How to generate report rom sql server c# 2015 with barcode sdk in Crystal Reports with ByteScout Premium Suite

How to write a robust code in Crystal Reports to generate report rom sql server c# 2015 with barcode sdk with this step-by-step tutorial

The sample source code below will teach you how to generate report rom sql server c# 2015 with barcode sdk in Crystal Reports. 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 report rom sql server c# 2015 with barcode sdk with Crystal Reports.

This prolific sample source code in Crystal Reports for ByteScout Premium Suite contains various functions and other necessary options you should do calling the API to generate report rom sql server c# 2015 with barcode sdk. Simply copy and paste in your Crystal Reports project or application you and then run your app! Further improvement of the code will make it more robust.

Trial version of ByteScout Premium Suite is available for free. Source code samples are included to help you with your Crystal Reports app.

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.cs

```csharp
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated by a tool.
//     Runtime Version:4.0.30319.42000
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace ReportFromSqlServer {
    using System;
    using System.ComponentModel;
    using CrystalDecisions.Shared;
    using CrystalDecisions.ReportSource;
    using CrystalDecisions.CrystalReports.Engine;


    public class CrystalReport1 : ReportClass {

        public CrystalReport1() {
        }

        public override string ResourceName {
            get {
                return "CrystalReport1.rpt";
            }
            set {
                // Do nothing
            }
        }

        public override bool NewGenerator {
            get {
                return true;
            }
            set {
                // Do nothing
            }
        }

        public override string FullResourceName {
            get {
                return "ReportFromSqlServer.CrystalReport1.rpt";
            }
            set {
                // Do nothing
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public CrystalDecisions.CrystalReports.Engine.Section Section1 {
            get {
                return this.ReportDefinition.Sections[0];
```

```csharp
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public CrystalDecisions.CrystalReports.Engine.Section Section2 {
            get {
                return this.ReportDefinition.Sections[1];
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public CrystalDecisions.CrystalReports.Engine.Section Section3 {
            get {
                return this.ReportDefinition.Sections[2];
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public CrystalDecisions.CrystalReports.Engine.Section Section4 {
            get {
                return this.ReportDefinition.Sections[3];
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public CrystalDecisions.CrystalReports.Engine.Section Section5 {
            get {
                return this.ReportDefinition.Sections[4];
            }
        }
    }

    [System.Drawing.ToolboxBitmapAttribute(typeof(CrystalDecisions.Shared.ExportOptions
    public class CachedCrystalReport1 : Component, ICachedReport {

        public CachedCrystalReport1() {
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public virtual bool IsCacheable {
            get {
                return true;
            }
            set {
                //
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public virtual bool ShareDBLogonInfo {
            get {
                return false;
            }
            set {
```

```csharp
                //
            }
        }

        [Browsable(false)]
        [DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerial
        public virtual System.TimeSpan CacheTimeOut {
            get {
                return CachedReportConstants.DEFAULT_TIMEOUT;
            }
            set {
                //
            }
        }

        public virtual CrystalDecisions.CrystalReports.Engine.ReportDocument CreateRepo
            CrystalReport1 rpt = new CrystalReport1();
            rpt.Site = this.Site;
            return rpt;
        }

        public virtual string GetCustomizedCacheKey(RequestContext request) {
            String key = null;
            // // 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;
        }
    }
}
```

