dLSoft

Barcode Tools for Crystal Reports

By dLSoft

This manual was produced using *ComponentOne Doc-To-Help.*™

Contents

Barcode Tools for Crystal Reports 1

Barcode Tools for Crystal Reports

Introduction

dLSoft Barcode Tools for Crystal Reports provides the tools needed to include most common barcodes within Crystal Reports XI or later, either as graphic images or as font-based barcodes. Font-based barcodes may also be included in reports created using earlier versions of Crystal Reports, and barcode graphic images may be included in packages created using the Crystal Reports Designer Component for VB6 included with Crystal Reports 8.5 and 9.

It is important to understand the fundamental difference between the use of the tools with Crystal Reports XI compared with earlier version. Distributing or remote viewing of barcode-containing reports based on versions of Crystal Reports earlier than version XI require that one or more of the barcode tools be available on the machine viewing the report, whereas with version XI barcode images may be embedded in the report and so do not require the presence of one of the barcode tools on the machine opening the report.

Licensing

Barcode Tools for Crystal Reports may be licensed as a single computer version for a single PC running Windows XP/Vista/Windows 7/8, or as a Server/multi-user version. Licences covers the supplied 32/64 bit servlet for IIS 6 or later, the 32 bit OLE sever, the CRUFLs (and older versions) and fonts.

The following licence types are available:

Barcode Tools for Crystal Reports - Full version (supports 1D and 2D barcodes)

Barcode Tools for Crystal Reports - 1D version (supports only 1D barcodes)

Barcode Tools for Crystal Reports - 2D version (supports only 2D barcodes)

Installation

Installation of the Barcode Tools for Crystal Reports package will install a number of files and folders in a folder of your choice – by default this will be Documents\BarcodesForCrystal. The following items will be created in this folder:

1. A product Help file, manual and a README.TXT file, which may contain information that was not available when this Help system was created. A general Barcodes Help file is also included.
2. Two dummy barcode images that should be used when graphic placeholders are created in reports
3. One or more sample directories containing example reports suitable for the versions of Crystal Reports selected during installation

4. A folder name dlbc which contains the barcode image servlet for use with IIS and Crystal Reports XI or later. For single PC licensed versions this folder may be copied to the localhost IIS website folder (normally inetpub). For developer/server licensed version the folder may be copied to the relevant website folder on the IIS machine.
5. A folder name dlbc2d which contains the 2D barcode image servlet for use with IIS and Crystal Reports XI or later. This folder is not installed for the Barcodes for Crystal Reports 1D version.
6. A folder named CRUFL that contains an example of a cmd file for installing the CRUFL_DL1D.dll and CRUFL_DL2D User Function Libraries (UFL) on machines other than that on which the installation has been run (see Appendix). The CRUFL_DL2D library is not installed for the Barcodes for Crystal Reports 1D version.
7. A folder named fonts that contains the fonts used for creating font-based barcodes. These fonts are also registered (with the operating system) during installation.

Installation will also place the older UFLs (U25UNIF.DLL for 1D barcodes, and U25U2D.DLL for 2D barcodes in the full versions of Barcodes for Crystal Reports only) in the Windows System32 directory (or SysWOW64 on 64 bit systems). These UFLs are provided for compatibility with earlier releases, although we recommend users to install the more modern CRUFL if using Crystal Reports 2008 or later because it permits a wider range of compatibility.

Both UFLs and CRUFLs provide the facility to create font-based barcodes with reports using a User-Defined Formula.

Note: Font-based barcodes produced using trial versions of the tools will create barcodes in which the data is randomly scrambled, and in the case of GS1 barcodes, bars of varying heights. Picture barcodes produced using trial versions of the tools will be displayed with a portion missing.

Additional steps for Picture barcodes in Crystal Reports XI or later

Although you will be able to open the sample reports containing barcode images using Crystal Report XI (or later), you will not be able to add or update barcodes on reports until the barcode server has been created. The following steps should be followed in the order given to ensure that this one-time process is painless. Note that to setup for both 1D and 2D barcodes with the full version of Barcodes for Crystal Reports, the steps referring to the dlbc folder should also be followed for the dlbc2d folder.

Step 1.

Ensure that the Microsoft Internet Information Service is installed and available.

Note: The Barcode servlet does not need to receive data from outside your computer, so unless you need to use Internet Information Service for other purposes you should ensure that access to this service from outside your computer is not permitted by your firewall.

Windows XP/2003: You can check for the presence of Internet Information Services on Windows XP or Windows Server 2003 by opening the Administrative Tools item on the Control Panel. If it contains an icon for Internet Information Services then the service is installed and you can proceed to step 2.

If the service is not installed you need to install it from the Windows XP CD. Choose the Add/Remove programs item in the Control Panel. In the dialog box displayed select the Add/Remove Windows Components icon and check the box alongside the entry labeled Internet Information Services (IIS). Then push the Next button and follow the instructions.

Windows Vista/7/8/10: You can check for the presence of Internet Information Services on Windows Vista/7/8/10 by opening the Control Panel (Classic View) and selecting the Programs and Features icon. Double-click the icon and select the Turn Windows Features On or Off icon. In the dialog box displayed locate Internet Information Services. If the check box alongside is checked then the Internet Information Service is installed and you can proceed to step 2.

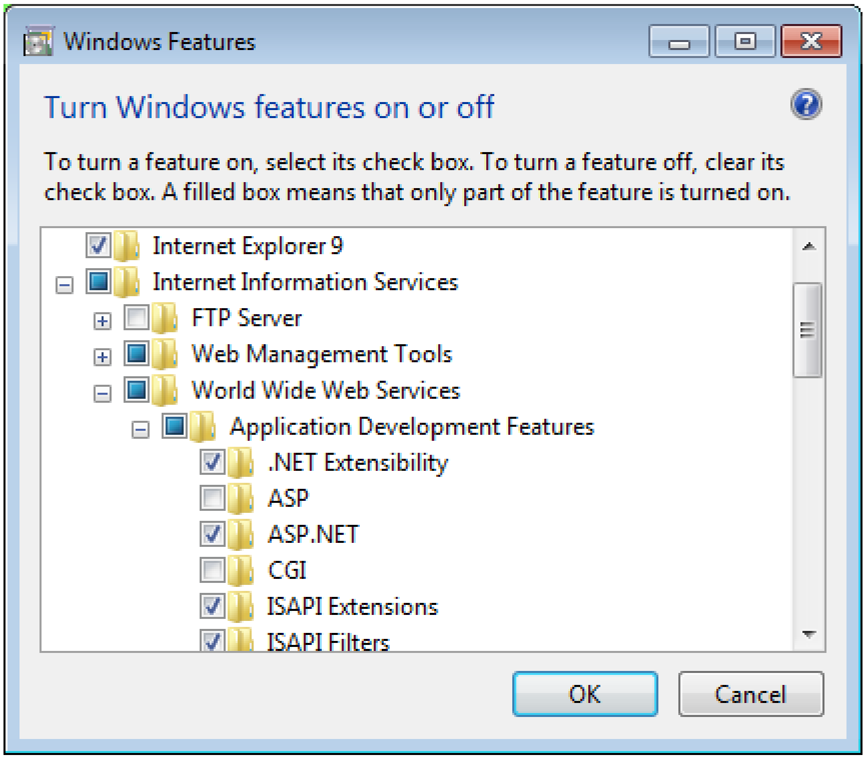
If the service is not installed you need to install as follows:

Expand the item Internet Information Services. Expand Web management tools and check the boxes for:

IIS Management console
IIS Management scripts and tools
IIS Management service

Expand World Wide Web services. Expand Application Development Features and check the box for:

ASP.NET (others will be added automatically).



Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☑ Internet Explorer 9
- ⊟ ■ Internet Information Services
  - ⊞ ☐ FTP Server
  - ⊞ ■ Web Management Tools
  - ⊟ ■ World Wide Web Services
    - ⊟ ■ Application Development Features
      - ☑ .NET Extensibility
      - ☐ ASP
      - ☑ ASP.NET
      - ☐ CGI
      - ☑ ISAPI Extensions
      - ☑ ISAPI Filters

Windows Server 2008: You can check for the presence of IIS on Server 2008 system by opening the Server Manage and selecting the Roles member. If Web Server (IIS) appear in the Roles list then the service is installed. Otherwise it can be added by selecting the Add Roles link and following the instructions.

If you install Internet Information Services be sure to run Windows Update when installation is complete to check for any security updates for the service.

Step 2.

Ensure that the Microsoft .NET Framework version 2.0 or later is installed.

Note: Version 2.0 or later of the Framework is required only to enable IIS to run Barcode image server's ASP.NET page. Its presence will not interfere with older versions of the Framework which may be installed. Windows Vista/7 normally has this version installed by default.

You can check for the presence of the .NET Framework using Windows Explorer. Open the folder C:\Windows\Microsoft.NET\Framework If a folder with the name v2.0.50727 or a higher version is present the then required version is available and you can proceed to step 3. On 64 bit systems there may also be a C:\Windows\Microsoft.NET\Framework64 folder. The barcode image server may be used with either 32 or 64 bit frameworks.

If version 2 or later of the Framework is not installed then it may be installed by running Windows Update and selecting the Microsoft.NET Framework from the list of Optional or Custom software updates available.

Step3.

