDeleting(Me, New ProductsRowChangeEvent(CType(e.R
        End If
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Sub RemoveProductsRow(ByVal row As ProductsRow)
        Me.Rows.Remove(row)
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Shared Function GetTypedTableSchema(ByVal xs As Global.System.Xml.Schema
        Dim type As Global.System.Xml.Schema.XmlSchemaComplexType = New Global.Syst
        Dim sequence As Global.System.Xml.Schema.XmlSchemaSequence = New Global.Sys
        Dim ds As ProductsDataSet = New ProductsDataSet()
        Dim any1 As Global.System.Xml.Schema.XmlSchemaAny = New Global.System.Xml.S
        any1.Namespace = "http://www.w3.org/2001/XMLSchema"
        any1.MinOccurs = New Decimal(0)
```

```
            any1.MaxOccurs = Decimal.MaxValue
            any1.ProcessContents = Global.System.Xml.Schema.XmlSchemaContentProcessing
            sequence.Items.Add(any1)
            Dim any2 As Global.System.Xml.Schema.XmlSchemaAny = New Global.System.Xml.S
            any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1"
            any2.MinOccurs = New Decimal(1)
            any2.ProcessContents = Global.System.Xml.Schema.XmlSchemaContentProcessing
            sequence.Items.Add(any2)
            Dim attribute1 As Global.System.Xml.Schema.XmlSchemaAttribute = New Global
            attribute1.Name = "namespace"
            attribute1.FixedValue = ds.Namespace
            type.Attributes.Add(attribute1)
            Dim attribute2 As Global.System.Xml.Schema.XmlSchemaAttribute = New Global
            attribute2.Name = "tableTypeName"
            attribute2.FixedValue = "ProductsDataTable"
            type.Attributes.Add(attribute2)
            type.Particle = sequence
            Dim dsSchema As Global.System.Xml.Schema.XmlSchema = ds.GetSchemaSerializab
            If xs.Contains(dsSchema.TargetNamespace) Then
                Dim s1 As Global.System.IO.MemoryStream = New Global.System.IO.MemorySt
                Dim s2 As Global.System.IO.MemoryStream = New Global.System.IO.MemorySt
                Try
                    Dim schema As Global.System.Xml.Schema.XmlSchema = Nothing
                    dsSchema.Write(s1)
                    Dim schemas As Global.System.Collections.IEnumerator = xs.Schemas(d
                    Do While schemas.MoveNext
                        schema = CType(schemas.Current,Global.System.Xml.Schema.XmlSche
                        s2.SetLength(0)
                        schema.Write(s2)
                        If (s1.Length = s2.Length) Then
                            s1.Position = 0
                            s2.Position = 0

                            Do While ((s1.Position <> s1.Length)  _
                                        AndAlso (s1.ReadByte = s2.ReadByte))


                            Loop
                            If (s1.Position = s1.Length) Then
                                Return type
                            End If
                        End If

                    Loop
                Finally
                    If (Not (s1) Is Nothing) Then
                        s1.Close
                    End If
                    If (Not (s2) Is Nothing) Then
                        s2.Close
                    End If
                End Try
            End If
            xs.Add(dsSchema)
            Return type
        End Function
    End Class

    '''<summary>
    '''Represents strongly named DataRow class.
```

```
    '''</summary>
    Partial Public Class ProductsRow
        Inherits Global.System.Data.DataRow

        Private tableProducts As ProductsDataTable

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Friend Sub New(ByVal rb As Global.System.Data.DataRowBuilder)
            MyBase.New(rb)
            Me.tableProducts = CType(Me.Table,ProductsDataTable)
        End Sub