DataSet1.Designer.cs

```csharp
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated by a tool.
//     Runtime Version:4.0.30319.42000
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------
```

```csharp
#pragma warning disable 1591

namespace ReportFromSqlServer {


    /// <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("DataSet1")]
    [global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.DataSet")]
    public partial class DataSet1 : global::System.Data.DataSet {

        private ProductsDataTable tableProducts;

        private global::System.Data.SchemaSerializationMode _schemaSerializationMode =

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        public DataSet1() {
            this.BeginInit();
            this.InitClass();
            global::System.ComponentModel.CollectionChangeEventHandler schemaChangedHan
            base.Tables.CollectionChanged += schemaChangedHandler;
            base.Relations.CollectionChanged += schemaChangedHandler;
            this.EndInit();
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        protected DataSet1(global::System.Runtime.Serialization.SerializationInfo info
                base(info, context, false) {
            if ((this.IsBinarySerialized(info, context) == true)) {
                this.InitVars(false);
                global::System.ComponentModel.CollectionChangeEventHandler schemaChange
                this.Tables.CollectionChanged += schemaChangedHandler1;
                this.Relations.CollectionChanged += schemaChangedHandler1;
                return;
            }
            string strSchema = ((string)(info.GetValue("XmlSchema", typeof(string))));
            if ((this.DetermineSchemaSerializationMode(info, context) == global::System
                global::System.Data.DataSet ds = new global::System.Data.DataSet();
                ds.ReadXmlSchema(new global::System.Xml.XmlTextReader(new global::Syste
                if ((ds.Tables["Products"] != null)) {
                    base.Tables.Add(new ProductsDataTable(ds.Tables["Products"]));
                }
                this.DataSetName = ds.DataSetName;
                this.Prefix = ds.Prefix;
                this.Namespace = ds.Namespace;
                this.Locale = ds.Locale;
                this.CaseSensitive = ds.CaseSensitive;
                this.EnforceConstraints = ds.EnforceConstraints;
                this.Merge(ds, false, global::System.Data.MissingSchemaAction.Add);
                this.InitVars();
            }
            else {
```

```csharp
            this.ReadXmlSchema(new global::System.Xml.XmlTextReader(new global::Sys
        }
    this.GetSerializationData(info, context);
    global::System.ComponentModel.CollectionChangeEventHandler schemaChangedHar
    base.Tables.CollectionChanged += schemaChangedHandler;
    this.Relations.CollectionChanged += schemaChangedHandler;
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Browsable(false)]
[global::System.ComponentModel.DesignerSerializationVisibility(global::System.C
public ProductsDataTable Products {
    get {
        return this.tableProducts;
    }
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.BrowsableAttribute(true)]
[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global
public override global::System.Data.SchemaSerializationMode SchemaSerialization
    get {
        return this._schemaSerializationMode;
    }
    set {
        this._schemaSerializationMode = value;
    }
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global
public new global::System.Data.DataTableCollection Tables {
    get {
        return base.Tables;
    }
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.DesignerSerializationVisibilityAttribute(global
public new global::System.Data.DataRelationCollection Relations {
    get {
        return base.Relations;
    }
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
protected override void InitializeDerivedDataSet() {
    this.BeginInit();
    this.InitClass();
    this.EndInit();
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
public override global::System.Data.DataSet Clone() {
```

```csharp
            DataSet1 cln = ((DataSet1)(base.Clone()));
            cln.InitVars();
            cln.SchemaSerializationMode = this.SchemaSerializationMode;
            return cln;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        protected override bool ShouldSerializeTables() {
            return false;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        protected override bool ShouldSerializeRelations() {
            return false;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        protected override void ReadXmlSerializable(global::System.Xml.XmlReader reader
            if ((this.DetermineSchemaSerializationMode(reader) == global::System.Data.S
                this.Reset();
                global::System.Data.DataSet ds = new global::System.Data.DataSet();
                ds.ReadXml(reader);
                if ((ds.Tables["Products"] != null)) {
                    base.Tables.Add(new ProductsDataTable(ds.Tables["Products"]));
                }
                this.DataSetName = ds.DataSetName;
                this.Prefix = ds.Prefix;
                this.Namespace = ds.Namespace;
                this.Locale = ds.Locale;
                this.CaseSensitive = ds.CaseSensitive;
                this.EnforceConstraints = ds.EnforceConstraints;
                this.Merge(ds, false, global::System.Data.MissingSchemaAction.Add);
                this.InitVars();
            }
            else {
                this.ReadXml(reader);
                this.InitVars();
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        protected override global::System.Xml.Schema.XmlSchema GetSchemaSerializable()
            global::System.IO.MemoryStream stream = new global::System.IO.MemoryStream
            this.WriteXmlSchema(new global::System.Xml.XmlTextWriter(stream, null));
            stream.Position = 0;
            return global::System.Xml.Schema.XmlSchema.Read(new global::System.Xml.Xml
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        internal void InitVars() {
            this.InitVars(true);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
```

```csharp
        internal void InitVars(bool initTable) {
            this.tableProducts = ((ProductsDataTable)(base.Tables["Products"]));
            if ((initTable == true)) {
                if ((this.tableProducts != null)) {
                    this.tableProducts.InitVars();
                }
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        private void InitClass() {
            this.DataSetName = "DataSet1";
            this.Prefix = "";
            this.Namespace = "http://tempuri.org/DataSet1.xsd";
            this.EnforceConstraints = true;
            this.SchemaSerializationMode = global::System.Data.SchemaSerializationMode
            this.tableProducts = new ProductsDataTable();
            base.Tables.Add(this.tableProducts);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        private bool ShouldSerializeProducts() {
            return false;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        private void SchemaChanged(object sender, global::System.ComponentModel.Collec
            if ((e.Action == global::System.ComponentModel.CollectionChangeAction.Remov
                this.InitVars();
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        public static global::System.Xml.Schema.XmlSchemaComplexType GetTypedDataSetSch
            DataSet1 ds = new DataSet1();
            global::System.Xml.Schema.XmlSchemaComplexType type = new global::System.Xm
            global::System.Xml.Schema.XmlSchemaSequence sequence = new global::System.X
            global::System.Xml.Schema.XmlSchemaAny any = new global::System.Xml.Schema
            any.Namespace = ds.Namespace;
            sequence.Items.Add(any);
            type.Particle = sequence;
            global::System.Xml.Schema.XmlSchema dsSchema = ds.GetSchemaSerializable();
            if (xs.Contains(dsSchema.TargetNamespace)) {
                global::System.IO.MemoryStream s1 = new global::System.IO.MemoryStream
                global::System.IO.MemoryStream s2 = new global::System.IO.MemoryStream