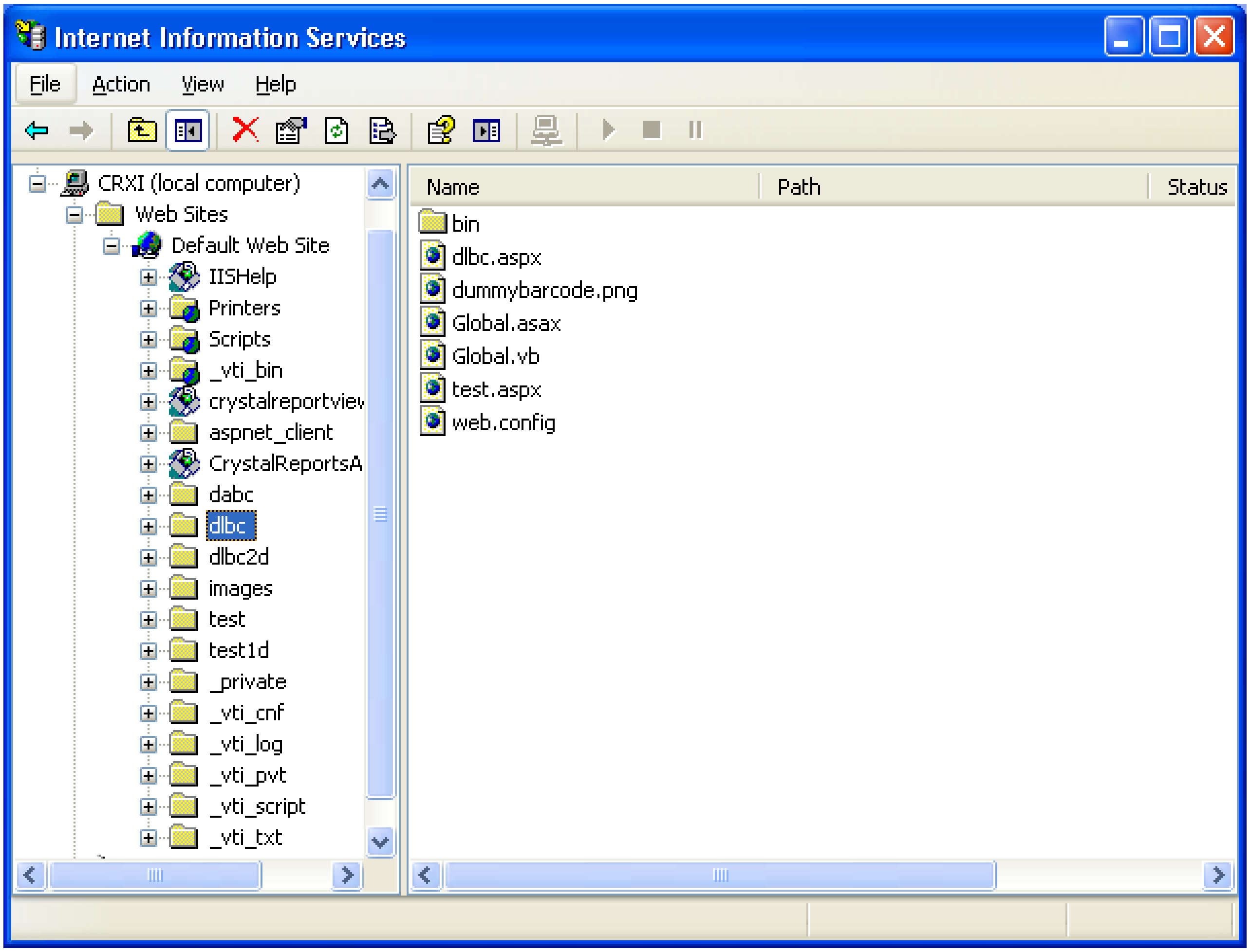
If you have installed Internet Information Services on a Windows XP machine that already had the .NET Framework version 2.0 installed you should “Refresh” the .NET installation as shown below, otherwise proceed to Step 4. Choose the Add/Remove programs item in the Control Panel. In the dialog box displayed select the Microsoft .NET Framework 2.0 item and push the Change/Remove button. Select Repair in the Setup dialog and push the Next button.

Step 4

Now *copy* the folder DLBC from the Barcodes for Crystal Reports installation directory to the directory C:\Inetpub\wwwroot. The folder C:\Inetpub\wwwroot\dlbc will now contain the ASP.NET code that your barcodes in Crystal Reports will use to generate their graphic images.

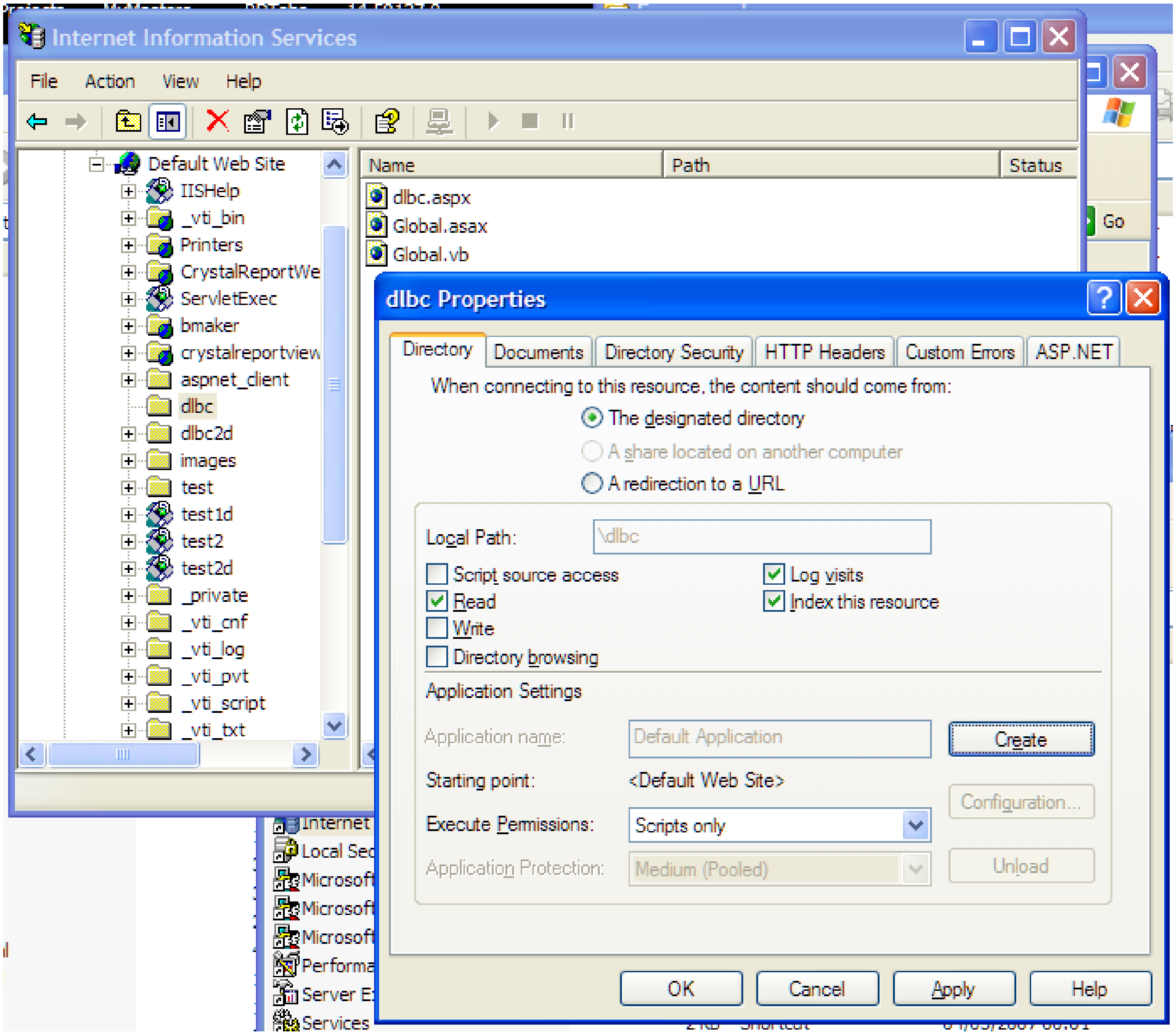
Step 5 for Windows XP/2003

To make the ASP.NET code available on your computer open the Administrative Tools icon on the Control Panel and double-click on the Internet Information Services icon. Expand the tree by the name of your computer, expand the Web Sites entry, and expand the Default Web Site entry. You should see a folder icon labeled dlbc in the list.



Name	Path	Status
bin		
dlbc.aspx		
dummybarcode.png		
Global.asax		
Global.vb		
test.aspx		
web.config		

Select the dlbc folder, then right-click on it and choose Properties from the pop-up menu displayed.