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Property ID() As Integer
            Get
                Return CType(Me(Me.tableProducts.IDColumn),Integer)
            End Get
            Set
                Me(Me.tableProducts.IDColumn) = value
            End Set
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Property ProductName() As String
            Get
                Try
                    Return CType(Me(Me.tableProducts.ProductNameColumn),String)
                Catch e As Global.System.InvalidCastException
                    Throw New Global.System.Data.StrongTypingException("The value for 
                End Try
            End Get
            Set
                Me(Me.tableProducts.ProductNameColumn) = value
            End Set
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Property ProductDescription() As String
            Get
                Try
                    Return CType(Me(Me.tableProducts.ProductDescriptionColumn),String)
                Catch e As Global.System.InvalidCastException
                    Throw New Global.System.Data.StrongTypingException("The value for 
                End Try
            End Get
            Set
                Me(Me.tableProducts.ProductDescriptionColumn) = value
            End Set
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Property ProductPrice() As String
            Get
                Try
                    Return CType(Me(Me.tableProducts.ProductPriceColumn),String)
```

```
            Catch e As Global.System.InvalidCastException
                Throw New Global.System.Data.StrongTypingException("The value for c
            End Try
        End Get
        Set
            Me(Me.tableProducts.ProductPriceColumn) = value
        End Set
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Property BarcodeImage() As Byte()
        Get
            Try
                Return CType(Me(Me.tableProducts.BarcodeImageColumn),Byte())
            Catch e As Global.System.InvalidCastException
                Throw New Global.System.Data.StrongTypingException("The value for c
            End Try
        End Get
        Set
            Me(Me.tableProducts.BarcodeImageColumn) = value
        End Set
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Function IsProductNameNull() As Boolean
        Return Me.IsNull(Me.tableProducts.ProductNameColumn)
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Sub SetProductNameNull()
        Me(Me.tableProducts.ProductNameColumn) = Global.System.Convert.DBNull
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Function IsProductDescriptionNull() As Boolean
        Return Me.IsNull(Me.tableProducts.ProductDescriptionColumn)
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Sub SetProductDescriptionNull()
        Me(Me.tableProducts.ProductDescriptionColumn) = Global.System.Convert.DBNu
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Function IsProductPriceNull() As Boolean
        Return Me.IsNull(Me.tableProducts.ProductPriceColumn)
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Sub SetProductPriceNull()
        Me(Me.tableProducts.ProductPriceColumn) = Global.System.Convert.DBNull
    End Sub
```

```
        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Function IsBarcodeImageNull() As Boolean
            Return Me.IsNull(Me.tableProducts.BarcodeImageColumn)
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Sub SetBarcodeImageNull()
            Me(Me.tableProducts.BarcodeImageColumn) = Global.System.Convert.DBNull
        End Sub
    End Class

    '''<summary>
    '''Row event argument class
    '''</summary>
    <Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDat
    Public Class ProductsRowChangeEvent
        Inherits Global.System.EventArgs

        Private eventRow As ProductsRow

        Private eventAction As Global.System.Data.DataRowAction

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Sub New(ByVal row As ProductsRow, ByVal action As Global.System.Data.Dat
            MyBase.New
            Me.eventRow = row
            Me.eventAction = action
        End Sub

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public ReadOnly Property Row() As ProductsRow
            Get
                Return Me.eventRow
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public ReadOnly Property Action() As Global.System.Data.DataRowAction
            Get
                Return Me.eventAction
            End Get
        End Property
    End Class
End Class

Namespace ProductsDataSetTableAdapters

    '''<summary>
    '''Represents the connection and commands used to retrieve and save data.
    '''</summary>
    <Global.System.ComponentModel.DesignerCategoryAttribute("code"),  _
     Global.System.ComponentModel.ToolboxItem(true),  _
     Global.System.ComponentModel.DataObjectAttribute(true),  _
     Global.System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.De
        ", Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"),  _
```

```
    Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")>
Partial Public Class ProductsTableAdapter
    Inherits Global.System.ComponentModel.Component