                try {
                    global::System.Xml.Schema.XmlSchema schema = null;
                    dsSchema.Write(s1);
                    for (global::System.Collections.IEnumerator schemas = xs.Schemas(ds
                        schema = ((global::System.Xml.Schema.XmlSchema)(schemas.Current
                        s2.SetLength(0);
                        schema.Write(s2);
                        if ((s1.Length == s2.Length)) {
                            s1.Position = 0;
                            s2.Position = 0;
                            for (; ((s1.Position != s1.Length)
```

```csharp
                               && (s1.ReadByte() == s2.ReadByte())); ) {
                            ;
                        }
                        if ((s1.Position == s1.Length)) {
                            return type;
                        }
                    }
                }
            }
            finally {
                if ((s1 != null)) {
                    s1.Close();
                }
                if ((s2 != null)) {
                    s2.Close();
                }
            }
        }
        xs.Add(dsSchema);
        return type;
    }

    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
    public delegate void ProductsRowChangeEventHandler(object sender, ProductsRowCh

    /// <summary>
    ///Represents the strongly named DataTable class.
    ///</summary>
    [global::System.Serializable()]
    [global::System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSch
    public partial class ProductsDataTable : global::System.Data.DataTable, global

        private global::System.Data.DataColumn columnProduct_ID;

        private global::System.Data.DataColumn columnProduct_Name;

        private global::System.Data.DataColumn columnProduct_Description;

        private global::System.Data.DataColumn columnBarCode;

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public ProductsDataTable() {
            this.TableName = "Products";
            this.BeginInit();
            this.InitClass();
            this.EndInit();
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        internal ProductsDataTable(global::System.Data.DataTable table) {
            this.TableName = table.TableName;
            if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {
                this.CaseSensitive = table.CaseSensitive;
            }
            if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {
                this.Locale = table.Locale;
            }
            if ((table.Namespace != table.DataSet.Namespace)) {
```

```csharp
            this.Namespace = table.Namespace;
        }
        this.Prefix = table.Prefix;
        this.MinimumCapacity = table.MinimumCapacity;
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    protected ProductsDataTable(global::System.Runtime.Serialization.Serializat
            base(info, context) {
        this.InitVars();
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public global::System.Data.DataColumn Product_IDColumn {
        get {
            return this.columnProduct_ID;
        }
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public global::System.Data.DataColumn Product_NameColumn {
        get {
            return this.columnProduct_Name;
        }
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public global::System.Data.DataColumn Product_DescriptionColumn {
        get {
            return this.columnProduct_Description;
        }
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public global::System.Data.DataColumn BarCodeColumn {
        get {
            return this.columnBarCode;
        }
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    [global::System.ComponentModel.Browsable(false)]
    public int Count {
        get {
            return this.Rows.Count;
        }
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public ProductsRow this[int index] {
        get {
            return ((ProductsRow)(this.Rows[index]));
        }
```

```csharp
        }

        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public event ProductsRowChangeEventHandler ProductsRowChanging;

        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public event ProductsRowChangeEventHandler ProductsRowChanged;

        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public event ProductsRowChangeEventHandler ProductsRowDeleting;

        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public event ProductsRowChangeEventHandler ProductsRowDeleted;

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public void AddProductsRow(ProductsRow row) {
            this.Rows.Add(row);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public ProductsRow AddProductsRow(int Product_ID, string Product_Name, str
            ProductsRow rowProductsRow = ((ProductsRow)(this.NewRow()));
            object[] columnValuesArray = new object[] {
                    Product_ID,
                    Product_Name,
                    Product_Description,
                    BarCode};
            rowProductsRow.ItemArray = columnValuesArray;
            this.Rows.Add(rowProductsRow);
            return rowProductsRow;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public virtual global::System.Collections.IEnumerator GetEnumerator() {
            return this.Rows.GetEnumerator();
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public override global::System.Data.DataTable Clone() {
            ProductsDataTable cln = ((ProductsDataTable)(base.Clone()));
            cln.InitVars();
            return cln;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        protected override global::System.Data.DataTable CreateInstance() {
            return new ProductsDataTable();
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        internal void InitVars() {
            this.columnProduct_ID = base.Columns["Product ID"];
            this.columnProduct_Name = base.Columns["Product Name"];
            this.columnProduct_Description = base.Columns["Product Description"];
```

```csharp
            this.columnBarCode = base.Columns["BarCode"];
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        private void InitClass() {
            this.columnProduct_ID = new global::System.Data.DataColumn("Product ID"
            base.Columns.Add(this.columnProduct_ID);
            this.columnProduct_Name = new global::System.Data.DataColumn("Product N
            base.Columns.Add(this.columnProduct_Name);
            this.columnProduct_Description = new global::System.Data.DataColumn("Pr
            base.Columns.Add(this.columnProduct_Description);
            this.columnBarCode = new global::System.Data.DataColumn("BarCode", type
            base.Columns.Add(this.columnBarCode);
            this.columnProduct_Name.MaxLength = 100;
            this.columnProduct_Description.MaxLength = 255;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public ProductsRow NewProductsRow() {
            return ((ProductsRow)(this.NewRow()));
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        protected override global::System.Data.DataRow NewRowFromBuilder(global::Sy
            return new ProductsRow(builder);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        protected override global::System.Type GetRowType() {
            return typeof(ProductsRow);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        protected override void OnRowChanged(global::System.Data.DataRowChangeEvent
            base.OnRowChanged(e);
            if ((this.ProductsRowChanged != null)) {
                this.ProductsRowChanged(this, new ProductsRowChangeEvent(((Products
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        protected override void OnRowChanging(global::System.Data.DataRowChangeEver
            base.OnRowChanging(e);
            if ((this.ProductsRowChanging != null)) {
                this.ProductsRowChanging(this, new ProductsRowChangeEvent(((Produc
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        protected override void OnRowDeleted(global::System.Data.DataRowChangeEvent
            base.OnRowDeleted(e);
            if ((this.ProductsRowDeleted != null)) {
