TBarCode .NET

Barcode Generator Software for .NET

Version 1.2.0

Barcodes in Microsoft SQL Server Reporting Services (SSRS)

14 March 2017

TEC-IT Datenverarbeitung GmbH
Hans-Wagner-Str. 6
A-4400 Steyr, Austria

t ++43 (0)7252 72720
f ++43 (0)7252 72720 77
office@tec-it.com
www.tec-it.com

1 Content

2 Disclaimer

The actual version of this product (document) is available as is. TEC-IT declines all warranties which go beyond applicable rights. The licensee (or reader) bears all risks that might take place during the use of the system (the documentation). TEC-IT and its contractual partners cannot be penalized for direct and indirect damages or losses (this includes non-restrictive, damages through loss of revenues, constriction in the exercise of business, loss of business information or any kind of commercial loss), which is caused by use or inability to use the product (documentation), although the possibility of such damage was pointed out by TEC-IT.

TEC-IT Datenverarbeitung GmbH
Wagnerstr. 6

A-4400 Austria
t.: +43 (0)7252 72720
f.: +43 (0)7252 72720 77
http://www.tec-it.com

3 Haftungsausschluss

Dieses Produkt (bzw. Dokument) steht Ihnen in der aktuellen Version „WIE BESEHEN – ohne Gewährleistung“ zur Verfügung. TEC-IT weist alle Garantien, die über das anwendbare Recht hinausgehen, zurück. Risiken, die aus der Benutzung des Produkts und der Dokumentation entstehen, trägt der Lizenznehmer bzw. Benutzer. TEC-IT und seine Vertragspartner dürfen nicht für direkte oder indirekte Schäden oder Verluste belangt werden (dies beinhaltet, uneingeschränkt, Schäden durch den Verlust von Einkünften, Einschränkungen in der Geschäftsausübung, Verlust von Geschäftsinformationen sowie andere wirtschaftliche Verluste), die aus der Benutzung oder Unfähigkeit zur Benutzung des Produkts (der Dokumentation) entstanden sind, selbst wenn TEC-IT auf die Möglichkeit solcher Schäden hingewiesen hat.

TEC-IT Datenverarbeitung GmbH
Wagnerstr. 6

A-4400 Austria
t.: +43 (0)7252 72720
f.: +43 (0)7252 72720 77
http://www.tec-it.com

4 Before You Start

4.1 MS SQL Server Reporting Services and TBarCode .NET

In the following samples we will create an SSRS report with barcode images using *TBarCode .NET* and Visual Studio .NET.

There is a method with direct integration of the component into a report (bar code generation “on the fly”) and a method with indirect integration (bar code images stored in the database before running the report).

Beside that we offer URL based bar code generation – this method is recommended if you want to generate the bar codes during report runtime and avoid installing TBarCode on the Client or on the Reporting Server. The bar code generator service is instead hosted on a web server (IIS).

4.1.1 Prerequisites

Basically you need to install *TBarCode SDK* on your system. You can download a free evaluation version of TBarCode directly from our web site:

http://www.tec-it.com/download/tbarcode/windows/Download.aspx

From the SDK we will use the *TBarCode .NET* component. The MSI setup will install the *TBarCode .NET* assemblies into the GAC[1].

Additional you need the following applications installed:

Microsoft SQL Server – Reporting Services

Microsoft .NET Framework 3.5 or higher

Direct Integration: Microsoft Visual Studio .NET 2008 or higher

Other Samples: Microsoft Visual Studio .NET 2015 or higher

[1] For alternative assembly locations see here: https://msdn.microsoft.com/en-us/library/ms155034(v=sql.110).aspx

5 TBarCode.NET Direct Integration (Method 1)

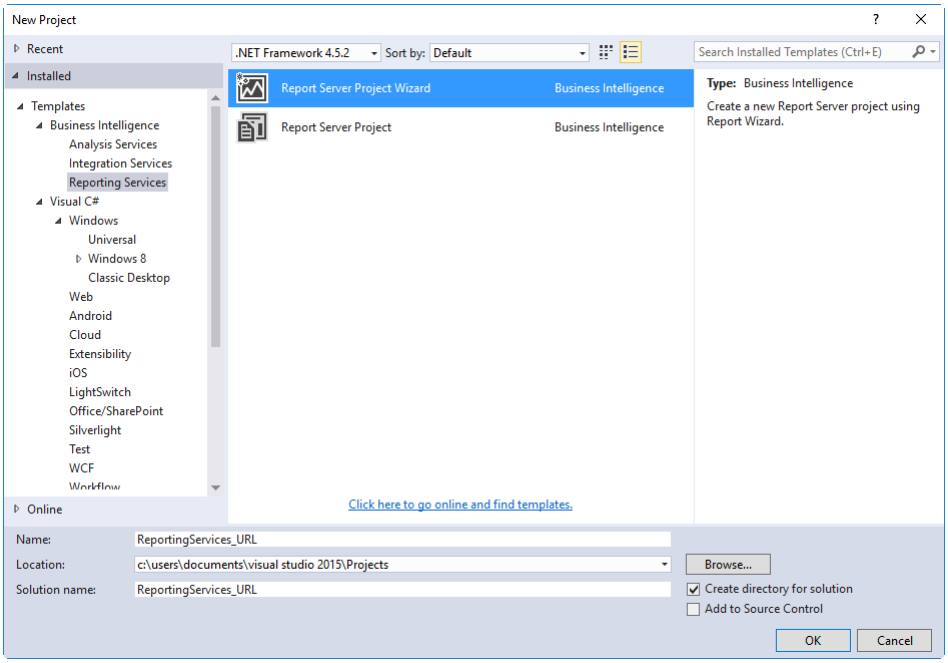
The following method integrates bar code generation directly into the report. Each time the report is run, the bar codes are generated “on the fly” by the barcode generator SDK and are then inserted into the image box on the report.

The advantage of this solution is that you don’t have to change anything in the database and the bar codes are always up to date. Disadvantage of this method is that you need to deal with security and trusted code modules if you run the report on the server. That is why we recommend this method only if you render the report on the Client as we show in our sample code.

For this solution we provide sample code and a sample Visual Studio project, which uses a local report.

5.1 Creating the Reporting Project

Open Visual Studio 2015 and create a new report project



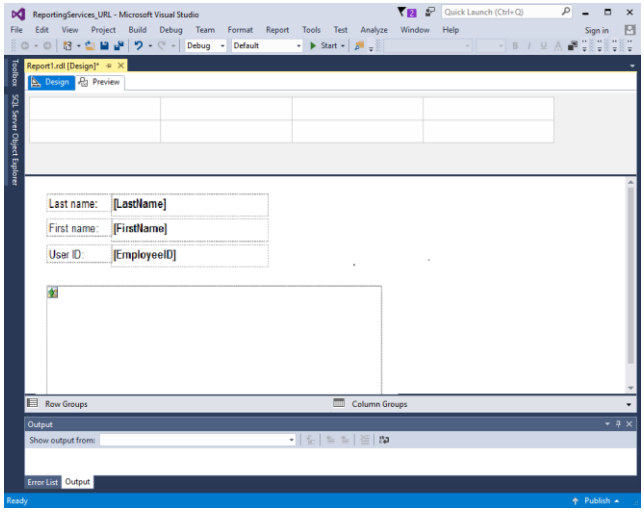
In the project wizard choose your database connection (in the sample the Northwind database[2] is used).