    Private WithEvents _adapter As Global.System.Data.OleDb.OleDbDataAdapter

    Private _connection As Global.System.Data.OleDb.OleDbConnection

    Private _commandCollection() As Global.System.Data.OleDb.OleDbCommand

    Private _clearBeforeFill As Boolean

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Public Sub New()
        MyBase.New
        Me.ClearBeforeFill = true
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Private ReadOnly Property Adapter() As Global.System.Data.OleDb.OleDbDataAdapte
        Get
            If (Me._adapter Is Nothing) Then
                Me.InitAdapter
            End If
            Return Me._adapter
        End Get
    End Property

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Friend Property Connection() As Global.System.Data.OleDb.OleDbConnection
        Get
            If (Me._connection Is Nothing) Then
                Me.InitConnection
            End If
            Return Me._connection
        End Get
        Set
            Me._connection = value
            If (Not (Me.Adapter.InsertCommand) Is Nothing) Then
                Me.Adapter.InsertCommand.Connection = value
            End If
            If (Not (Me.Adapter.DeleteCommand) Is Nothing) Then
                Me.Adapter.DeleteCommand.Connection = value
            End If
            If (Not (Me.Adapter.UpdateCommand) Is Nothing) Then
                Me.Adapter.UpdateCommand.Connection = value
            End If
            Dim i As Integer = 0
            Do While (i < Me.CommandCollection.Length)
                If (Not (Me.CommandCollection(i)) Is Nothing) Then
                    CType(Me.CommandCollection(i),Global.System.Data.OleDb.OleDbCo
                End If
                i = (i + 1)
            Loop
        End Set
    End Property
```

```
        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Protected ReadOnly Property CommandCollection() As Global.System.Data.OleDb.Ol
            Get
                If (Me._commandCollection Is Nothing) Then
                    Me.InitCommandCollection
                End If
                Return Me._commandCollection
            End Get
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Public Property ClearBeforeFill() As Boolean
            Get
                Return Me._clearBeforeFill
            End Get
            Set
                Me._clearBeforeFill = value
            End Set
        End Property

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
        Private Sub InitAdapter()
            Me._adapter = New Global.System.Data.OleDb.OleDbDataAdapter()
            Dim tableMapping As Global.System.Data.Common.DataTableMapping = New Globa
            tableMapping.SourceTable = "Table"
            tableMapping.DataSetTable = "Products"
            tableMapping.ColumnMappings.Add("ID", "ID")
            tableMapping.ColumnMappings.Add("ProductName", "ProductName")
            tableMapping.ColumnMappings.Add("ProductDescription", "ProductDescription")
            tableMapping.ColumnMappings.Add("ProductPrice", "ProductPrice")
            Me._adapter.TableMappings.Add(tableMapping)
            Me._adapter.DeleteCommand = New Global.System.Data.OleDb.OleDbCommand()
            Me._adapter.DeleteCommand.Connection = Me.Connection
            Me._adapter.DeleteCommand.CommandText = "DELETE FROM `Products` WHERE ((`I
                "R (`ProductName` = ?)) AND ((? = 1 AND `ProductDescription` IS NULL) (
                "tDescription` = ?)) AND ((? = 1 AND `ProductPrice` IS NULL) OR (`Produ
                " ?)))"
            Me._adapter.DeleteCommand.CommandType = Global.System.Data.CommandType.Tex
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.DeleteCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.InsertCommand = New Global.System.Data.OleDb.OleDbCommand()
            Me._adapter.InsertCommand.Connection = Me.Connection
            Me._adapter.InsertCommand.CommandText = "INSERT INTO `Products` (`ProductN
                "ES (?, ?, ?)"
            Me._adapter.InsertCommand.CommandType = Global.System.Data.CommandType.Tex
            Me._adapter.InsertCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.InsertCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.InsertCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
            Me._adapter.UpdateCommand = New Global.System.Data.OleDb.OleDbCommand()
            Me._adapter.UpdateCommand.Connection = Me.Connection
            Me._adapter.UpdateCommand.CommandText = "UPDATE `Products` SET `ProductNam
                " = ? WHERE ((`ID` = ?) AND ((? = 1 AND `ProductName` IS NULL) OR (`Pr
```