                this.ProductsRowDeleted(this, new ProductsRowChangeEvent(((Products
```

```csharp
                }
            }

            [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
            [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
            protected override void OnRowDeleting(global::System.Data.DataRowChangeEver
                base.OnRowDeleting(e);
                if ((this.ProductsRowDeleting != null)) {
                    this.ProductsRowDeleting(this, new ProductsRowChangeEvent(((Product
                }
            }

            [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
            [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
            public void RemoveProductsRow(ProductsRow row) {
                this.Rows.Remove(row);
            }

            [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
            [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
            public static global::System.Xml.Schema.XmlSchemaComplexType GetTypedTableS
                global::System.Xml.Schema.XmlSchemaComplexType type = new global::Syste
                global::System.Xml.Schema.XmlSchemaSequence sequence = new global::Syst
                DataSet1 ds = new DataSet1();
                global::System.Xml.Schema.XmlSchemaAny any1 = new global::System.Xml.Sc
                any1.Namespace = "http://www.w3.org/2001/XMLSchema";
                any1.MinOccurs = new decimal(0);
                any1.MaxOccurs = decimal.MaxValue;
                any1.ProcessContents = global::System.Xml.Schema.XmlSchemaContentProces
                sequence.Items.Add(any1);
                global::System.Xml.Schema.XmlSchemaAny any2 = new global::System.Xml.Sc
                any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";
                any2.MinOccurs = new decimal(1);
                any2.ProcessContents = global::System.Xml.Schema.XmlSchemaContentProces
                sequence.Items.Add(any2);
                global::System.Xml.Schema.XmlSchemaAttribute attribute1 = new global::S
                attribute1.Name = "namespace";
                attribute1.FixedValue = ds.Namespace;
                type.Attributes.Add(attribute1);
                global::System.Xml.Schema.XmlSchemaAttribute attribute2 = new global::S
                attribute2.Name = "tableTypeName";
                attribute2.FixedValue = "ProductsDataTable";
                type.Attributes.Add(attribute2);
                type.Particle = sequence;
                global::System.Xml.Schema.XmlSchema dsSchema = ds.GetSchemaSerializable
                if (xs.Contains(dsSchema.TargetNamespace)) {
                    global::System.IO.MemoryStream s1 = new global::System.IO.MemoryStr
                    global::System.IO.MemoryStream s2 = new global::System.IO.MemoryStr
                    try {
                        global::System.Xml.Schema.XmlSchema schema = null;
                        dsSchema.Write(s1);
                        for (global::System.Collections.IEnumerator schemas = xs.Schema
                            schema = ((global::System.Xml.Schema.XmlSchema)(schemas.Cur
                            s2.SetLength(0);
                            schema.Write(s2);
                            if ((s1.Length == s2.Length)) {
                                s1.Position = 0;
                                s2.Position = 0;
                                for (; ((s1.Position != s1.Length)
                                            && (s1.ReadByte() == s2.ReadByte())); ) {
```

```csharp
                        ;
                    }
                    if ((s1.Position == s1.Length)) {
                        return type;
                    }
                }
            }
        }
        finally {
            if ((s1 != null)) {
                s1.Close();
            }
            if ((s2 != null)) {
                s2.Close();
            }
        }
    }
    xs.Add(dsSchema);
    return type;
    }
}


/// <summary>
///Represents strongly named DataRow class.
///</summary>
public partial class ProductsRow : global::System.Data.DataRow {

    private ProductsDataTable tableProducts;

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    internal ProductsRow(global::System.Data.DataRowBuilder rb) :
            base(rb) {
        this.tableProducts = ((ProductsDataTable)(this.Table));
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public int Product_ID {
        get {
            try {
                return ((int)(this[this.tableProducts.Product_IDColumn]));
            }
            catch (global::System.InvalidCastException e) {
                throw new global::System.Data.StrongTypingException("The value
            }
        }
        set {
            this[this.tableProducts.Product_IDColumn] = value;
        }
    }

    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
    public string Product_Name {
        get {
            try {
                return ((string)(this[this.tableProducts.Product_NameColumn]))
            }
            catch (global::System.InvalidCastException e) {
```

```csharp
                    throw new global::System.Data.StrongTypingException("The value
                }
            }
            set {
                this[this.tableProducts.Product_NameColumn] = value;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public string Product_Description {
            get {
                try {
                    return ((string)(this[this.tableProducts.Product_DescriptionCo
                }
                catch (global::System.InvalidCastException e) {
                    throw new global::System.Data.StrongTypingException("The value
                }
            }
            set {
                this[this.tableProducts.Product_DescriptionColumn] = value;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public byte[] BarCode {
            get {
                try {
                    return ((byte[])(this[this.tableProducts.BarCodeColumn]));
                }
                catch (global::System.InvalidCastException e) {
                    throw new global::System.Data.StrongTypingException("The value
                }
            }
            set {
                this[this.tableProducts.BarCodeColumn] = value;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public bool IsProduct_IDNull() {
            return this.IsNull(this.tableProducts.Product_IDColumn);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public void SetProduct_IDNull() {
            this[this.tableProducts.Product_IDColumn] = global::System.Convert.DBNu
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public bool IsProduct_NameNull() {
            return this.IsNull(this.tableProducts.Product_NameColumn);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
```

```csharp
        public void SetProduct_NameNull() {
            this[this.tableProducts.Product_NameColumn] = global::System.Convert.DB
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public bool IsProduct_DescriptionNull() {
            return this.IsNull(this.tableProducts.Product_DescriptionColumn);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public void SetProduct_DescriptionNull() {
            this[this.tableProducts.Product_DescriptionColumn] = global::System.Con
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public bool IsBarCodeNull() {
            return this.IsNull(this.tableProducts.BarCodeColumn);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public void SetBarCodeNull() {
            this[this.tableProducts.BarCodeColumn] = global::System.Convert.DBNull
        }
    }

    /// <summary>
    ///Row event argument class
    ///</summary>
    [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
    public class ProductsRowChangeEvent : global::System.EventArgs {

        private ProductsRow eventRow;

        private global::System.Data.DataRowAction eventAction;

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public ProductsRowChangeEvent(ProductsRow row, global::System.Data.DataRowA
            this.eventRow = row;
            this.eventAction = action;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public ProductsRow Row {