[2] Download: https://northwinddatabase.codeplex.com/

5.2 Report Design

Open the automatically generated “Report1.rdlc” and design your report as you want.

Insert an image object into your report at the position where you want to create the barcode in.

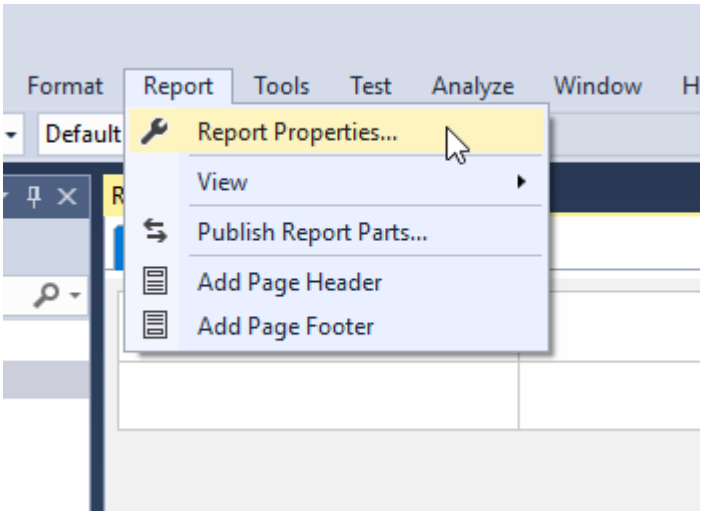


5.3 Creating the Barcode

In this step we’ll create the barcode. This sample creates a Code 128 barcode, but with TBarCode you’re able to create over 100 different barcode types. A list of all supported barcode symbologies is available here: Barcode symbologies supported by TBarCode

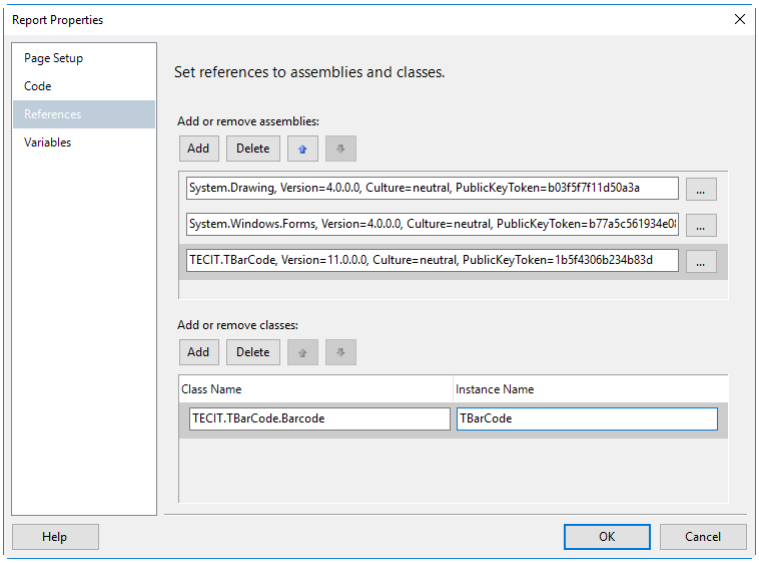
5.3.1 Report Properties

Open the Report1.rdlc in Visual Studio 2015. In the menu select *Report* → *Report Properties*…



Open the *References* tab and add the following references:

- TECIT.TBarCode
- System.Drawing
- System.Windows.Forms



Now you have to create an instance of TBarCode. Therefore add the following *Class name*:

TECIT.TBarCode.Barcode

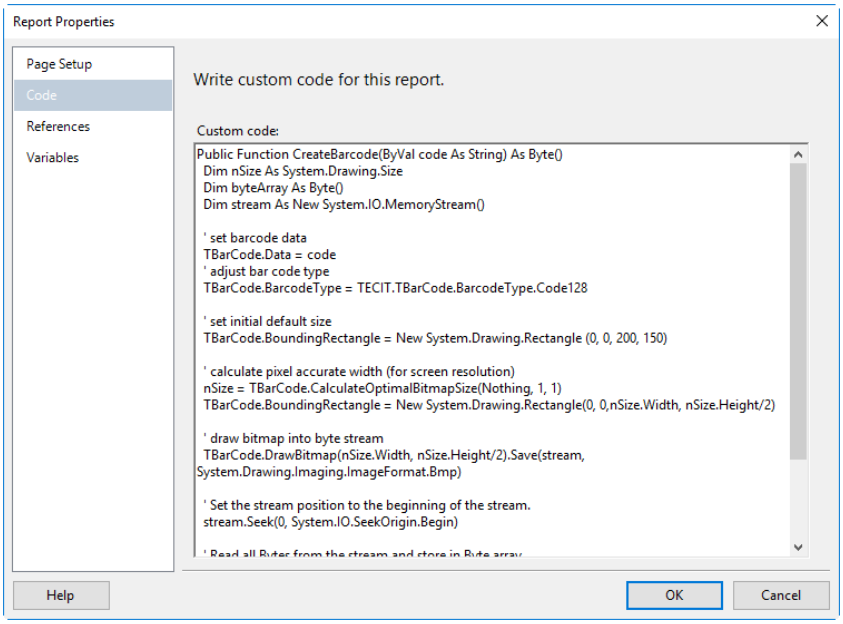
For the *Instance name* you use:

TBarCode

5.3.2 Barcode Generator Code (VB)

In the next step you have to open the *Code* tab. Enter the function as specified below. The basic principle is to generate a barcode stream which is than shown in the image object.

```
Public Function CreateBarcode(ByVal code As String) As Byte()
    Dim nSize As System.Drawing.Size
    Dim byteArray As Byte()
    Dim stream As New System.IO.MemoryStream()

    ' Set the barcode data to encode
    TBarCode.Data = code

    ' Set the barcode symbology
    TBarCode.BarcodeType = TECIT.TBarCode.BarcodeType.Code128

    ' Set initial default size
    TBarCode.BoundingRectangle = New System.Drawing.Rectangle (0, 0, 200, 150)

    ' calculate pixel accurate width (for screen resolution)
    nSize = TBarCode.CalculateOptimalBitmapSize(Nothing, 1, 1)
    TBarCode.BoundingRectangle = New System.Drawing.Rectangle(0, 0,_
                        nSize.Width, nSize.Height)

    ' Create the barcode image (BMP) and write to stream
    TBarCode.DrawBitmap(nSize.Width, nSize.Height).Save(stream, _
                                System.Drawing.Imaging.ImageFormat.Bmp)

    ' Set the stream position to the beginning of the stream.
    stream.Seek(0, System.IO.SeekOrigin.Begin)

    ' Read all Bytes from the stream.
    byteArray = New Byte(CType(stream.Length, Integer)) {}
    stream.Read(byteArray, 0, stream.Length)

    Return byteArray
End Function
```



The code above is for generic use (linear and 2D Codes). If you have specific requirements you need to adjust the barcode properties in detail. See Appendix for a 2D Barcode (Data Matrix) Code Sample.