```
            "= ?)) AND ((? = 1 AND `ProductDescription` IS NULL) OR (`ProductDescr
            ")) AND ((? = 1 AND `ProductPrice` IS NULL) OR (`ProductPrice` = ?)))"
        Me._adapter.UpdateCommand.CommandType = Global.System.Data.CommandType.Tex
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
        Me._adapter.UpdateCommand.Parameters.Add(New Global.System.Data.OleDb.OleDb
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Private Sub InitConnection()
        Me._connection = New Global.System.Data.OleDb.OleDbConnection()
        Me._connection.ConnectionString = Global.BarcodeInCrystalReports.My.MySett
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
    Private Sub InitCommandCollection()
        Me._commandCollection = New Global.System.Data.OleDb.OleDbCommand(0) {}
        Me._commandCollection(0) = New Global.System.Data.OleDb.OleDbCommand()
        Me._commandCollection(0).Connection = Me.Connection
        Me._commandCollection(0).CommandText = "SELECT ID, ProductName, ProductDesc
        Me._commandCollection(0).CommandType = Global.System.Data.CommandType.Text
    End Sub

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
     Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapte
     Global.System.ComponentModel.DataObjectMethodAttribute(Global.System.Componen
    Public Overloads Overridable Function Fill(ByVal dataTable As ProductsDataSet.
        Me.Adapter.SelectCommand = Me.CommandCollection(0)
        If (Me.ClearBeforeFill = true) Then
            dataTable.Clear
        End If
        Dim returnValue As Integer = Me.Adapter.Fill(dataTable)
        Return returnValue
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
     Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapte
     Global.System.ComponentModel.DataObjectMethodAttribute(Global.System.Componen
    Public Overloads Overridable Function GetData() As ProductsDataSet.ProductsDat
        Me.Adapter.SelectCommand = Me.CommandCollection(0)
        Dim dataTable As ProductsDataSet.ProductsDataTable = New ProductsDataSet.P
        Me.Adapter.Fill(dataTable)
        Return dataTable
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
     Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapte
    Public Overloads Overridable Function Update(ByVal dataTable As ProductsDataSe
```

```
            Return Me.Adapter.Update(dataTable)
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
         Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter
        Public Overloads Overridable Function Update(ByVal dataSet As ProductsDataSet)
            Return Me.Adapter.Update(dataSet, "Products")
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
         Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter
        Public Overloads Overridable Function Update(ByVal dataRow As Global.System.Dat
            Return Me.Adapter.Update(New Global.System.Data.DataRow() {dataRow})
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
         Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter
        Public Overloads Overridable Function Update(ByVal dataRows() As Global.System
            Return Me.Adapter.Update(dataRows)
        End Function

        <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
         Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
         Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter
         Global.System.ComponentModel.DataObjectMethodAttribute(Global.System.Componen
        Public Overloads Overridable Function Delete(ByVal Original_ID As Integer, ByV
            Me.Adapter.DeleteCommand.Parameters(0).Value = CType(Original_ID,Integer)
            If (Original_ProductName Is Nothing) Then
                Me.Adapter.DeleteCommand.Parameters(1).Value = CType(1,Object)
                Me.Adapter.DeleteCommand.Parameters(2).Value = Global.System.DBNull.Va
            Else
                Me.Adapter.DeleteCommand.Parameters(1).Value = CType(0,Object)
                Me.Adapter.DeleteCommand.Parameters(2).Value = CType(Original_ProductN
            End If
            If (Original_ProductDescription Is Nothing) Then
                Me.Adapter.DeleteCommand.Parameters(3).Value = CType(1,Object)
                Me.Adapter.DeleteCommand.Parameters(4).Value = Global.System.DBNull.Va
            Else
                Me.Adapter.DeleteCommand.Parameters(3).Value = CType(0,Object)
                Me.Adapter.DeleteCommand.Parameters(4).Value = CType(Original_ProductD
            End If
            If (Original_ProductPrice Is Nothing) Then
                Me.Adapter.DeleteCommand.Parameters(5).Value = CType(1,Object)
                Me.Adapter.DeleteCommand.Parameters(6).Value = Global.System.DBNull.Va
            Else
                Me.Adapter.DeleteCommand.Parameters(5).Value = CType(0,Object)
                Me.Adapter.DeleteCommand.Parameters(6).Value = CType(Original_ProductP
            End If
            Dim previousConnectionState As Global.System.Data.ConnectionState = Me.Ada
            If ((Me.Adapter.DeleteCommand.Connection.State And Global.System.Data.Conne
                        <> Global.System.Data.ConnectionState.Open) Then
                Me.Adapter.DeleteCommand.Connection.Open
            End If
            Try
                Dim returnValue As Integer = Me.Adapter.DeleteCommand.ExecuteNonQuery
                Return returnValue
            Finally
```