When connecting to this resource, the content should come from:

- ● The designated directory
- ○ A share located on another computer
- ○ A redirection to a URL

Local Path: \dlbc

- □ Script source access
- ☑ Read
- □ Write
- □ Directory browsing
- ☑ Log visits
- ☑ Index this resource

Application Settings

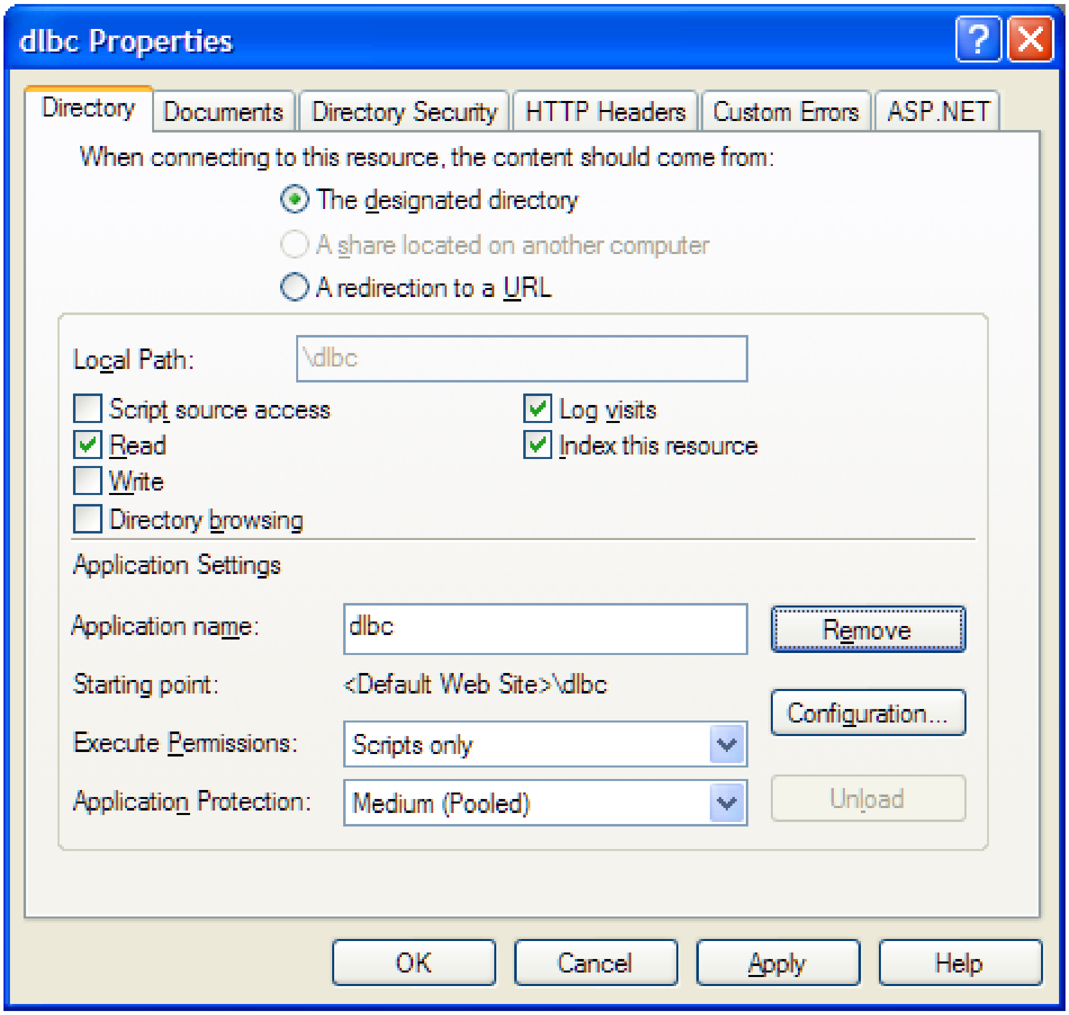
Application name: Default Application

Starting point: <Default Web Site>

Execute Permissions: Scripts only

Application Protection: Medium (Pooled)

Push the Create button to register the dlbc folder as a program.



When connecting to this resource, the content should come from:

● The designated directory

○ A share located on another computer

○ A redirection to a URL

Local Path: \dlbc

□ Script source access

☑ Read

□ Write

□ Directory browsing

☑ Log visits

☑ Index this resource

Application Settings

Application name: dlbc

Starting point: <Default Web Site>\dlbc

Execute Permissions: Scripts only

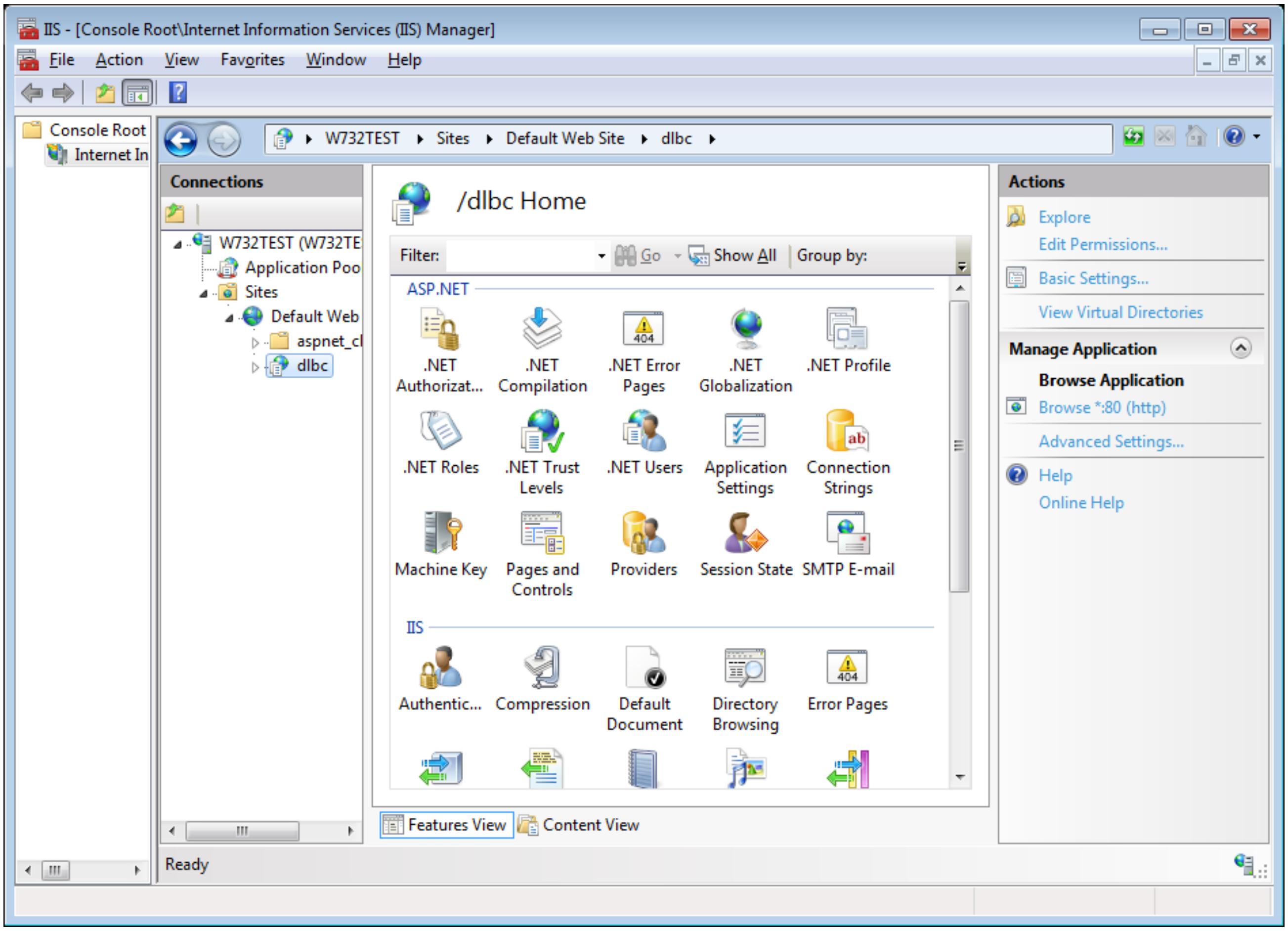
Application Protection: Medium (Pooled)

Then push the OK button and close the IIS window and the control panel.

Step 5 for Windows 7/8/10

The IIS Manage may be opened from the Control Panel by selecting System and Security, then Administrative Tools, and double-clicking on Internet Information Services (IIS) Manager

Select the arrow alongside the computer name to expand the Connections list.



Select Sites and expand the item. Select Default Web Site and expand it.

Select the dlbc item. Right-click on the dlbc item and select: Convert to application.

Select the Default Site item. Look in right hand column in the Manage Web Site section to ensure that the web site is "Started". If it isn't, then click on Start.

Step 6

Finally, and to check that the image server is functioning, copy the following line into your web browser URL window and press the enter key.

http://localhost/dlbc/test.aspx?data=12345

You should see a (rather large) barcode in your browser window.

1D Picture barcodes in CR XI or later

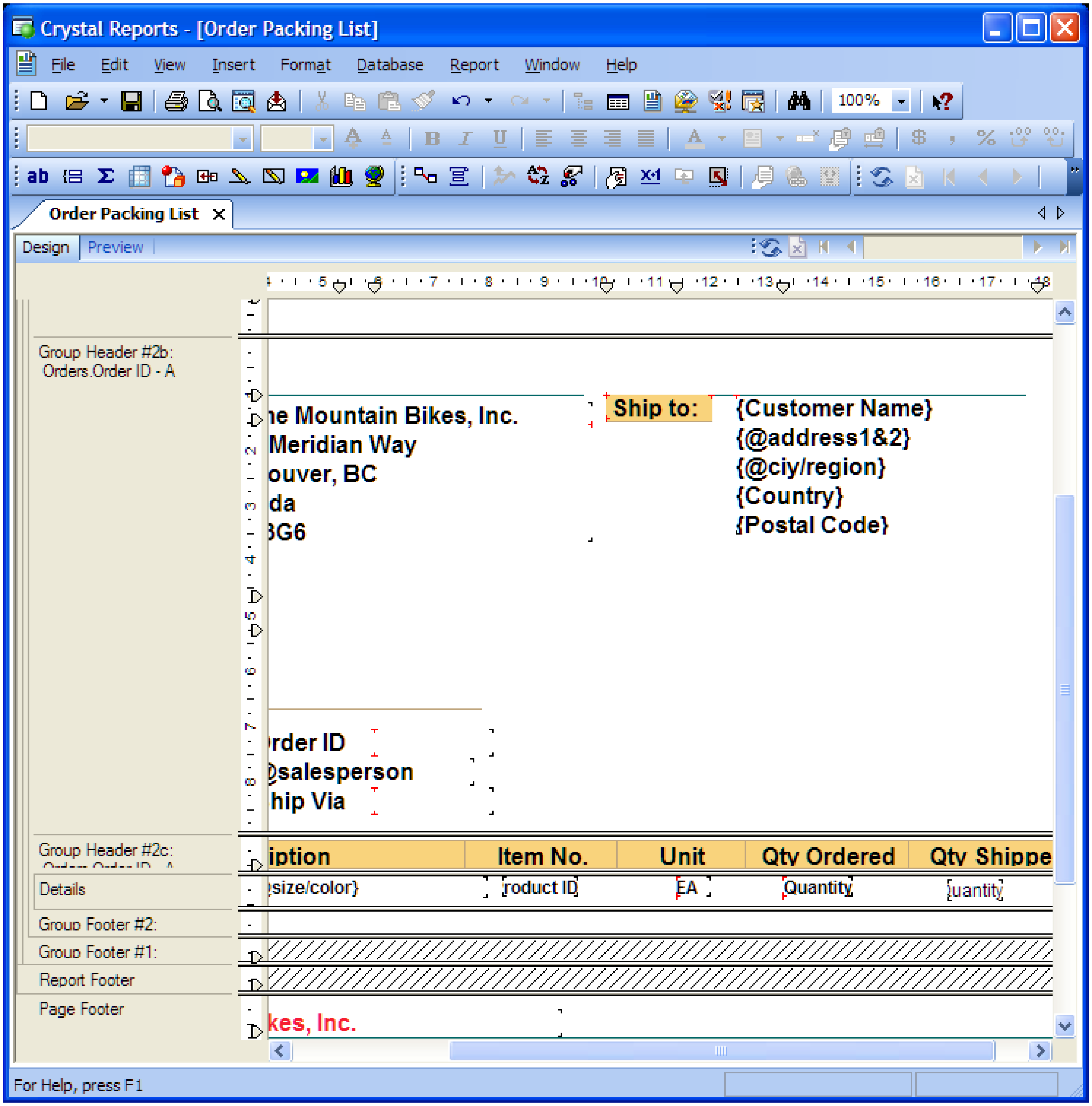
Picture barcodes may be embedded into reports created with Crystal Reports XI or later. Picture barcodes are Windows Metafile (wmf) images by default. Details on selecting alternative image types are in the IIS servlet section.