5.4 Security ➔ Trusted Code Modules

To use the specified assemblies you have to trust them[3]. Therefore open the *Form* which was created with the project. Open this form in the *Code View* and add a *Load* function with the following lines:

```
Me.ReportViewer1.LocalReport.ExecuteReportInCurrentAppDomain(AppDomain.CurrentDomain.Evidence)
Me.ReportViewer1.LocalReport.AddTrustedCodeModuleInCurrentAppDomain("TECIT.TBarCode,
Version=11.0.0.0, Culture=neutral, PublicKeyToken=1b5f4306b234b83d")
Me.ReportViewer1.LocalReport.AddTrustedCodeModuleInCurrentAppDomain("System.Drawing,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
Me.ReportViewer1.LocalReport.AddTrustedCodeModuleInCurrentAppDomain("System.Windows.Forms,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089")
```

This moves the report into the actual app domain which treats our code modules as trusted. Additionally, if you use .NET Framework 4 or higher, you have to enable the Trusted Code Modules by adding the following lines into the app.config file:

```
<configuration>
   <runtime>
      <NetFx40_LegacySecurityPolicy enabled="true"/>
   </runtime>
</configuration>
```

Finally you have to change the settings of the image object in the report. Set the *Source* property to ***Database*** and the *Value* property to call the *CreateBarcode*.

For example:
=Code.CreateBarcode(Fields!LastName.Value.ToString())

[3] More information:
http://blogs.msdn.com/b/mosharaf/archive/2005/12/20/localreportcustomcode.aspx
https://msdn.microsoft.com/en-us/library/ms155034(v=sql.110).aspx

6 Embed Images / Indirect Integration (Method 2)

The following method generates bar codes as image data and stores them in the database. The report reads the image data from the database and displays the bar codes during runtime.

The advantage of this solution is that no additional code modules are required in the report (you don't have to deal with security policy settings of custom assemblies). The disadvantage is that you have to adapt your database table(s) and update/insert the bar code images whenever the encoded data changes.

► For Microsoft Dynamics NAV® 2013 and later we recommend the Indirect Integration (Method 2) or URL Integration (Method 3).

The following tutorial / sample code shows how to

- Add a binary field in your table
- Create a bar code image (GIF) and stores it into the binary field in the table
- Display the bar code image from the binary field in your SSRS report

6.1 Table

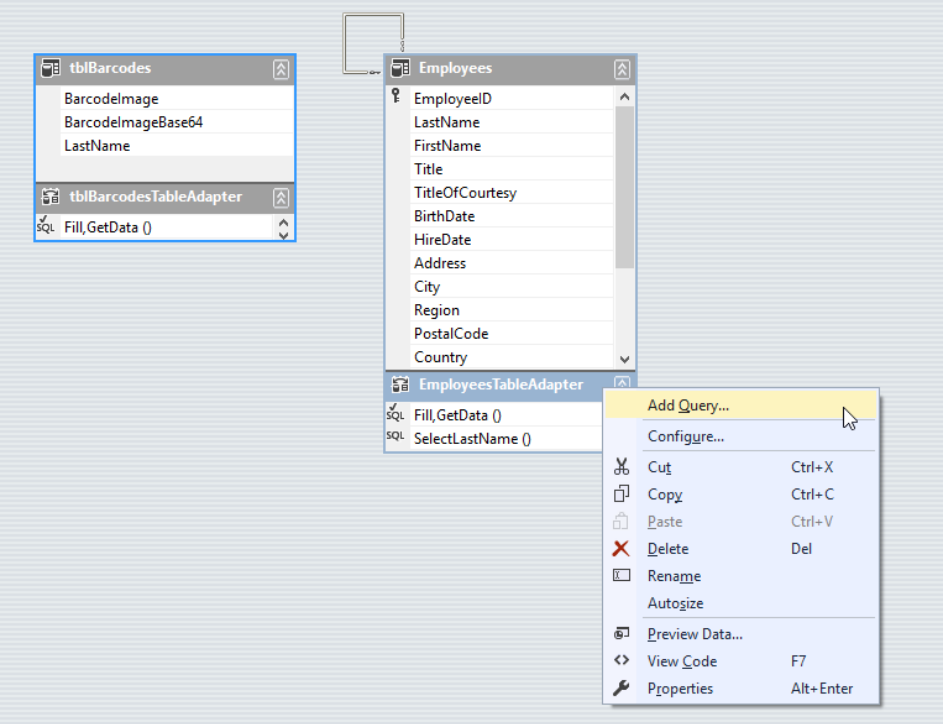
For this tutorial we have created a table, which has a *varbinary* field for storing the bar code image in binary format, a *varchar* field for storing the barcode image in base64 encoded form and also a *nchar* field for storing the data (in case of this sample the last name) encoded in the barcode image.

Column Name	Data Type	Allow Nulls
BarcodeImage	varbinary(MAX)	☑
BarcodeImageBase64	varchar(MAX)	☑
LastName	nchar(20)	☑
		☐

You can use one of the two field variants depending on your database. In our sample we use both field variants for demonstration.

6.2 Data Set

In our Visual Studio project, we have created two data sets as follows:



To clear the tblBarcodes table, we have to add a Delete query to the tblBarcodesTableAdapter:

```
Delete from tblBarcodes;
```

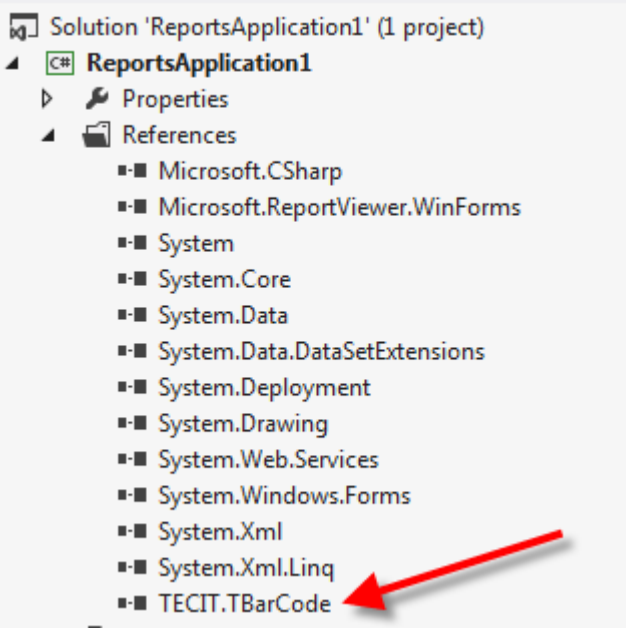
And for selecting the LastName as data we need to add a Select query to the EmployeesTableAdapter:

```
Select LastName from Employees;
```

This will help us to update and insert the bar code images. You can use also other methods (e.g. LINQ) or whatever you want.

6.3 References

Prerequisite for bar code generation: Add a reference to the assembly *TECIT.TBarCode.dll* to your project (see TBarCode .NET Developer manual, section 8 for details):



6.4 Barcode Generator Code (C# .NET)

We need this barcode creation function.