            get {
                return this.eventRow;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design
        public global::System.Data.DataRowAction Action {
            get {
                return this.eventAction;
            }
```

```csharp
            }
        }
    }
}
namespace ReportFromSqlServer.DataSet1TableAdapters {


    /// <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.[
        ", Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]
    [global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]
    public partial class ProductsTableAdapter : global::System.ComponentModel.Component

        private global::System.Data.SqlClient.SqlDataAdapter _adapter;

        private global::System.Data.SqlClient.SqlConnection _connection;

        private global::System.Data.SqlClient.SqlCommand[] _commandCollection;

        private bool _clearBeforeFill;

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        public ProductsTableAdapter() {
            this.ClearBeforeFill = true;
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        private global::System.Data.SqlClient.SqlDataAdapter Adapter {
            get {
                if ((this._adapter == null)) {
                    this.InitAdapter();
                }
                return this._adapter;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        internal global::System.Data.SqlClient.SqlConnection Connection {
            get {
                if ((this._connection == null)) {
                    this.InitConnection();
                }
                return this._connection;
            }
            set {
                this._connection = value;
                if ((this.Adapter.InsertCommand != null)) {
                    this.Adapter.InsertCommand.Connection = value;
                }
                if ((this.Adapter.DeleteCommand != null)) {
                    this.Adapter.DeleteCommand.Connection = value;
                }
```

```csharp
                if ((this.Adapter.UpdateCommand != null)) {
                    this.Adapter.UpdateCommand.Connection = value;
                }
                for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {
                    if ((this.CommandCollection[i] != null)) {
                        ((global::System.Data.SqlClient.SqlCommand)(this.CommandCollect
                    }
                }
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        protected global::System.Data.SqlClient.SqlCommand[] CommandCollection {
            get {
                if ((this._commandCollection == null)) {
                    this.InitCommandCollection();
                }
                return this._commandCollection;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        public bool ClearBeforeFill {
            get {
                return this._clearBeforeFill;
            }
            set {
                this._clearBeforeFill = value;
            }
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        private void InitAdapter() {
            this._adapter = new global::System.Data.SqlClient.SqlDataAdapter();
            global::System.Data.Common.DataTableMapping tableMapping = new global::Syst
            tableMapping.SourceTable = "Table";
            tableMapping.DataSetTable = "Products";
            tableMapping.ColumnMappings.Add("Product ID", "Product ID");
            tableMapping.ColumnMappings.Add("Product Name", "Product Name");
            tableMapping.ColumnMappings.Add("Product Description", "Product Description
            this._adapter.TableMappings.Add(tableMapping);
            this._adapter.InsertCommand = new global::System.Data.SqlClient.SqlCommand(
            this._adapter.InsertCommand.Connection = this.Connection;
            this._adapter.InsertCommand.CommandText = "INSERT INTO [dbo].[Products] ([F
                ") VALUES (@Product_ID, @Product_Name, @Product_Description)";
            this._adapter.InsertCommand.CommandType = global::System.Data.CommandType.T
            this._adapter.InsertCommand.Parameters.Add(new global::System.Data.SqlClier
            this._adapter.InsertCommand.Parameters.Add(new global::System.Data.SqlClier
            this._adapter.InsertCommand.Parameters.Add(new global::System.Data.SqlClier
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        private void InitConnection() {
            this._connection = new global::System.Data.SqlClient.SqlConnection();
            this._connection.ConnectionString = global::ReportFromSqlServer.Properties
        }
```

```csharp
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
private void InitCommandCollection() {
    this._commandCollection = new global::System.Data.SqlClient.SqlCommand[1];
    this._commandCollection[0] = new global::System.Data.SqlClient.SqlCommand()
    this._commandCollection[0].Connection = this.Connection;
    this._commandCollection[0].CommandText = "SELECT [Product ID], [Product Nam
    this._commandCollection[0].CommandType = global::System.Data.CommandType.Te
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compone
public virtual int Fill(DataSet1.ProductsDataTable dataTable) {
    this.Adapter.SelectCommand = this.CommandCollection[0];
    if ((this.ClearBeforeFill == true)) {
        dataTable.Clear();
    }
    int returnValue = this.Adapter.Fill(dataTable);
    return returnValue;
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compone
public virtual DataSet1.ProductsDataTable GetData() {
    this.Adapter.SelectCommand = this.CommandCollection[0];
    DataSet1.ProductsDataTable dataTable = new DataSet1.ProductsDataTable();
    this.Adapter.Fill(dataTable);
    return dataTable;
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
public virtual int Update(DataSet1.ProductsDataTable dataTable) {
    return this.Adapter.Update(dataTable);
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
public virtual int Update(DataSet1 dataSet) {
    return this.Adapter.Update(dataSet, "Products");
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
public virtual int Update(global::System.Data.DataRow dataRow) {
    return this.Adapter.Update(new global::System.Data.DataRow[] {
                dataRow});
}

[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
```

```csharp
        public virtual int Update(global::System.Data.DataRow[] dataRows) {
            return this.Adapter.Update(dataRows);
        }

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.Typ
        [global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapte
        [global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compone
        public virtual int Insert(global::System.Nullable<int> Product_ID, string Produ
            if ((Product_ID.HasValue == true)) {
                this.Adapter.InsertCommand.Parameters[0].Value = ((int)(Product_ID.Valu
            }
            else {
                this.Adapter.InsertCommand.Parameters[0].Value = global::System.DBNull
            }
            if ((Product_Name == null)) {
                this.Adapter.InsertCommand.Parameters[1].Value = global::System.DBNull
            }
            else {
                this.Adapter.InsertCommand.Parameters[1].Value = ((string)(Product_Name
            }
            if ((Product_Description == null)) {
                this.Adapter.InsertCommand.Parameters[2].Value = global::System.DBNull
            }
            else {
                this.Adapter.InsertCommand.Parameters[2].Value = ((string)(Product_Des
            }
            global::System.Data.ConnectionState previousConnectionState = this.Adapter
            if (((this.Adapter.InsertCommand.Connection.State & global::System.Data.Cor
                        != global::System.Data.ConnectionState.Open)) {
                this.Adapter.InsertCommand.Connection.Open();
            }
            try {
                int returnValue = this.Adapter.InsertCommand.ExecuteNonQuery();
                return returnValue;
            }
            finally {
                if ((previousConnectionState == global::System.Data.ConnectionState.Clo
                    this.Adapter.InsertCommand.Connection.Close();
                }
            }
        }
    }
}

#pragma warning restore 1591
```