Creating embedded 1D barcode images

To place embedded barcode images into an existing report follow these steps.

Step 1.

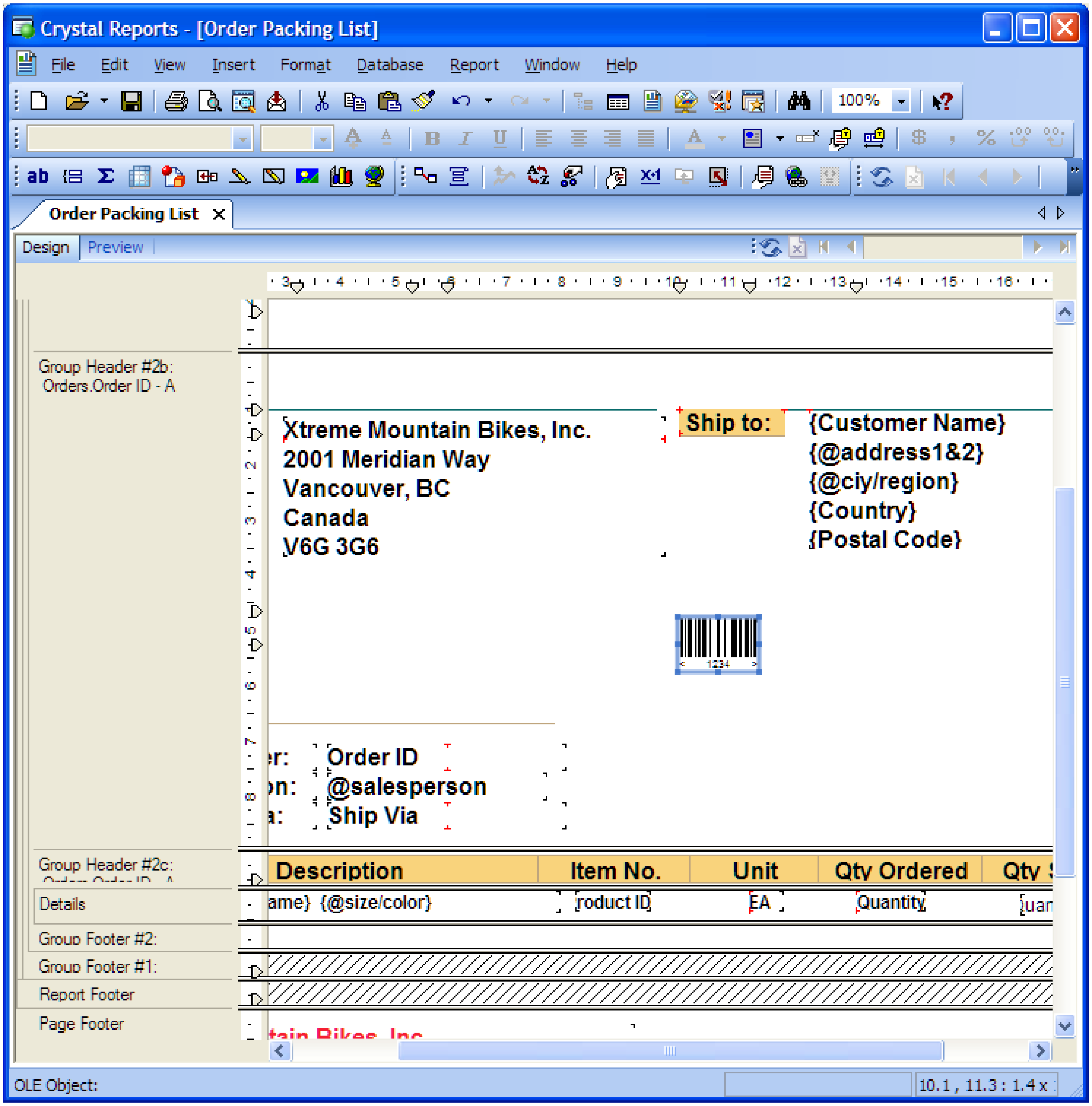
Open the report to which you wish to add barcodes



Step 2.

From the edit menu choose Insert and then choose Picture. Navigate to the Barcodes for Crystal Reports installation directory and choose one of the dummy barcode image files, then push the Open button.

A movable rectangle appears. Position this at the point at which you wish the barcode to appear then click the left mouse button. Do not change the size of the rectangle. The rectangle containing a dummy barcode image appears on the report.



Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to: {Customer Name}
{@address1&2}
{@ciy/region}
{Country}
{Postal Code}

1234

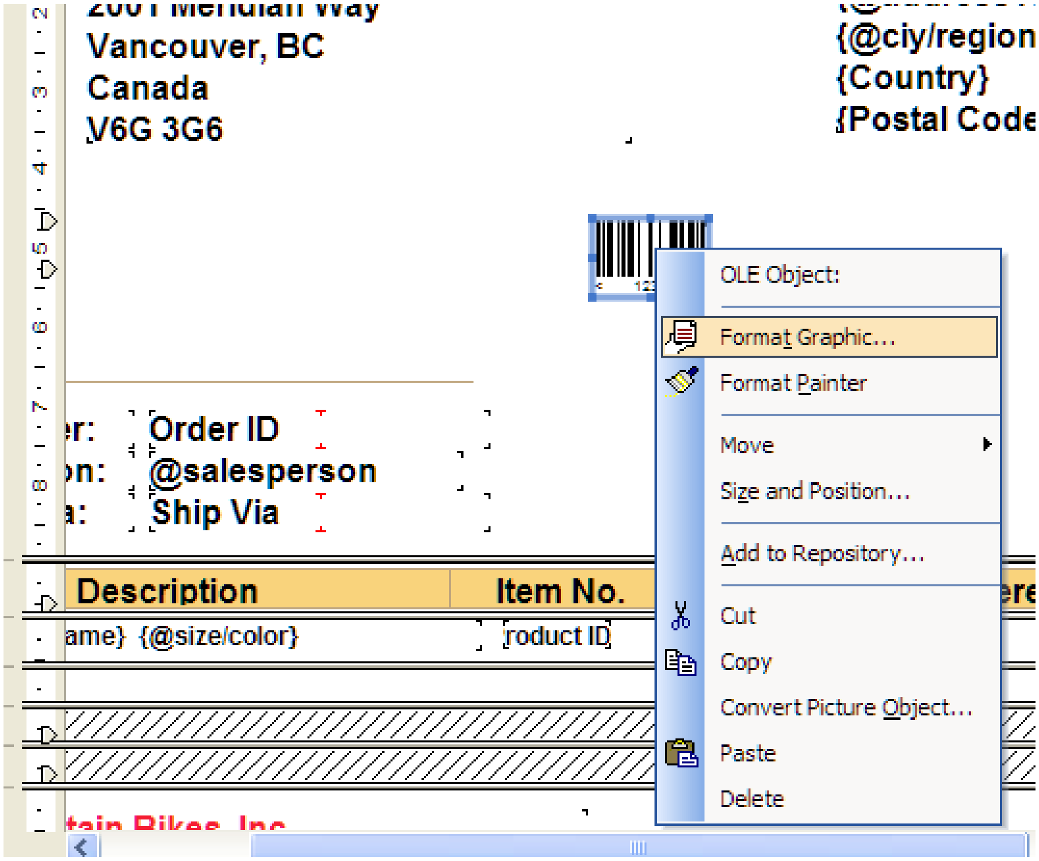
r: Order ID
on: @salesperson
a: Ship Via