6.4.1 Linear Code (Code-128)

```
public Byte[] GenerateBarcodeStream (String data)
{
  TECIT.TBarCode.Barcode barcode = new TECIT.TBarCode.Barcode();
  barcode.BarcodeType = TECIT.TBarCode.BarcodeType.Code128;
  barcode.Data = data;
  barcode.Font = new System.Drawing.Font("Arial", 10f);
  barcode.Dpi = 100;
  Size optimalSize = barcode.CalculateOptimalBitmapSize(null, 1, 1);
  barcode.BoundingRectangle = new Rectangle(0, 0, optimalSize.Width, 60 /* 60px */);

  using (System.IO.MemoryStream ms = new System.IO.MemoryStream())
  {
    Bitmap bitmap = barcode.DrawBitmap();
    bitmap.Save(ms, System.Drawing.Imaging.ImageFormat.Gif);
    ms.Position = 0;
    return ms.ToArray();
  }
}
```

6.4.2 2D Code (Data Matrix)

```
public Byte[] GenerateBarcodeStream2D (String data)
{
  TECIT.TBarCode.Barcode barcode = new TECIT.TBarCode.Barcode();
  barcode.BarcodeType = TECIT.TBarCode.BarcodeType.DataMatrix;
  barcode.Data = data;
  barcode.Dpi = 100;
  Size optimalSize = barcode.CalculateOptimalBitmapSize(null, 2, 2);
  barcode.BoundingRectangle = new Rectangle(0, 0, optimalSize.Width,
    optimalSize.Height);

  using (System.IO.MemoryStream ms = new System.IO.MemoryStream())
  {
    Bitmap bitmap = barcode.DrawBitmap();
    bitmap.Save(ms, System.Drawing.Imaging.ImageFormat.Gif);
    /*
    // or save as 24 bit JPG
    ImageCodecInfo jpgCodec = ImageCodecInfo.GetImageEncoders().Where(codec =>
                  codec.FormatID.Equals(ImageFormat.Jpeg.Guid)).FirstOrDefault();
    System.Drawing.Imaging.EncoderParameters parameters = new
                  System.Drawing.Imaging.EncoderParameters();
    parameters.Param[0] = new EncoderParameter
                  (System.Drawing.Imaging.Encoder.ColorDepth, 24);
    bitmap.Save(ms, jpgCodec, parameters);
    */
    ms.Position = 0;
    return ms.ToArray();
  }
}
```

6.5 Insert Barcode Image for One Record

If you want to insert one record with a bar code image, you could to it like that:

```
Byte[] barcodeImageStream = GenerateBarcodeImage ("12345");

DataRow myRow = TestDataSet.tblBarcodes.NewRow();
myRow [0] = "12345";
myRow [1] = barcodeImageStream;                          // add as binary data stream
myRow [2] = Convert.ToBase64String(barcodeImageStream); // add as base64 string
TestDataSet.tblBarcodes.Rows.Add(anyRow);

tblBarcodesTableAdapter.Update(TestDataSet);
TestDataSet.tblBarcodes.AcceptChanges();
```

After creating the barcode image and writing the bitmap to the table it looks as follows:

	BarcodeImage	BarcodeImageBase64	LastName
1	0x4749463839611C001C00F70000000000000033000066000...	R0lGODlhcAA8APcAAAAAAAAAMwAAZgAAmQAAzAAA/wArAAArM...	Davolio
2	0x4749463839611C001C00F70000000000000033000066000...	R0lGODlhZQA8APcAAAAAAAAAMwAAZgAAmQAAzAAA/wArAAArM...	Fuller

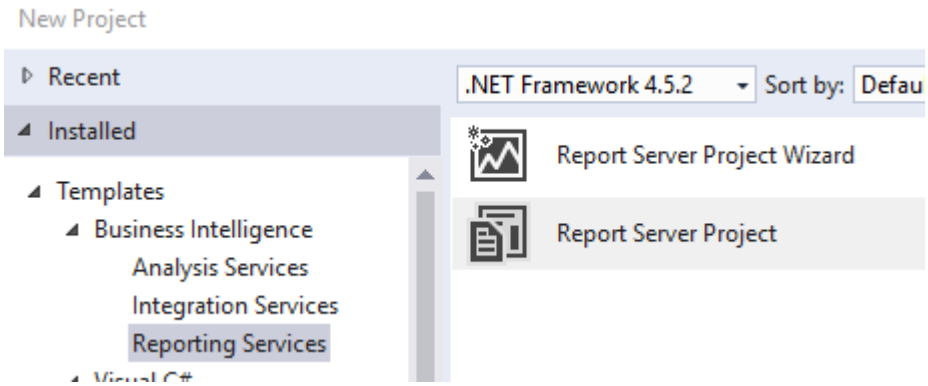
6.6 Update Barcode Images for All Records

If you want to iterate through a table and convert a specific field (*LastName* in this sample) into a bar code image field, you could do it that way:

```
public void insertIntoDB()
        {
            //initializeis filds
            //adapter for bar code table
            tblBarcodesTableAdapter tableAdapterBar = new tblBarcodesTableAdapter();
            //adapter fpr employees table
            EmployeesTableAdapter tableAdapterEmp = new EmployeesTableAdapter();
            DataSet1 dataSet = new DataSet1();
            //select rows from Employees
            DataSet1.EmployeesDataTable empData = tableAdapterEmp.GetData();
            tableAdapterBar.DeleteQuery(); //delete all entryes in bar code table
            foreach (DataRow empRow in empData)
            {
                if (empRow["EmployeeID"] != null)
                {
                    //LastName is used as data source for bar code generation
                    string myData = empRow["LastName"].ToString().Trim();
                    try
                    {
                        //generate bar code  Data Matrix
                        Byte[] barcodeImageStream = GenerateBarcodeStream2D(myData);
                        //generate bar code  Code128
                        Byte[] barcodeImageStream2 = GenerateBarcodeStream(myData);
                        //insert images into bar code table
                        int ret = tableAdapterBar.Insert(barcodeImageStream,
                            Convert.ToBase64String(barcodeImageStream2), myData);
                        dataSet.tblBarcodes.AcceptChanges();
                        Console.WriteLine("Barcode " + myData + " inserted!
                            Returnvalue= " + ret);
                    }
                    catch (Exception ex)
                    {
                        // error in bar code generation (invalid data?)
                    }
                }
            }
        }
```

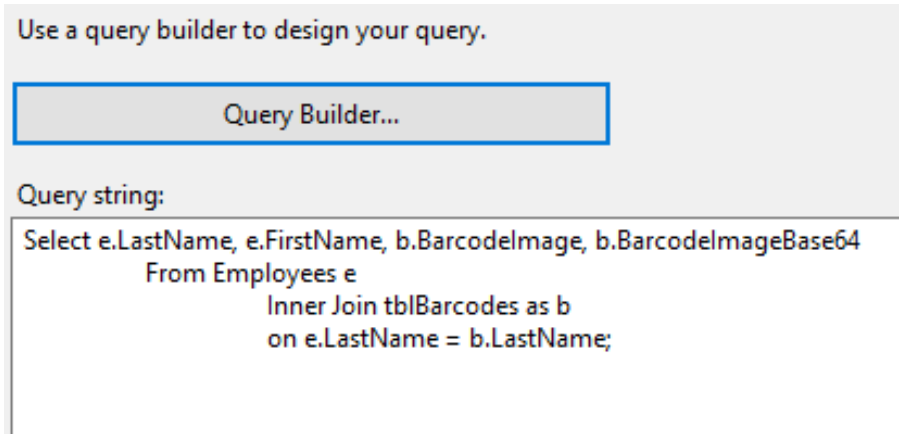
6.7 Display Barcode Image in Report

Open Visual Studio 2015 and create a new report project



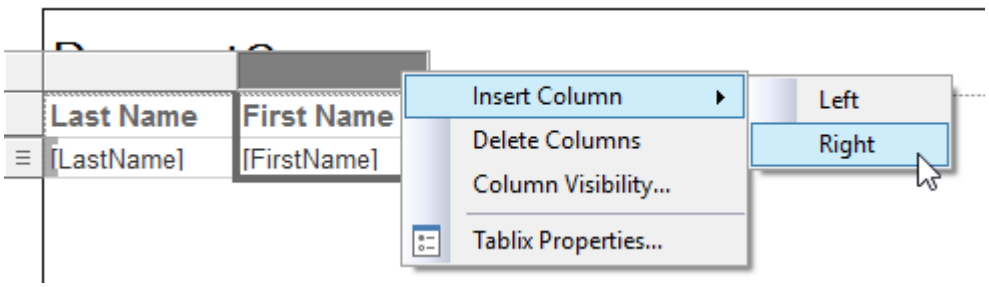
In the project wizard choose your database connection (in the sample the Northwind database[4] is used). Then create a new report and insert the following SQl Select statement as query:

```
Select e.LastName, e.FirstName, b.BarcodeImage, b.BarcodeImageBase64
       From Employees e
              Inner Join tblBarcodes as b
              on e.LastName = b.LastName;
```