DataSet1.xsc

```xml
<?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>
  </TableUISettings>
</DataSetUISetting>
```

DataSet1.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="DataSet1" targetNamespace="http://tempuri.org/DataSet1.xsd" xmlns:mstns=
  <xs:annotation>
    <xs:appinfo source="urn:schemas-microsoft-com:xml-msdatasource">
      <DataSource DefaultConnectionIndex="0" FunctionsComponentName="QueriesTableAdapte
        <Connections>
          <Connection AppSettingsObjectName="Settings" AppSettingsPropertyName="example
          </Connection>
        </Connections>
        <Tables>
          <TableAdapter BaseClass="System.ComponentModel.Component" DataAccessorModifie
            <MainSource>
              <DbSource ConnectionRef="example_dbConnectionString (Settings)" DbObjectN
                <InsertCommand>
                  <DbCommand CommandType="Text" ModifiedByUser="False">
                    <CommandText>INSERT INTO [dbo].[Products] ([Product ID], [Product N
                    <Parameters>
                      <Parameter AllowDbNull="True" AutogeneratedName="" DataSourceName
                      </Parameter>
                      <Parameter AllowDbNull="True" AutogeneratedName="" DataSourceName
                      </Parameter>
                      <Parameter AllowDbNull="True" AutogeneratedName="" DataSourceName
                      </Parameter>
                    </Parameters>
                  </DbCommand>
                </InsertCommand>
                <SelectCommand>
                  <DbCommand CommandType="Text" ModifiedByUser="False">
                    <CommandText>SELECT [Product ID], [Product Name], [Product Descript
                    <Parameters>
                    </Parameters>
                  </DbCommand>
                </SelectCommand>
              </DbSource>
            </MainSource>
            <Mappings>
              <Mapping SourceColumn="Product ID" DataSetColumn="Product ID" />
              <Mapping SourceColumn="Product Name" DataSetColumn="Product Name" />
              <Mapping SourceColumn="Product Description" DataSetColumn="Product Descri
            </Mappings>
```