Description	Item No.	Unit	Qty Ordered	Qty S
ame} {@size/color}	roduct ID	EA	Quantity	uan

tain Bikes, Inc

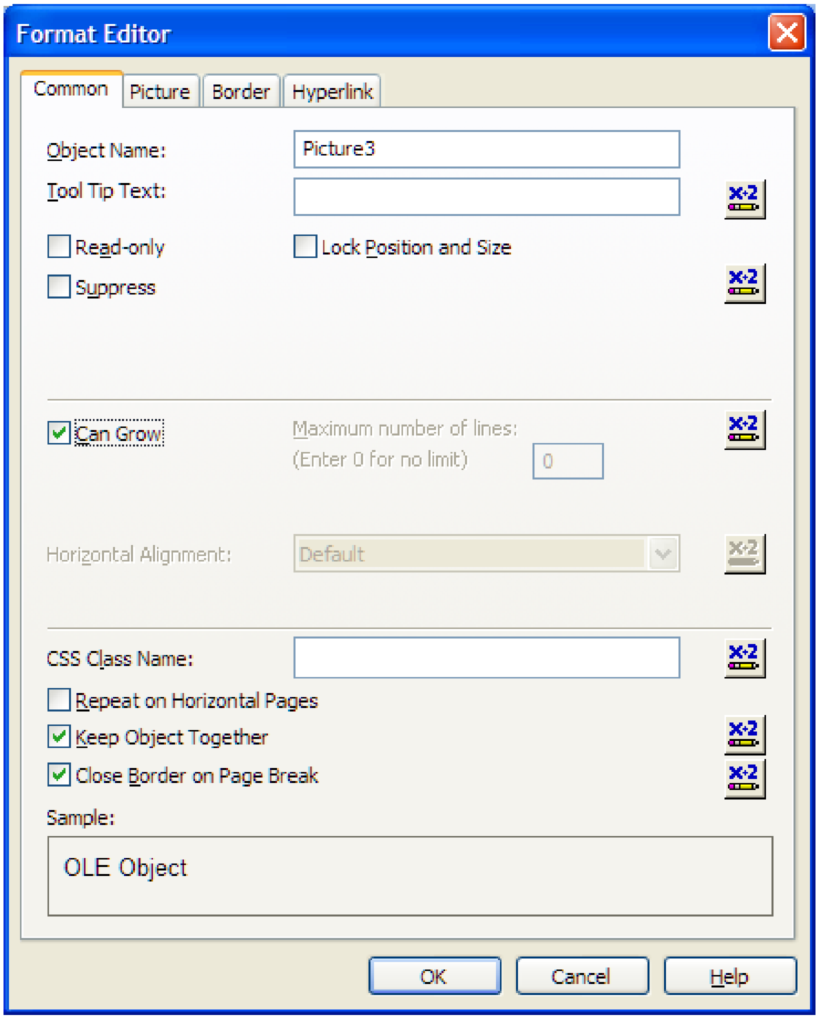
Step 3.

Right-click on the selected barcode image and choose Format Graphic from the pop-up menu that appears.



Step 4.

In the Format editor dialog that appears ensure that the Can Grow check box is checked then select the Picture tab.



Object Name: Picture3

Tool Tip Text:

□ Read-only

□ Lock Position and Size

□ Suppress

☑ Can Grow

Maximum number of lines: (Enter 0 for no limit) 0

Horizontal Alignment: Default

CSS Class Name:

□ Repeat on Horizontal Pages

☑ Keep Object Together

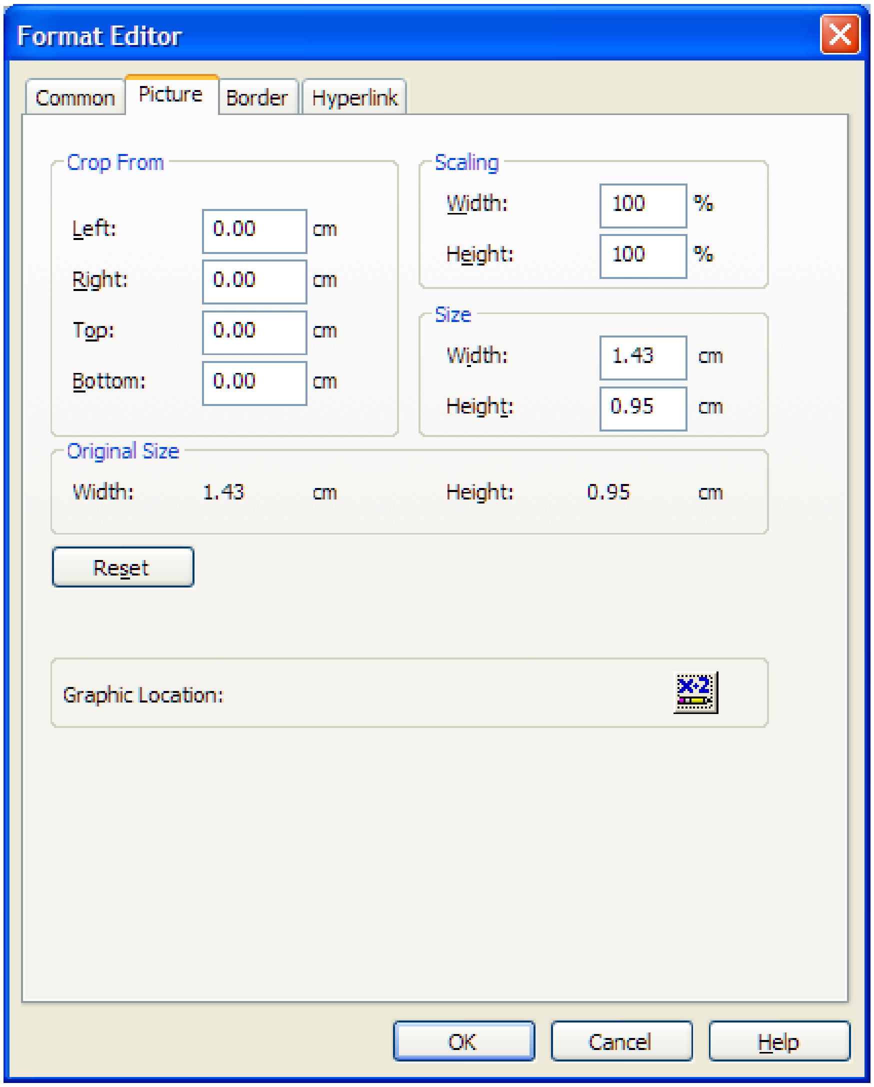
☑ Close Border on Page Break

Sample:

OLE Object

Step 5.

Push the icon that appears alongside the Graphic Location label



Crop From

Left: 0.00 cm

Right: 0.00 cm

Top: 0.00 cm

Bottom: 0.00 cm

Scaling

Width: 100 %

Height: 100 %

Size

Width: 1.43 cm

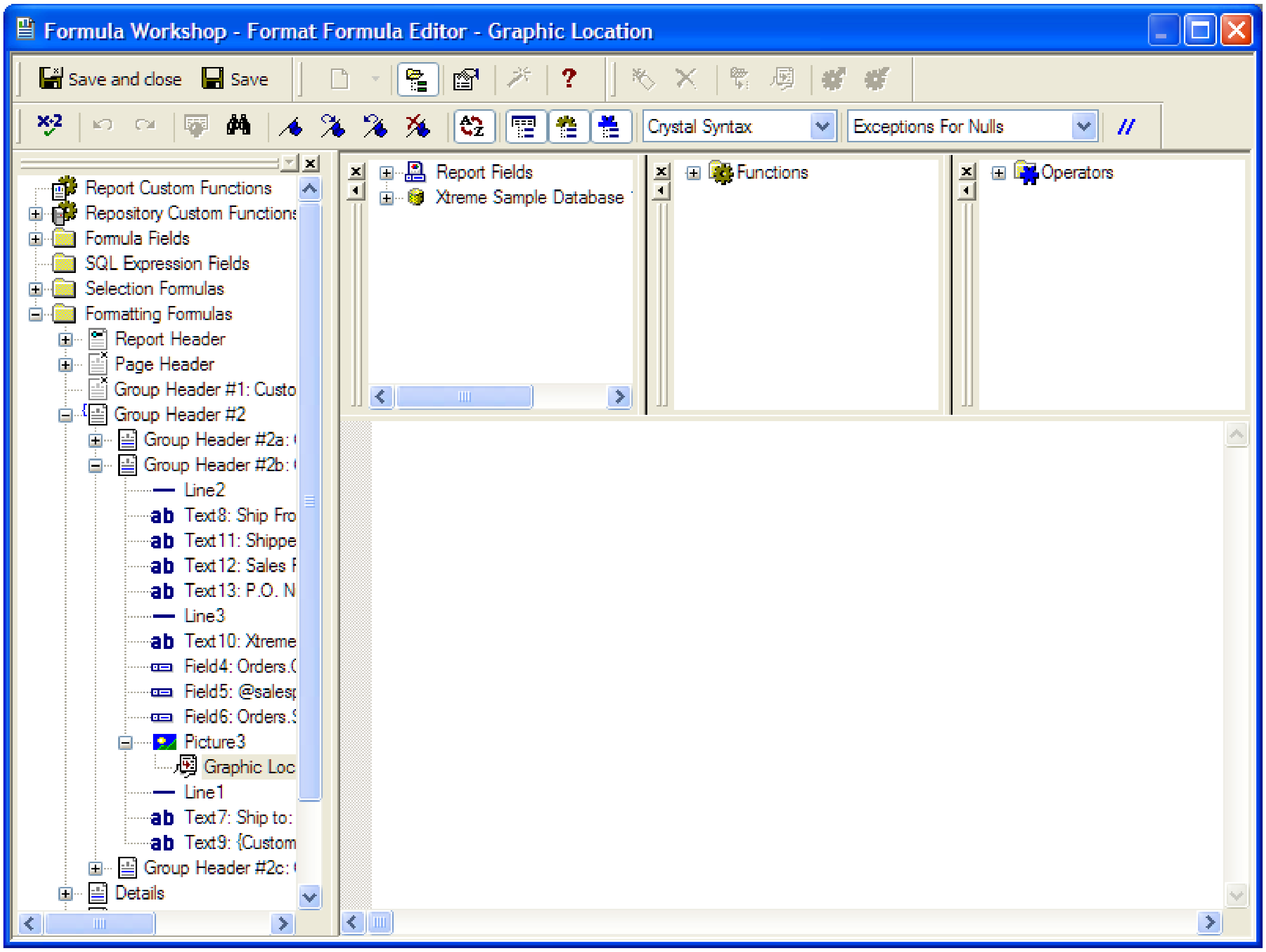
Height: 0.95 cm

Original Size

Width: 1.43 cm Height: 0.95 cm

Step 6.