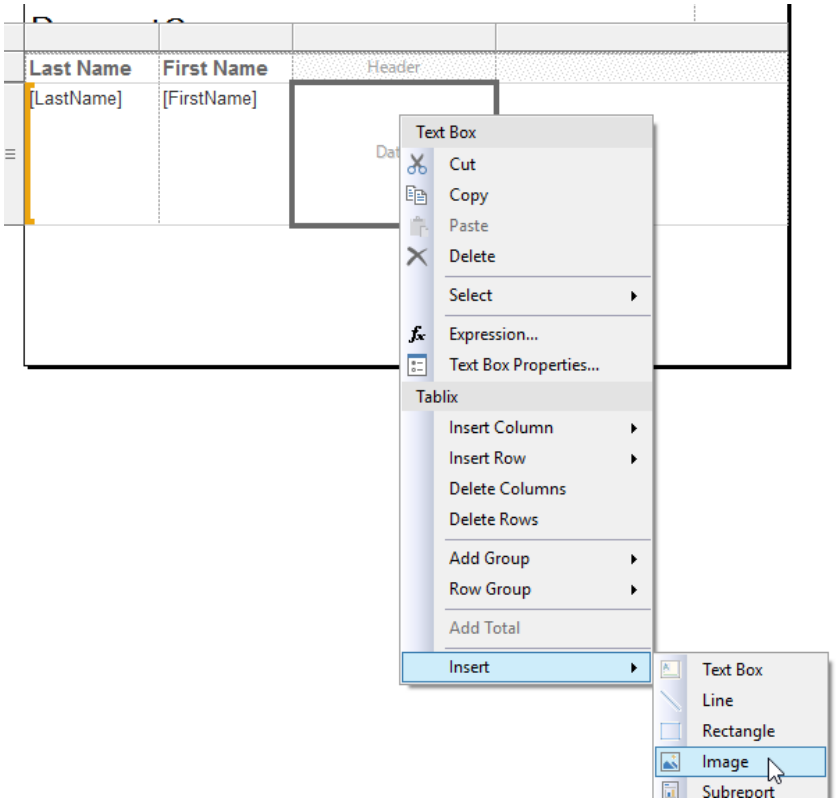
In this sample a table based report is used and the fields *LastName* and *FirstName* are added as Details.

Afterwards you have to insert two extra columns for the bar codes.

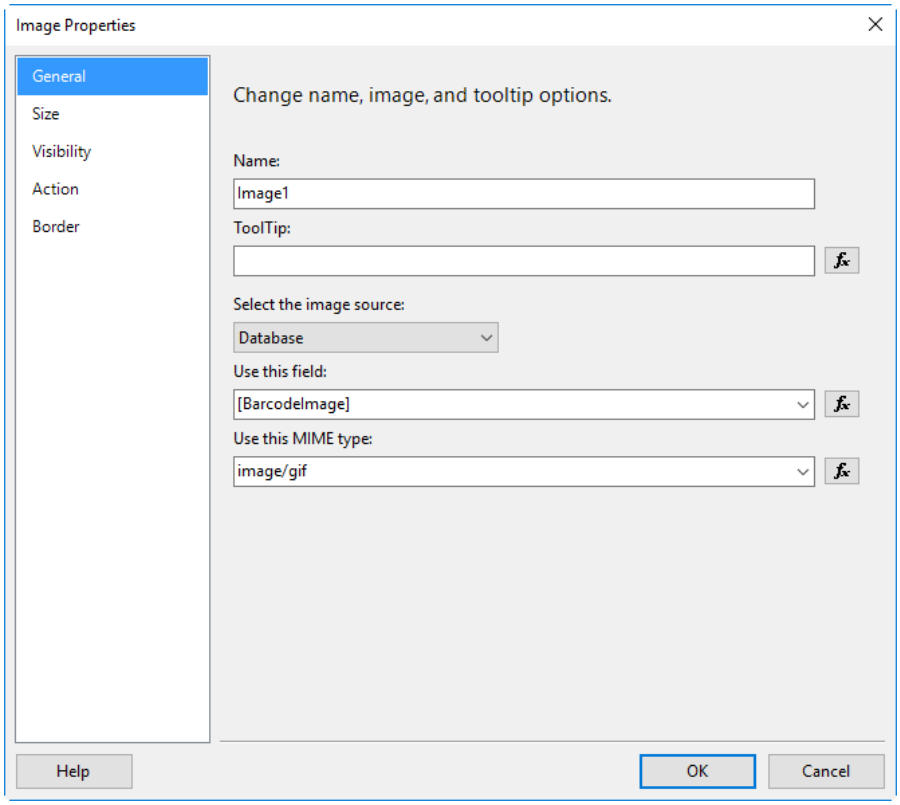


[4] Download: https://northwinddatabase.codeplex.com/

In order to display the bar code image in your report you have to insert an *Image* report item from the toolbox inside the new added column:

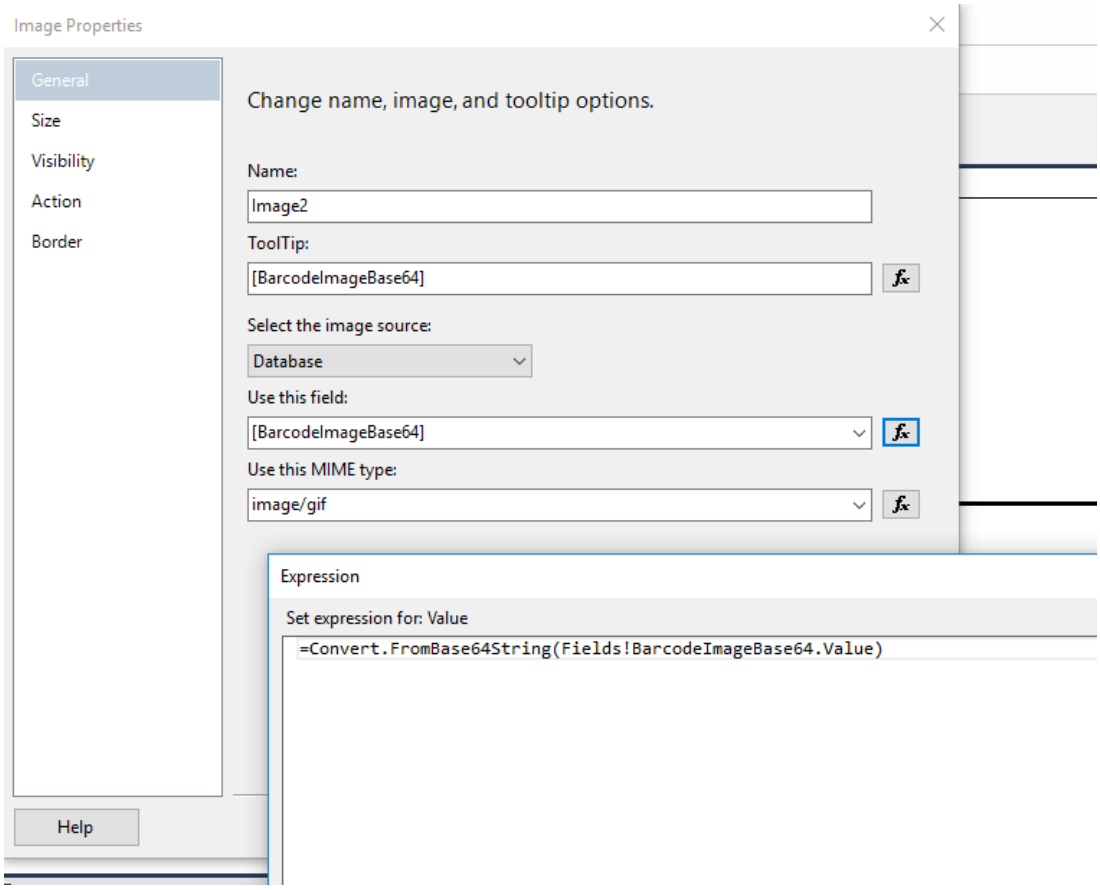


Then you have to adjust the properties as follows:



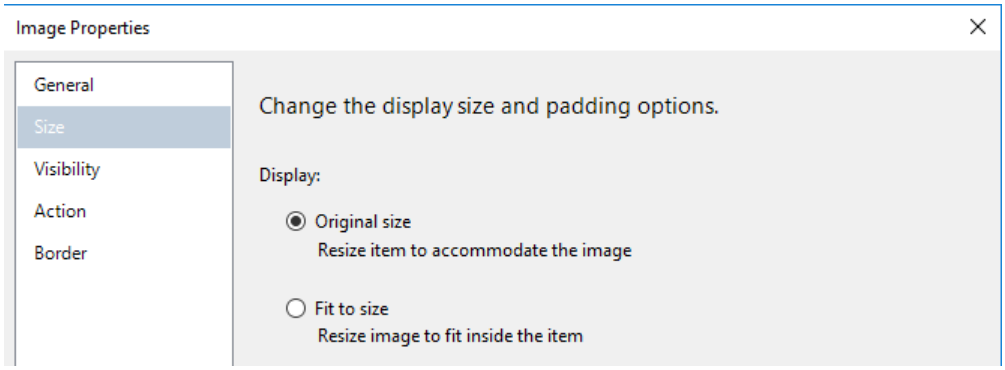
For the image stored in base64 format you need to convert it back to binary format by using an expression:

```
=Convert.FromBase64String(Fields!BarcodeImageBase64.Value)
```



6.7.1 Image Size

If you take a look into our bar code generation routine you will see the command *CalculateOptimalBitmapSize*. That means the size is already optimized for best quality and therefore we should keep the original size of the image:

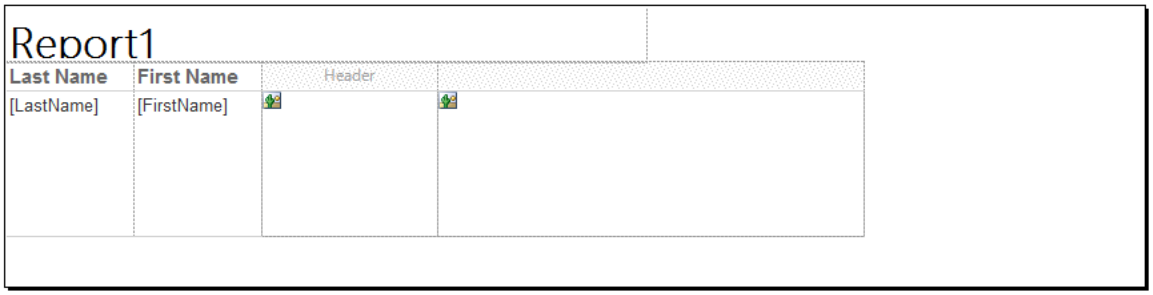


- If you want to change the size, modify the parameters in the *CalculateOptimalBitmap* function.
- For linear bar codes adjust the height in Pixels in the *BoundingRectangle*.

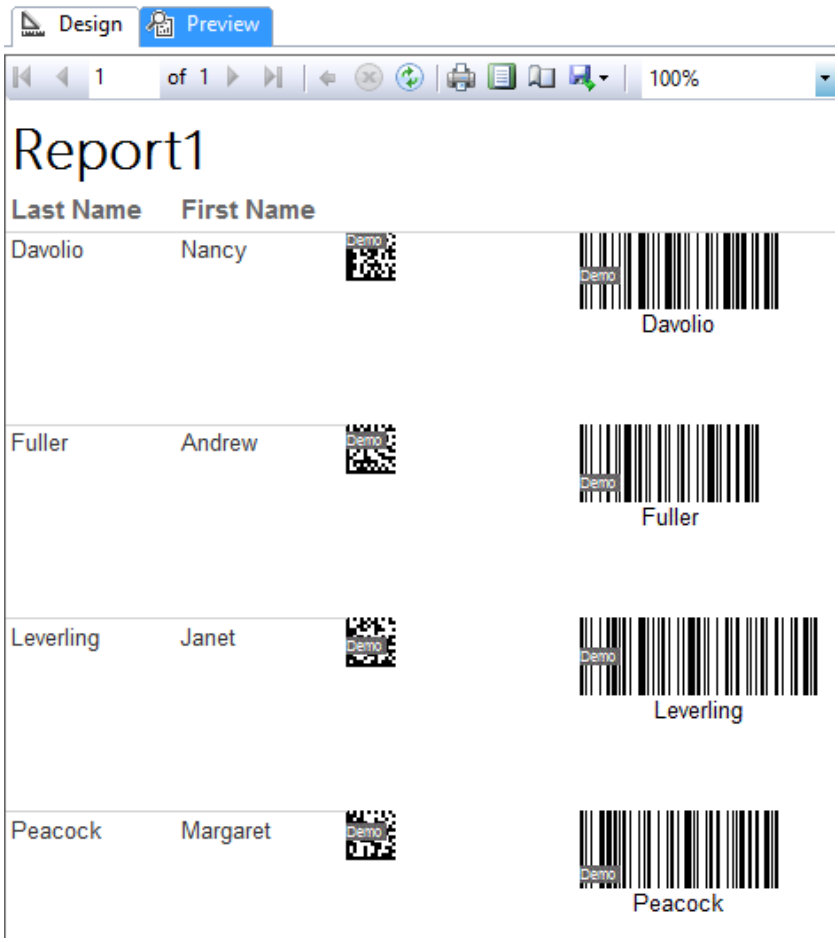
- If this does not lead to the desired result or you have specific requirements depending on your printer resolution (DPI) you can change the image properties to *Fit proportional* and use a pre-calculated size.
- We recommend a high printing quality/output resolution and test scans.