```
            If (previousConnectionState = Global.System.Data.ConnectionState.Closed
                Me.Adapter.DeleteCommand.Connection.Close
            End If
        End Try
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
     Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter
     Global.System.ComponentModel.DataObjectMethodAttribute(Global.System.Component
    Public Overloads Overridable Function Insert(ByVal ProductName As String, ByVa
        If (ProductName Is Nothing) Then
            Me.Adapter.InsertCommand.Parameters(0).Value = Global.System.DBNull.Va
        Else
            Me.Adapter.InsertCommand.Parameters(0).Value = CType(ProductName,String
        End If
        If (ProductDescription Is Nothing) Then
            Me.Adapter.InsertCommand.Parameters(1).Value = Global.System.DBNull.Va
        Else
            Me.Adapter.InsertCommand.Parameters(1).Value = CType(ProductDescriptio
        End If
        If (ProductPrice Is Nothing) Then
            Me.Adapter.InsertCommand.Parameters(2).Value = Global.System.DBNull.Va
        Else
            Me.Adapter.InsertCommand.Parameters(2).Value = CType(ProductPrice,Stri
        End If
        Dim previousConnectionState As Global.System.Data.ConnectionState = Me.Ada
        If ((Me.Adapter.InsertCommand.Connection.State And Global.System.Data.Conne
                    <> Global.System.Data.ConnectionState.Open) Then
            Me.Adapter.InsertCommand.Connection.Open
        End If
        Try
            Dim returnValue As Integer = Me.Adapter.InsertCommand.ExecuteNonQuery
            Return returnValue
        Finally
            If (previousConnectionState = Global.System.Data.ConnectionState.Closed
                Me.Adapter.InsertCommand.Connection.Close
            End If
        End Try
    End Function

    <Global.System.Diagnostics.DebuggerNonUserCodeAttribute(),  _
     Global.System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Type
     Global.System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter
     Global.System.ComponentModel.DataObjectMethodAttribute(Global.System.Component
    Public Overloads Overridable Function Update(ByVal ProductName As String, ByVa
        If (ProductName Is Nothing) Then
            Me.Adapter.UpdateCommand.Parameters(0).Value = Global.System.DBNull.Va
        Else
            Me.Adapter.UpdateCommand.Parameters(0).Value = CType(ProductName,String
        End If
        If (ProductDescription Is Nothing) Then
            Me.Adapter.UpdateCommand.Parameters(1).Value = Global.System.DBNull.Va
        Else
            Me.Adapter.UpdateCommand.Parameters(1).Value = CType(ProductDescriptio
        End If
        If (ProductPrice Is Nothing) Then
            Me.Adapter.UpdateCommand.Parameters(2).Value = Global.System.DBNull.Va
        Else
            Me.Adapter.UpdateCommand.Parameters(2).Value = CType(ProductPrice,Stri
```