The Formula Workshop dialog appears. If the Graphic Location item for your picture is not already selected, expand the entries under Formatting Formulas and select the Graphic Location item.



Step 7.

In the formula space enter the first part of the formula, such as

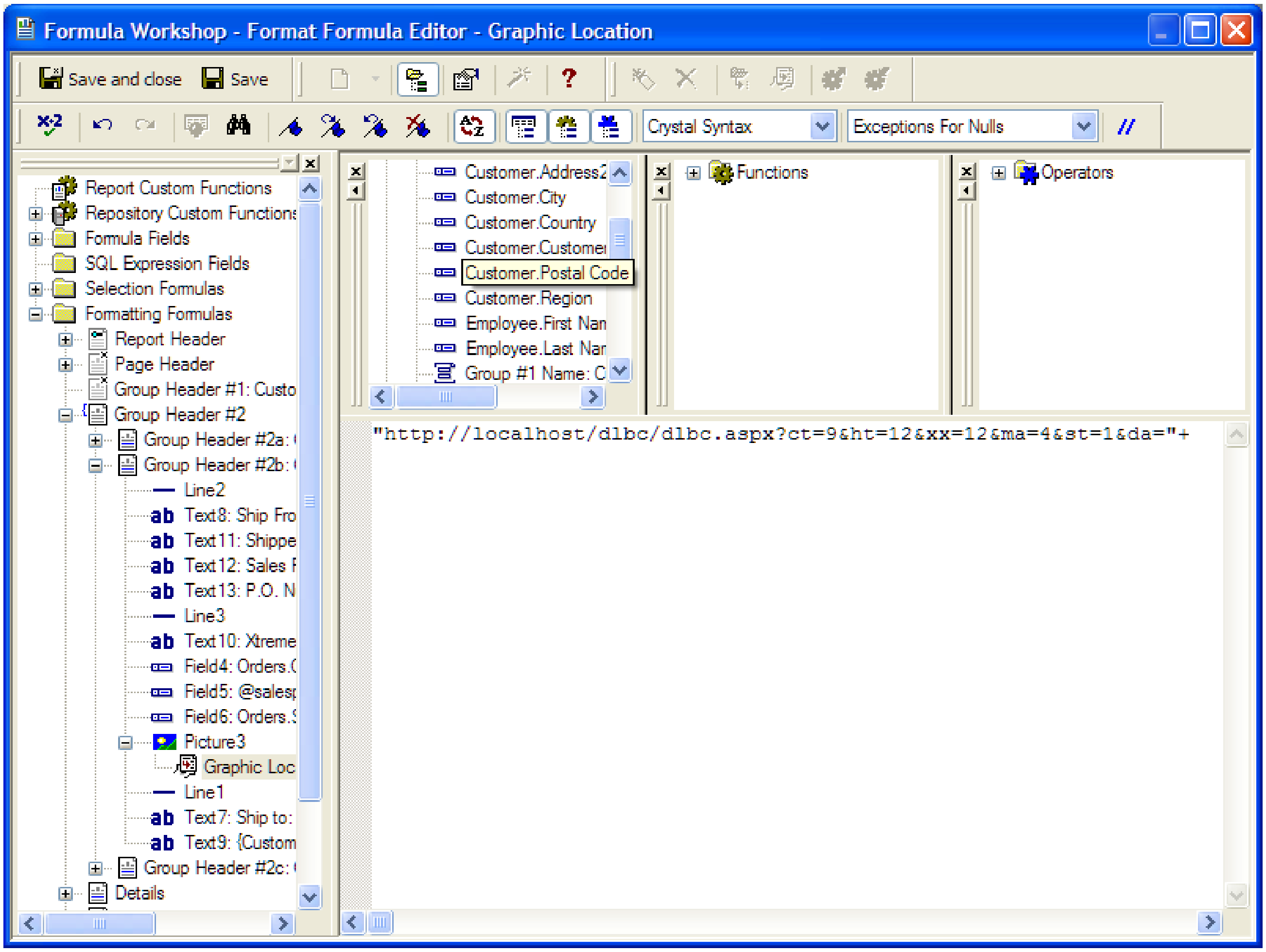
"http://localhost/dlbc/dlbc.aspx?CodeType=9&ImageHeight=12&Xunit=12&MarginSize=4&Data="+

including the quotation marks and the + sign at the end. Click once just after the + sign to ensure that the insertion point cursor is flashing.

Details of the available parameters are given in the Reference section.

The last item within the quotation marks is "Data=" which is pointing to the data that will be used to create the barcode.

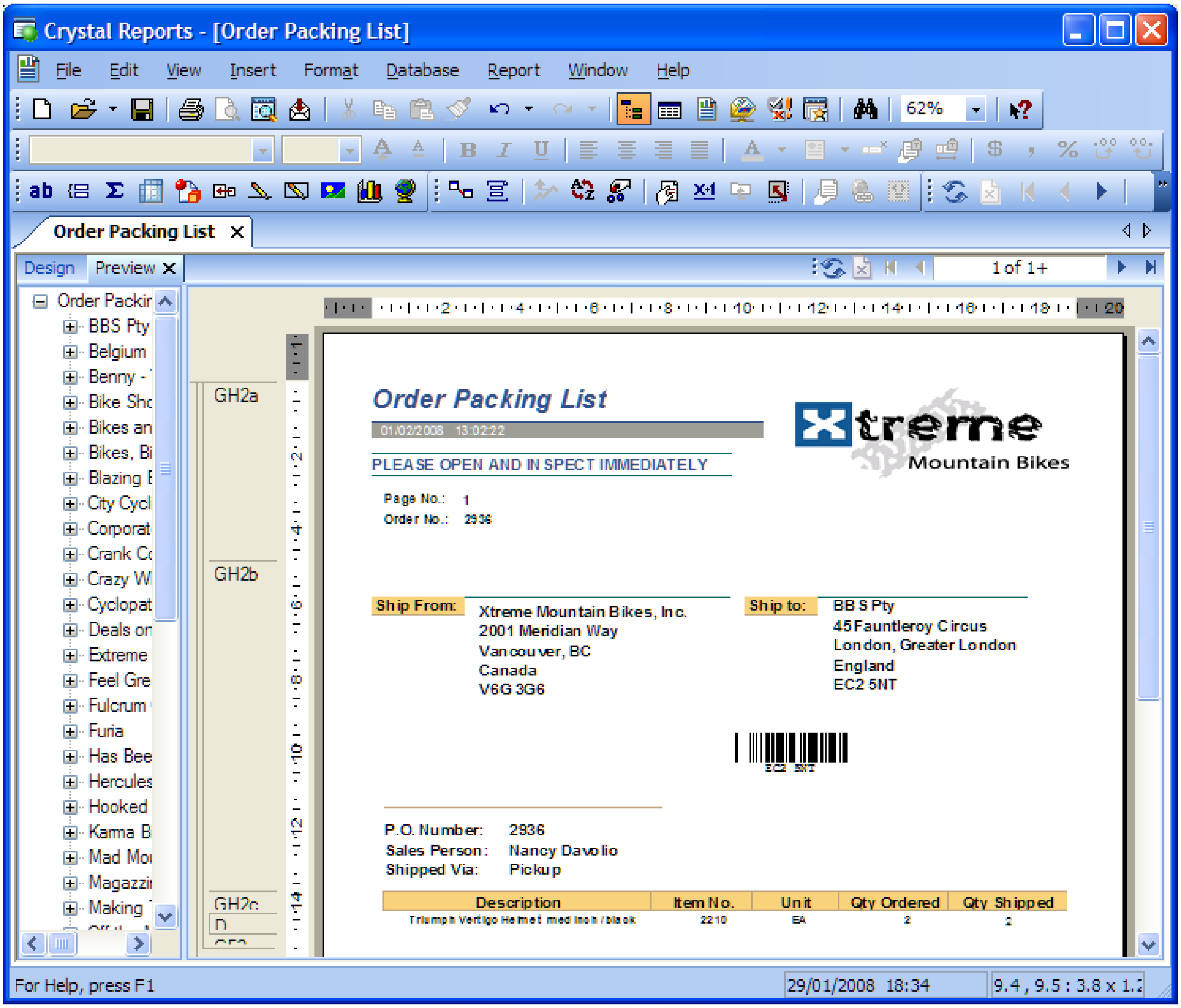
Expand the Report Fields tree and locate the field you wish to use as the data source (the Customer.Postal Code field in this example). Double-click on the field name and the name (complete with braces) will be copied to the end of the formula.



```
"http://localhost/dlbc/dlbc.aspx?ct=9&ht=12&xx=12&ma=4&st=1&da="+
```

Step 8.

Push the Save and Close button, then push OK button in the Format Editor to return to the main window in Design view. Switch to Preview and the report complete with barcodes will be visible



Order Packing List

01/02/2008 13:02:22

PLEASE OPEN AND INSPECT IMMEDIATELY

Page No.: 1
Order No.: 2936

Ship From: Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to: BBS Pty
45 Fauntleroy Circus
London, Greater London
England
EC2 5NT

EC2 5NT

P.O. Number: 2936
Sales Person: Nancy Davolio
Shipped Via: Pickup

Description	Item No.	Unit	Qty Ordered	Qty Shipped
Triumph Vertigo Helmet med Inch / black	2210	EA	2	2

If you Save the report complete with data then the barcode images become embedded within the report. If the data changes the report should be refreshed by selecting Refresh Report Data from the Report menu.