6.8 Report Output

With the following (simple) report design



you will get the following output when running the report:



7 URL Based Barcode Integration (Method 3)

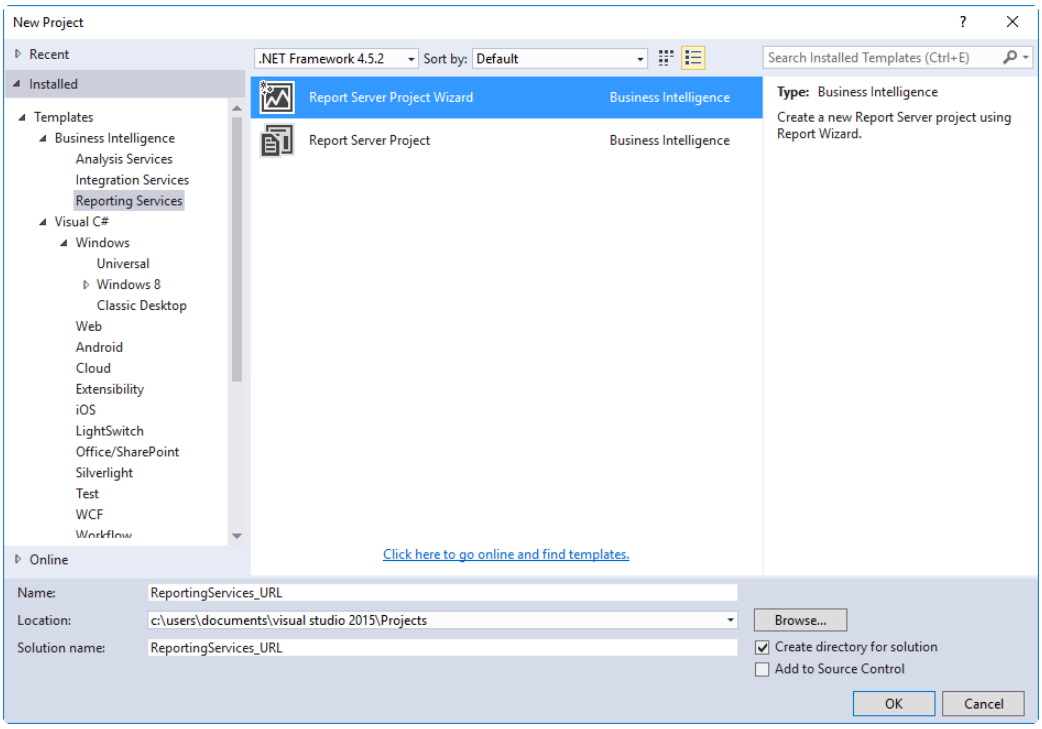
The following method integrates bar code generation directly into the report by using an image box, the file-name of the image to be printed is specified as an http/https-URL to an online barcode generator service. Each time the report is run, the bar codes are generated on the fly.

The advantage of this solution is that you don't have to change anything in the database and the bar codes are always "up to date". The barcode generator service can be hosted on the internet or on your intranet, the barcode creation can be controlled via URL parameters.

- ATTENTION: This sample uses the online barcode generator service http://barcode.tec-it.com. Please note, that this barcode generator service must not be used for production use or bulk testing. Please contact sales@tec-it.com for details.
- TBarCode SDK comes with a ready to use barcode generator service[5] – ideal for your intranet.

7.1 Creating the Reporting Project

Open Visual Studio 2015 and create a new report project



In the project wizard choose your database connection (in the sample the Northwind database[6] is used) and select the fields you want to put into your report.

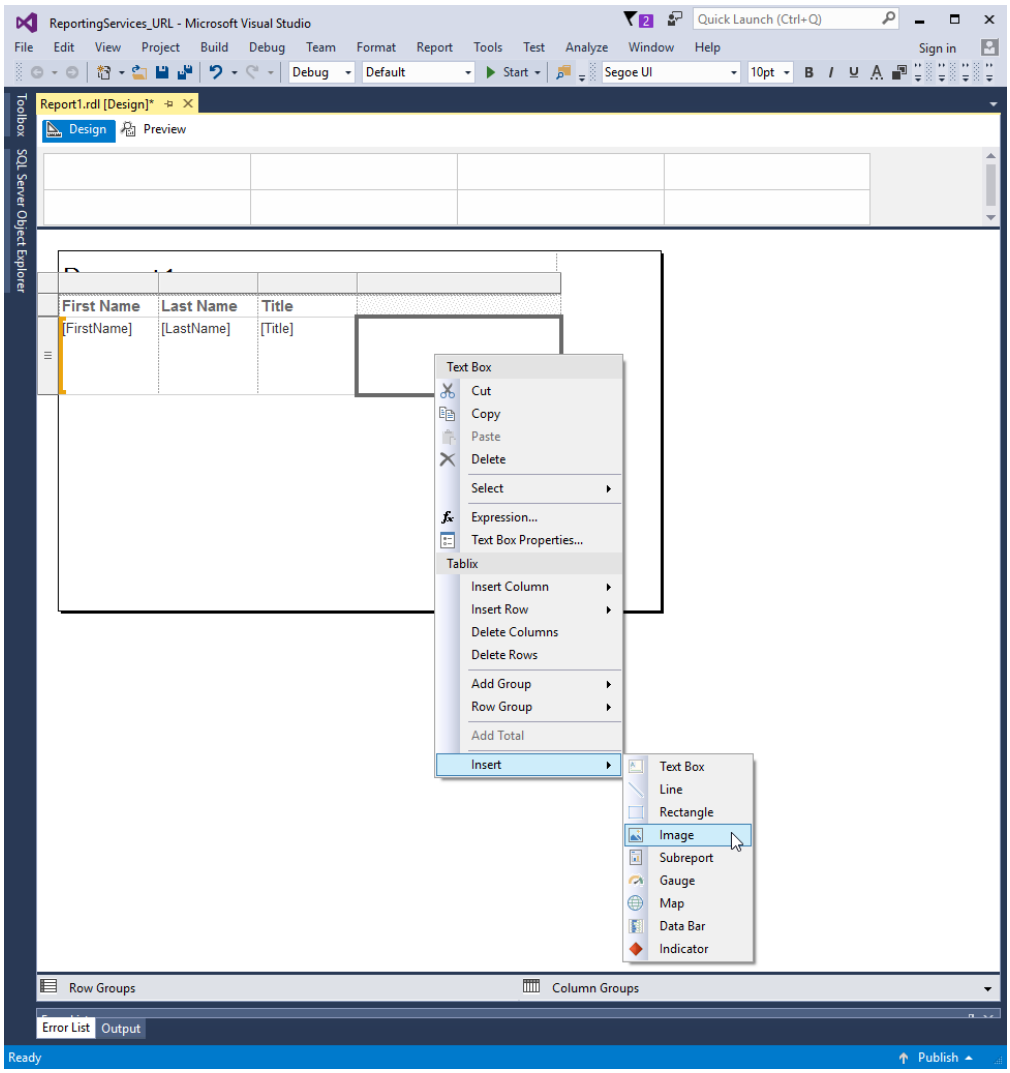
[5] For this solution you need a TBarCode SDK Web License. Please contact sales@tec-it.com for more information.
[6] Download: https://northwinddatabase.codeplex.com/

7.2 Report Design

Use the wizard to design your report as you want.

Insert an extra column for the generated barcode.