```
            End If
            Me.Adapter.UpdateCommand.Parameters(3).Value = CType(Original_ID,Integer)
            If (Original_ProductName Is Nothing) Then
                Me.Adapter.UpdateCommand.Parameters(4).Value = CType(1,Object)
                Me.Adapter.UpdateCommand.Parameters(5).Value = Global.System.DBNull.Va
            Else
                Me.Adapter.UpdateCommand.Parameters(4).Value = CType(0,Object)
                Me.Adapter.UpdateCommand.Parameters(5).Value = CType(Original_ProductN
            End If
            If (Original_ProductDescription Is Nothing) Then
                Me.Adapter.UpdateCommand.Parameters(6).Value = CType(1,Object)
                Me.Adapter.UpdateCommand.Parameters(7).Value = Global.System.DBNull.Va
            Else
                Me.Adapter.UpdateCommand.Parameters(6).Value = CType(0,Object)
                Me.Adapter.UpdateCommand.Parameters(7).Value = CType(Original_ProductD
            End If
            If (Original_ProductPrice Is Nothing) Then
                Me.Adapter.UpdateCommand.Parameters(8).Value = CType(1,Object)
                Me.Adapter.UpdateCommand.Parameters(9).Value = Global.System.DBNull.Va
            Else
                Me.Adapter.UpdateCommand.Parameters(8).Value = CType(0,Object)
                Me.Adapter.UpdateCommand.Parameters(9).Value = CType(Original_ProductP
            End If
            Dim previousConnectionState As Global.System.Data.ConnectionState = Me.Ada
            If ((Me.Adapter.UpdateCommand.Connection.State And Global.System.Data.Conn
                        <> Global.System.Data.ConnectionState.Open) Then
                Me.Adapter.UpdateCommand.Connection.Open
            End If
            Try
                Dim returnValue As Integer = Me.Adapter.UpdateCommand.ExecuteNonQuery
                Return returnValue
            Finally
                If (previousConnectionState = Global.System.Data.ConnectionState.Close
                    Me.Adapter.UpdateCommand.Connection.Close
                End If
            End Try
        End Function
    End Class
End Namespace
```

ProductsDataSet.xsc

```
<?xml version="1.0" encoding="utf-8"?>
<!--<autogenerated>
     This code was generated by a tool.
     Changes to this file may cause incorrect behavior and will be lost if
     the code is regenerated.
</autogenerated>-->
<DataSetUISetting Version="1.00" xmlns="urn:schemas-microsoft-com:xml-msdatasource">
  <TableUISettings />
```