```xml
                      <Sources>
                      </Sources>
                    </TableAdapter>
                  </Tables>
                  <Sources>
                  </Sources>
                </DataSource>
            </xs:appinfo>
          </xs:annotation>
          <xs:element name="DataSet1" msdata:IsDataSet="true" msdata:UseCurrentLocale="true" ms
            <xs:complexType>
              <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="Products" msprop:Generator_UserTableName="Products" msprop:G
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Product_x0020_ID" msprop:Generator_UserColumnName="Prod
                      <xs:element name="Product_x0020_Name" msprop:Generator_UserColumnName="Pr
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:maxLength value="100" />
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="Product_x0020_Description" msprop:Generator_UserColumnN
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:maxLength value="255" />
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="BarCode" msprop:Generator_UserColumnName="BarCode" mspr
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:complexType>
          </xs:element>
        </xs:schema>
```

DataSet1.xss

```xml
<?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="232" Y="126" Height="153" Width="19
  </Shapes>
```

```
  <Connectors />
</DiagramLayout>
```

Form1.Designer.cs

```csharp
namespace ReportFromSqlServer
{
        partial class Form1
        {
                private System.ComponentModel.IContainer components = null;

                protected override void Dispose(bool disposing)
                {
                        if (disposing && (components != null))
                        {
                                components.Dispose();
                        }
                        base.Dispose(disposing);
                }

                #region Windows Form Designer generated code

                private void InitializeComponent()
                {
                        this.crystalReportViewer1 = new CrystalDecisions.Windows.Forms
                        this.CrystalReport11 = new ReportFromSqlServer.CrystalReport1()
                        this.SuspendLayout();
                        //
                        // crystalReportViewer1
                        //
                        this.crystalReportViewer1.ActiveViewIndex = 0;
                        this.crystalReportViewer1.BorderStyle = System.Windows.Forms.Bo
                        this.crystalReportViewer1.Dock = System.Windows.Forms.DockStyle
                        this.crystalReportViewer1.Location = new System.Drawing.Point(
                        this.crystalReportViewer1.Name = "crystalReportViewer1";
                        this.crystalReportViewer1.ReportSource = this.CrystalReport11;
                        this.crystalReportViewer1.Size = new System.Drawing.Size(799,
                        this.crystalReportViewer1.TabIndex = 0;
                        //
                        // Form1
                        //
                        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
                        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
                        this.ClientSize = new System.Drawing.Size(799, 566);
                        this.Controls.Add(this.crystalReportViewer1);
                        this.Name = "Form1";
                        this.Text = "Form1";
                        this.ResumeLayout(false);