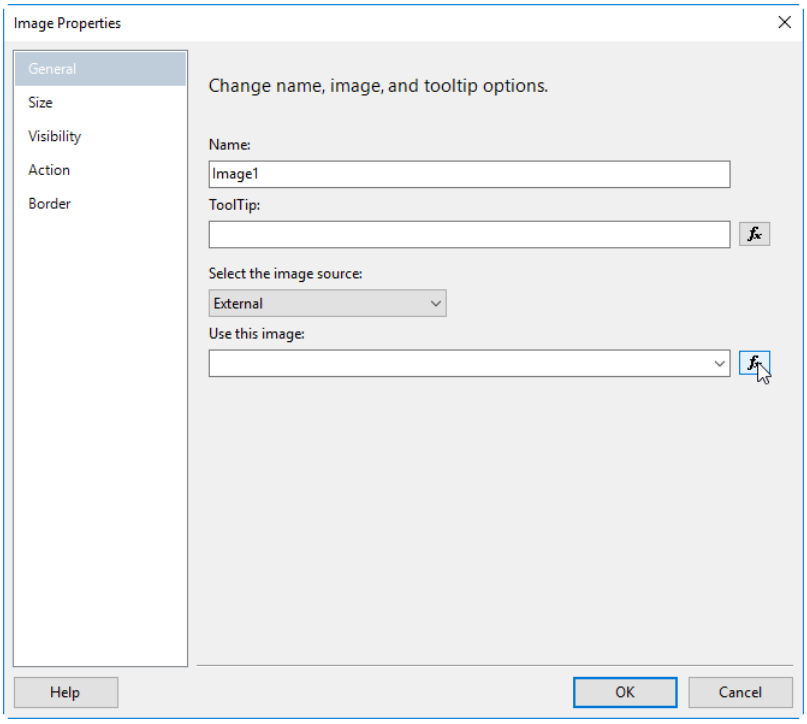
Insert an image object into the new column, where you want to create the barcode in.



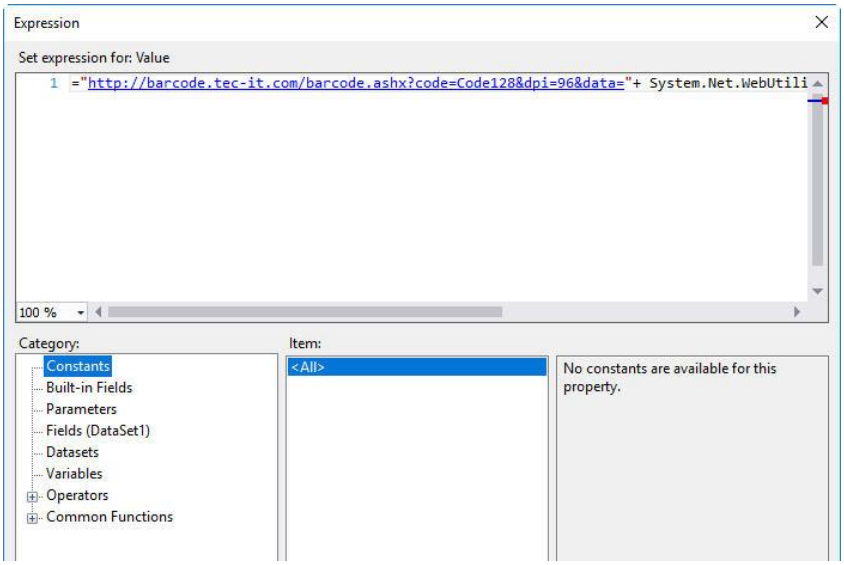
7.3 Creating the Barcode

In the Image Properties window select External as image source.

Klick on the expression icon to open the expression window.



In the expression window insert the required URL. For this sample, the online Barcode Generator from http://barcode.tec-it.com/en is used.



Finally add the data value which should be encoded as barcode. For this sample, the LastName field from the Employees table is used:

```
="http://barcode.tec-it.com/barcode.ashx?code=Code128&dpi=96&data="+
                    System.Net.WebUtility.UrlEncode(Fields!LastName.Value)
```

Make sure that you use UrlEncode for your data.

► ATTENTION: This sample uses the online barcode generator service http://barcode.tec-it.com. Please note, that this barcode generator service must not be used for production use or bulk testing. Please contact sales@tec-it.com for details.

Appendix A Code Examples

If you have special requirements and don't know how to set parameters, please contact us.

A.1 Generate Data Matrix Bitmap 44x44

The code below is for direct integration (method 1) and show how to adjust the Data Matrix properties for the following requirements:

- Dot Size = 0.254 mm suitable for 300 DPI
- Data Matrix Version = 44 x 44
- Quiet Zone = 4 Dots each side
- Data Matrix Macro 06 Header/Footer: [)><RS> 06<GS> + DATA + <RS><EOT>
- Encode <GS> via sequence \x1d
- Image Fit to size = (44 + 4 + 4) * 0.254 mm = 13.21 mm width/height

```
Public Function CreateDataMatrixBarcode (ByVal code As String) As Byte()

  Dim nSize As System.Drawing.Size
  Dim byteArray As Byte()
  Dim stream As New System.IO.MemoryStream()

  ' set barcode data
  TBarCode.Data = code

  ' enable escapesequences (e.g. \x1d = GS)
  TBarCode.TranslateEscapeSequences = True

  ' adjust bar code type and Data Matrix properties
  TBarCode.BarcodeType = TECIT.TBarCode.BarcodeType.DataMatrix
  TBarCode.DataMatrix.Size = TECIT.TBarCode.DataMatrixSize.Square44x44
  TBarCode.DataMatrix.Format = TECIT.TBarCode.DataMatrixFormat.Macro06

  ' adjust quiet zone
  Dim sizeQZ = 4
  TBarCode.QuietZones.Unit = TECIT.TBarCode.QuietZoneUnit.Modules
  TBarCode.QuietZones.Left.Size = sizeQZ
  TBarCode.QuietZones.Right.Size = sizeQZ
  TBarCode.QuietZones.Top.Size = sizeQZ
  TBarCode.QuietZones.Bottom.Size = sizeQZ

  ' Set initial default size
  TBarCode.BoundingRectangle = New System.Drawing.Rectangle(0, 0, 44 + sizeQZ, 44 + sizeQZ)

  TBarCode.Dpi = 300

  ' calculate pixel accurate width and height
  nSize = TBarCode.CalculateOptimalBitmapSize(Nothing, 1, 1)
  TBarCode.BoundingRectangle = New System.Drawing.Rectangle(0, 0, nSize.Width, nSize.Height)

  ' Create the barcode image (BMP) and write to stream
  TBarCode.DrawBitmap(nSize.Width, nSize.Height).Save(stream,
                                                      System.Drawing.Imaging.ImageFormat.Bmp)

  TBarCode.IsTextVisible = False

  ' Set the stream position to the beginning of the stream.
  stream.Seek(0, System.IO.SeekOrigin.Begin)

  ' Read all Bytes from the stream and store in Byte array.
  byteArray = New Byte(CType(stream.Length, Integer)) {}
  stream.Read(byteArray, 0, stream.Length)

  Return byteArray
End Function
```

8 Contact and Support Information

TEC-IT Datenverarbeitung GmbH

Address: Hans-Wagnerstr. 6
AT-4400 Steyr
Austria/Europe
Phone: +43 / (0)7252 / 72 72 0
Fax: +43 / (0)7252 / 72 72 0 – 77
Email: office@tec-it.com
Web: http://www.tec-it.com