```
</DataSetUISetting>
```

ProductsDataSet.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="ProductsDataSet" targetNamespace="http://tempuri.org/ProductsDataSet.xs
  <xs:annotation>
    <xs:appinfo source="urn:schemas-microsoft-com:xml-msdatasource">
      <DataSource DefaultConnectionIndex="0" FunctionsComponentName="QueriesTableAdapte
        <Connections>
          <Connection AppSettingsObjectName="MySettings" AppSettingsPropertyName="produ
        </Connections>
        <Tables>
          <TableAdapter BaseClass="System.ComponentModel.Component" DataAccessorModifie
            <MainSource>
              <DbSource ConnectionRef="productsConnectionString (MySettings)" DbObject
                <DeleteCommand>
                  <DbCommand CommandType="Text" ModifiedByUser="false">
                    <CommandText>DELETE FROM `Products` WHERE ((`ID` = ?) AND ((? = 1 A
                    <Parameters>
                      <Parameter AllowDbNull="false" AutogeneratedName="" DataSourceNa
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                    </Parameters>
                  </DbCommand>
                </DeleteCommand>
                <InsertCommand>
                  <DbCommand CommandType="Text" ModifiedByUser="false">
                    <CommandText>INSERT INTO `Products` (`ProductName`, `ProductDescri
                    <Parameters>
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
                    </Parameters>
                  </DbCommand>
                </InsertCommand>
                <SelectCommand>
                  <DbCommand CommandType="Text" ModifiedByUser="false">
                    <CommandText>SELECT ID, ProductName, ProductDescription, ProductPr
                    <Parameters />
                  </DbCommand>
                </SelectCommand>
                <UpdateCommand>
                  <DbCommand CommandType="Text" ModifiedByUser="false">
                    <CommandText>UPDATE `Products` SET `ProductName` = ?, `ProductDesc
                    <Parameters>
                      <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceNam
```

```
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="false" AutogeneratedName="" DataSourceNa
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                    <Parameter AllowDbNull="true" AutogeneratedName="" DataSourceName
                  </Parameters>
                </DbCommand>
              </UpdateCommand>
            </DbSource>
          </MainSource>
          <Mappings>
            <Mapping SourceColumn="ID" DataSetColumn="ID" />
            <Mapping SourceColumn="ProductName" DataSetColumn="ProductName" />
            <Mapping SourceColumn="ProductDescription" DataSetColumn="ProductDescript
            <Mapping SourceColumn="ProductPrice" DataSetColumn="ProductPrice" />
          </Mappings>
          <Sources />
        </TableAdapter>
      </Tables>
      <Sources />
    </DataSource>
  </xs:appinfo>
</xs:annotation>
<xs:element name="ProductsDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="Products" msprop:Generator_TableClassName="ProductsDataTable
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ID" msdata:AutoIncrement="true" msdata:AutoIncrementSe
            <xs:element name="ProductName" msprop:Generator_ColumnVarNameInTable="co
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="255" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="ProductDescription" msprop:Generator_ColumnVarNameInTa
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="255" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="ProductPrice" msprop:Generator_ColumnVarNameInTable="c
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="255" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="BarcodeImage" msprop:Generator_ColumnVarNameInTable="c
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
```

```
    </xs:complexType>
    <xs:unique name="Constraint1" msdata:PrimaryKey="true">
      <xs:selector xpath=".//mstns:Products" />
      <xs:field xpath="mstns:ID" />
    </xs:unique>
  </xs:element>
</xs:schema>
```

ProductsDataSet.xss

```
<?xml version="1.0" encoding="utf-8"?>
<!--<autogenerated>
     This code was generated by a tool to store the dataset designer's layout informat
     Changes to this file may cause incorrect behavior and will be lost if
     the code is regenerated.
</autogenerated>-->
<DiagramLayout xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http:/
  <Shapes>
    <Shape ID="DesignTable:Products" ZOrder="1" X="70" Y="70" Height="172" Width="196"
  </Shapes>
  <Connectors />
</DiagramLayout>
```

app.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
    </configSections>
    <connectionStrings>
        <add name="BarcodeInCrystalReports.My.MySettings.productsConnectionString"
            connectionString="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=|DataDirecto
            providerName="System.Data.OleDb" />
    </connectionStrings>
    <system.diagnostics>
        <sources>
            <!-- This section defines the logging configuration for My.Application.Log
            <source name="DefaultSource" switchName="DefaultSwitch">
                <listeners>
                    <add name="FileLog"/>
                    <!-- Uncomment the below section to write to the Application Event
                    <!--<add name="EventLog"/>-->
```

```
                </listeners>
            </source>
        </sources>
        <switches>
            <add name="DefaultSwitch" value="Information" />
        </switches>
        <sharedListeners>
            <add name="FileLog"
                 type="Microsoft.VisualBasic.Logging.FileLogTraceListener, Microsoft.V
                 initializeData="FileLogWriter"/>
            <!-- Uncomment the below section and replace APPLICATION_NAME with the nam
            <!--<add name="EventLog" type="System.Diagnostics.EventLogTraceListener" i
        </sharedListeners>
    </system.diagnostics>
    <startup useLegacyV2RuntimeActivationPolicy="true">
      <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
    </startup>
</configuration>
```

VIDEO

https://www.youtube.com/watch?v=NEwNs2b9YN8

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Premium Suite Home Page
Explore ByteScout Premium Suite Documentation
Explore Samples
Sign Up for ByteScout Premium Suite Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co