Creating the 1D formula

While every requirement will need a different formula there are some general points that should be taken into account when creating the formula to generate the barcode symbols. Every formula will need to specify a CodeType and we recommend the use of the Xunit parameter to specify the basic element width for the symbol, so every symbol will start with code of the form:

"http://localhost/dlbc/dlbc.aspx?CodeType=9&Xunit=16

The size of 1D barcodes may be specified in units of your choice (the default is millimeters) and the height of the barcode is specified using the ImageHeight property, so code of the form

&ImageHeight=16

is needed

(the ImageWidth parameter may be used as an alternative to specifying the Xunit value. Xunit values are provided in Mils – 1/1000 inches, while ImageHeight and ImageWidth may be in units of your choice. Note that either Xunit or ImageWidth must be specified.)

For barcodes that will include human readable text under the bars the font used to create such text may be specified using the FontName and FontSize properties – otherwise the default of 10 pt Arial will be used.

Many other optional parameters are available and are listed in the Reference section.

Finally every formula needs to end with code that specifies the Data, using code of the form &Data=12345 or, much more likely, &Data= followed by the source of the data, for example

http://localhost/dlbc/dlbc.aspx?CodeType=9&Xunit=12&ImageHeight=16& Data=+{fieldname}

for a Code 128 symbol with an Xunit value of 12 Mils, a bar height of 16 mm, and the data taken from a field name fieldname.

Using Crystal Reports Export function

The Crystal Reports Export to PDF function does not handle vector graphics - which is the default format in which barcode images are created. So if you need to Export reports (rather than Print to a PDF capable driver - such as Adobe Acrobat) it is necessary to modify the edit the servlet file to generate PNG (Portable Network Graphic) images of the barcodes. For full details see Pic() methods in the IIS 1D Server reference below.

To do this edit the file dlbc.aspx (eg using Notepad) and change the line

Session("ms")=m_barcode.Pic(0)

to

Session("ms")=m_barcode.Pic(300)

The 300 represents the graphics resolution you require. Here 300 dpi is used as an example, but if you intend the items to be printed on a printer with a higher resolution (eg 600 dpi) you could use that value. Note that the size of the image will change with resolution as Barcodes for Crystal Reports adjust the bar size to ensure that each bar in an integer number of dots.

IIS 1D Servlet reference

The IIS servlet is used for Crystal Reports XI and later. The servlet consists of an ASPX file (dlbc.aspx) and an associated library.

The image created by the servlet may be obtained within Crystal Reports by setting the source of a Picture element to the servlet's Pic ires() method.

Pic() methods

The pic() methods return a graphic image. These methods may be used by the APS.NET routine for loading an image into a CR XI or later report.

Note: The physical size of a Portable Network Graphic image is determined from the number or pixels per row (or column) and the resolution in pixels per inch. The PNG graphics generated contain the pHYs data block (which contains the size relevant information), but although this is supported by Crystal Reports, not all programs support this and so may display the image at a different size by assuming that there are 96 pixels per inch.

Pic()

returns a png format image using the current resolution setting as set by the DefResolution parameter.

Pic1()

returns a monochrome (1 bpp) png format image using the current resolution setting as set by the DefResolution parameter.

Pic(ires)

Where ires is an integer representing the resolution in dpi of the required image or 0.

If ires is 0 then the image is returned as a Windows Metafile, which is the ideal method for Crystal Reports running on Windows as Metafile images are small vector images.

If ires is greater than 90 then the image is returned as a Portable Network Graphic image with a pixel resolution specified by the value of ires. Typical value for ires would be 300 or 600 (in dpi), and would normally be chosen to suit the printer on which the barcodes will be printed. Lower values are not recommended as these will not generate bar widths that are sufficiently accurate, and higher values generate images that require large amounts of memory.

Pic1(ires)

As Pic(ires) except that the image returned when ires > 90 is a monochrome (1 bpp) Portable Network Graphic image.

Pic(type, imono)

Where type is a string representing the required graphics type, which may be any of:

"bmp" = ImageFormat.Bmp

"gif" = ImageFormat.Gif

"jpg" = ImageFormat.Jpeg

"png" = ImageFormat.Png

"tif" = ImageFormat.Tiff

"wmf" = ImageFormat.Wmf

"eps" = Encapsulated Postscript format

If imono is 0 then the image is returned as a 24 bpp colour graphic for bitmap formats, while if imono is not 0 a monochrome (1 bpp) image is returned.

This function returns an image with the resolution specified by the DefResolution parameter.

Summary of Properties

The image created depends on a number of Properties summarized below and described in more detail in the section following.

The properties that can be set in dlbc.aspx:

AutoCheckdigit specifies whether any check digits are calculated automatically (same as AutoCheck in the http section)

BackColor the colour behind the bars

BarRatio specifies the ratio of wide/narrow bar for some barcode types

BarcodeHeight required target height of barcode image (in units determined by the setting of the Unit property)

BarcodeWidth required target width of barcode image (in units determined by the setting of the Unit property)

BearerSize specifies the thickness of bearer bars for those barcodes that may have bearer bars

BorderWidth specifies the thickness of any border around the image

BothBearers determines whether both upper and lower bearer bars are displayed

Caption specifies the characters that make up the code (same as Data in the http section)

CodeType specifies the barcode type required

DefResolution Sets the default resolution for barcode images.

ExtendBearers Allows bearer bars to extend into light margins.

ForeColor the colour of the bars and any text under the bars

Font The font used to render any text under the barcode

Indicators specifies whether light margin indicators should be displayed for those barcode types that support these. Not that the prefix and suffix digits of UPC barcodes and the prefix digit of EAN-13 and EAN-8 barcodes are regarded as light margin indicators for this purpose.

MarginSize specifies the size of the light margins

Orientation specifies the orientation angle of the barcode image

Mirror specifies that a mirror image of the symbol is created

Reduction specifies the percentage reduction in bar thickness (useful for allowing for ink spread in wet-ink printing processes.

ShowText specifies that the text content of the barcode should be displayed under the bars

ShowCheckdigit specifies whether any automatic check digit is displayed (for those barcode type which permit this)

TextAlign specifies the text justification for human readable text under the barcode

Xunit specifies the thickness of each barcode element in mils (1/1000 inches)

Properties that can be set in the http call to the server:

AutoCheck specifies whether any check digits are calculated automatically

BarcodeHeight required target height of barcode image (in units determined by the setting of Units)

BarcodeWidth required target width of barcode image (in units determined by the setting of Units)

BarRatio specifies the ratio of wide/narrow bar (times ten) for some barcode types

BarReduce specifies the percentage reduction in bar thickness (useful for allowing for ink spread in wet-ink printing processes.

BearerSize specifies the thickness of bearer bars for those barcodes that may have bearer bars

BorderSize specifies the thickness of any border around the image (in points)

BothBearers determines whether both upper and lower bearer bars are displayed

CodeType specifies the barcode type required

Data the text to be encoded in the barcode

ExtendBearers Allows bearer bars to extend into light margins.

FontSize, FontName specifies the font for human readable text

FontBold, FontItalic font characteristics for human readable text

Indicators specifies whether light margin indicators should be displayed alongside human readable text for those barcode types that support these

MarginSize specifies the size of the light margins (in units determined by the setting of Units)

Orientation specifies the orientation of the barcode image

ShowText specifies that the text content of the barcode should be displayed under the bars

ShowCheck specifies whether any automatic check digit is displayed (for those barcode type which permit this)

Unit specifies the units of ImageHeight, ImageWidth, BorderSize and MarginSize parameters; default is mm (millimeter), also allowed is in (inch)

Xunit specifies the thickness of each barcode element in Mils (1/1000 inch)

Many of these properties have default values (see the reference section), so do not require changing if you can use the default values. The properties that must be set to obtain a barcode are

CodeType specifies the barcode type required

Data specifies the characters that make up the code

Barcode Types Table for the 1D IIS servlet

The barcode type can be set using either the CodeType property or the CodeTypeValue property

Barcode	**CodeTypeValue**
Code 39	0
Extended Code 39	1
Codabar	2
2 of 5	3
Interleaved 2 of 5	4
Matrix 2 of 5	5
Code 93	6
Extended Code 93	7
Code 128	8
GS1-128	9
SSCC	10
EAN 14	11
EAN 13	12
EAN 13 + 2 digits	13
EAN 13 + 5 digits	14
EAN 8	15
EAN 8 + 2 digits	16
EAN 8 + 5 digits	17
UPC-A	18
UPC-A + 2 digits	19
UPC-A + 5 digits	20
UPC-E	21
MSI/Plessey	22
Plessey	23
DeutschenPost	24
PostNet	25
Planet 12	26

Planet 14	27
Royal Mail	28
4-State	29
ISBN	30
ISSN	31
ISMN	32
TelePen Standard	33
TelePen ASCII	34
TelePen Numeric	35
Japan Postal code	36
Code 11	37
Code B	38
ITF-6	39
ITF-14	40
IATA 2 of 5	41
China Postal Code	42
OneCode	43
Code 128 Subtype A	44
Code 128 Subtype B	45
UPC E0	46
UPC E1	47
Korean Postal Authori	48
Italian Postal 2/5	49
Italian Postal 3/9	50
Australia Post	51
InfoMail Barcode A	52
PZN	53
Databar Omnidirection	54
Databar Truncated	55
Databar Limited	56
Databar Expanded	57
HIBC LIC Code 128	58
HIBC LIC Code 39	59
IM^{TM} Package Barcode	60

1D Properties

AutoCheckdigit

Type: BOOL

Default: FALSE

Allowed values: FALSE (check digit characters not calculated)

TRUE (check digit characters calculated and appended to code for appropriate code types)

This property is ignored for Code 128, GS1-128, Planet and Postnet codes where the check digit is mandatory and not available in human readable form.