                }
```

```
                #endregion

                private CrystalDecisions.Windows.Forms.CrystalReportViewer crystalRepor
                private CrystalReport1 CrystalReport11;


        }
}
```

Form1.cs

```
using System;
using System.Data;
using System.Diagnostics;
using System.Windows.Forms;
using System.Data.SqlClient;
using Bytescout.BarCode;

namespace ReportFromSqlServer
{
        public partial class Form1 : Form
        {
                public Form1()
                {
                        InitializeComponent();

                        try
                        {
                                // MODIFY THE CONNECTION STRING WITH YOUR SERVER CONNEC
                                const string connectionString = "Data Source=localhost

                                using (SqlConnection connection = new SqlConnection(cor
                                {
                                        connection.Open();

                                        // Create a database for demonstration purposes
                                        ////////////////////////////////////////////////

                                        Object o = ExecuteQueryScalar(connection, "SEL

                                        // if 'example_db' does not exist, create it
                                        if (o == null || o is DBNull)
                                        {
                                                // Create empty database
                                                ExecuteQueryWithoutResult(connection,
                                                // Switch to created database
                                                ExecuteQueryWithoutResult(connection,
                                                // Create a table
                                                ExecuteQueryWithoutResult(connection,
```

```csharp
                                    // Fill the table with data
                                    ExecuteQueryWithoutResult(connection,
                                    ExecuteQueryWithoutResult(connection,
                                    ExecuteQueryWithoutResult(connection,
                                    ExecuteQueryWithoutResult(connection,
                        }

                        // Create a datataset from query.
                        // Query result columns must conform to field

                        SqlDataAdapter dataAdapter = new SqlDataAdapter

                        // fill dataset
                        DataSet dataSet = new DataSet();
                        dataAdapter.Fill(dataSet);

                        // don't forget to close the connection
                        connection.Close();

                        // add virtual column into the result table
                        dataSet.Tables[0].Columns.Add(new DataColumn("

                        // create barcode object
                        Barcode bc = new Barcode(SymbologyType.Code39)
                        bc.DrawCaption = false;

                        foreach (DataRow row in dataSet.Tables[0].Rows)
                        {
                                    // set barcode value
                                    bc.Value = (Convert.ToString(row["Produ

                                    // retrieve generated image bytes
                                    byte[] barcodeBytes = bc.GetImageBytes

                                    // fill virtual column with generated
                                    row["BarCode"] = barcodeBytes;
                        }

                        // set report datasource
                        CrystalReport11.SetDataSource(dataSet.Tables[0]
                }
            }
            catch (Exception ex)
            {
                    Trace.WriteLine("Error: " + ex.Message);
            }
        }

        private static void ExecuteQueryWithoutResult(SqlConnection connection
        {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                        command.ExecuteNonQuery();
                }
        }

        private static object ExecuteQueryScalar(SqlConnection connection, str
        {
                using (SqlCommand command = new SqlCommand(query, connection))
```

```
                    {
                            return command.ExecuteScalar();
                    }
            }
    }
}
```

Program.cs

```csharp
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace ReportFromSqlServer
{
        static class Program
        {
                [STAThread]
                static void Main()
                {
                        Application.EnableVisualStyles();
                        Application.Run(new Form1());
                }
        }
}
```

ReportFromSqlServer.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 14
VisualStudioVersion = 14.0.25420.1
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ReportFromSqlServer", "ReportFrom
EndProject
Global
        GlobalSection(SolutionConfigurationPlatforms) = preSolution
                Debug|Any CPU = Debug|Any CPU
                Release|Any CPU = Release|Any CPU
        EndGlobalSection
```

```
        GlobalSection(ProjectConfigurationPlatforms) = postSolution
                {74C1A3D5-3DBB-44B7-8DC4-0D08576418FB}.Debug|Any CPU.ActiveCfg = Debug
                {74C1A3D5-3DBB-44B7-8DC4-0D08576418FB}.Debug|Any CPU.Build.0 = Debug|A
                {74C1A3D5-3DBB-44B7-8DC4-0D08576418FB}.Release|Any CPU.ActiveCfg = Rel
                {74C1A3D5-3DBB-44B7-8DC4-0D08576418FB}.Release|Any CPU.Build.0 = Relea
        EndGlobalSection
        GlobalSection(SolutionProperties) = preSolution
                HideSolutionNode = FALSE
        EndGlobalSection
EndGlobal
```

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