BackColor

Not available via http

Type: Color

Default: Color.White

Allowed values: any allowed Color value.

Sets the colour of the image background. This value may be over-ridden by the Transparent property.

BarcodeHeight

Type: float

Default: 16.0 mm

Sets the required target height of barcode image (in units determined by the setting of the Unit property)

BarcodeWidth

Type: float

Default: 30.0 mm

Sets the required target width of barcode image (in units determined by the setting of the Unit property).

Note: this value is ignored if a non-zero value is provided from the Xunit property.

BarRatio

Type: float

Default 2.5

Allowed values: 2.0 – 3.0

This setting allows some barcode types to have the Wide bar/Narrow bar ratio modified.

Applies mainly to Code 39 and Interleaved 2 of 5 barcodes.

BearerSize

Type: float

Default: 0

Allowed values: 0 - 10000

Allows the thickness of bearer bars for those barcodes which support bearers to be set (in units determined by the setting of the PageUnit property of the Graphics object passed into the component).

Note that for most barcode types the number of bearer bars displayed depends on the setting of the BothBearers property. If BothBearers is True the bars are displayed above and below the barcode. If BothBearers is False then only a single bearer bar is displayed above the barcode.

BorderWidth

Type: float

Default: 0.0 (no border)

Allowed values: any

Specifies the size (in units determined by the setting of the PageUnit property of the Graphics object passed into the component) of a border placed around the barcode. The border area is created using the background color set by the BackColor property.

BothBearers

Type: BOOL

Default: FALSE

Allowed values: TRUE or FALSE

When TRUE causes bearer bars to be draw above and below the barcode. When FALSE no bearer bars are drawn if the BearerSize property is zero. Otherwise a FALSE setting causing only a single bearer bar to be displayed above the barcode.

Caption

Not available via http. The equivalent http parameter is Data.

Type: string

Default: "12345"

Allowed values: Any text string.

Note: only text strings recognized as valid barcodes will result in a barcode picture. An Illegal character in the text string will cause an Error value to be set.

CodeType

Note: When CodeType has the same value and meaning as CodeTypeValue.

CodeTypeValue

Type: integer

Default: 0

Allowed values: The ranges of values defined for the individual components are shown in the barcode type tables above.

Controls

Not available via http

Type: String

Default: “ÅÆÇÈÉÊËÌÍÎÏÐ”

Encoding schemes based on Code 128 may include control characters, such as fnc1. These may be inserted by the user into barcode data and are handled in specific ways during the encoding process. (Each character may be created on a western keyboard by holding down the Alt key and type 0xyz on the numeric keypad, where xyz is one of the three digit codes listed below. The table also shows the function initiated by each character while in each of the Code 128 modes.)

xyz	character	Code A	Code B	Code C
197	Å	DEL		
198	Æ	func. 3	func. 3	
199	Ç	func. 2	func. 2	
200	È	shift	shift	
201	É	code C	code C	

202	Ê	code B	func. 4	code B
203	Ë	func. 4	code A	code A
204	Ì	func. 1	func. 1	func. 1
205	Í	Start A	Start A	Start A
206	Î	Start B	Start B	Start B
207	Ï	Start C	Start C	Start C
208	Ð	NUL		

The Controls property is a string containing the 12 allowed control codes. The programmer has the option to change any or all of these characters for alternative characters by setting the Control property to another string of 12 characters.

The facility to specify control codes is particularly useful for schemes such as GS1-128 where the func 1 character is essential for terminating variable length AIs.

DefResolution

Type: integer

Default: 96

Sets the default resolution for barcode images.

ExtendBearers

Type: BOOL

Default: FALSE

When TRUE allows bearer bars to extend into light margins. When FALSE bearer bars cover the bars only.

Extra1

Type: BOOL

Default FALSE

These additional properties are not normally used. However, they do provide additional functions for a limited number of specific barcode types. See Barcodes section for details.

Extra2

Type: BOOL

Default FALSE

These additional properties are not normally used. However, they do provide additional functions for a limited number of specific barcode types. See Barcodes section for details.

Font

Not available via http

Type: FONT

Default: Arial 10 point

Allowed values: Any accessible TrueType font.

ForeColor

Type: Color

Default: Color.Black

Allowed values: any valid Color

Sets the colour of the image foreground, i.e. the bars and text colour.

Indicators

Type: BOOL

Default: FALSE

A value of TRUE causes the light margin indicators to be displayed. For some EAN barcodes there are recommended ways for the margin indicators to be shown on the image, and the Prefix code and checkdigit for UPC-A and UPC-E codetype to be displayed in the light margins. A value of FALSE prevents the display of the light margin indicators.

MarginSize

Type: FLOAT

Default: 0.0

Allowed values: any

The MarginSize property sets the Light Margin space on either side of a barcode image. The units are the units specified by the current ScaleMode. (in units determined by the setting of the PageUnit property of the Graphics object passed into the component)

Mirror

Not available via http

Type: BOOL

Default: False

A value of True will cause the barcode image to be created as a mirror image. This facility is designed for use with printing technologies that require a mirror image. Mirror image barcodes may not be scannable unless printed correctly.

Orientation

Type: integer

Default: 0 The value of this parameter determines the orientation of the barcode image created.

Allowed values: 0 - 360 degrees

Note that the rotation of text is only supported by TrueType and other rotatable fonts. Note also that some applications do not correctly handle metafiles that contain rotated text.

Reduction

Type: float

Default: 0 the thickness of each line drawn on the barcode image is reduced by this percentage amount. This property may be used to compensate for ink spreading during wet-ink printing.

Allowed values: 0 - 50 (%)

A positive reduces the thickness of bars by the calculated amount of the thinnest bar. A negative value reduces the thickness of bars by the percentage applied to the bar thickness (so a bar three units wide is reduced by an absolute amount that is three times greater than that which would apply to a one unit bar)

ShowCheckdigit

Type: BOOL

Default: FALSE

When set to TRUE this property causes the any automatically calculated check digit to be included in the any text displayed along with the barcode. When set to FALSE the check digit is not displayed.

Note that this property has no effect on those codetypes for which the checkdigit display is mandatory (including EAN and UPC codes), or for which the check digit is never displayed (Code 128 and GS1-128).

ShowText

Type: BOOL

Default: TRUE

When FALSE text version of the code is NOT included in the barcode image. When TRUE text version of the code IS included in the barcode image.

This property is ignored for clocked codes which have no text version.

TextAlign

Not available via http

Type: `StringAlignment`

Default: Center

Allowed values: Near, Center, Far

TextAlign sets the alignment of any text displayed under the barcode. A value of Center centers the text, Near gives left justification and Far gives right justification. Justification is to the edge of the barcode – NOT the edge of the light margins.

Unit

Type GraphicsUnit

Default: Millimeter

The scaling mode applied to values of the following size-related properties:

BarcodeHeight
BarcodeWidth
BearerSize
BorderWidth
MarginSize

Allowed values:

Millimeter
Inch

When passed as an http parameter the value must be passed as a string of “IN” or “MM”.

Xunit

Type: float

Default: 0.0

Allowed range: 8.0 – 255.0

The Xunit property may be used to specify the width (in Mils) of the smallest element in the barcode.

Note that setting this property to a value other than 0 causes the image to resize itself to a width calculated from the number of X units in the barcodes – so the value of the BarcodeWidth property is ignored.

Using values smaller than 8 will produce a barcode image, but that image will not meet standard specifications and may not scan.

Additional 1D properties

The following additional properties are applicable only to http parameter passing:

FontName

Type: BSTR

Default:	Arial
Allowed values:	Any accessible TrueType font installed ON THE SERVER.

FontSize

Type: short

Default:	10
Allowed values:	Any allowed vlaue (typically 6 – 72)

FontBold

Type: BOOL

Default:	FALSE or 0
Allowed values:	FALSE (font is not bold) or TRUE (font is bold)

FontItalic

Type: BOOL

Default:	FALSE or 0
Allowed values:	FALSE (font is not italic) or TRUE (font is italic)

1D Information properties

Note that the barcode symbol is determined only when this Pic() method is called, so the information properties are not available until this method has been called.

Error

Type: integer

Returns a value representing the error code if a valid barcode image cannot be created. Otherwise returns 0.

Read only. Do not set this property.

The error codes and corresponding Status property values are shown below:

Status

Type: string

Returns a string interpreting the value of the Error property.

The error codes and corresponding Status property values are shown below:

Error	Status
0	OK
1	Illegal character in data
2	Wrong data length
3	Error in barcode data
9	Graphics drawing error

Read only. Do not set this property.

1D ASP.NET interface

The ASP.NET interface is provided by the files contained in the DLBC folder, consisting of:

DLBC.ASPX – the script pointed to by reports creating a barcode image
GLOBAL.ASAX – generic Run at server file
GLOBAL.VB – file for creating session state variables

The sample supplied creates an instance of the image creator each time the script is accessed and then sets the properties most commonly used. Additional properties may be set in this code if required by adding lines to set the values of any of the properties defined in the IIS Servlet Properties section above. Multiple scripts may be created by copying the dlbc.aspx file to different files names where multiple default barcode setups are required.

The properties that are likely to be changed in a report are passed to the ASP.NET script as parameters immediately after the page URL, eg.

http://localhost/dlbc/dlbc.aspx?CodeType=9&ImageHeight=12&Xunit=12&MarginSize=4&Data=12345

The first parameter is preceded by a ? symbol and subsequent parameters are separated from one another by & symbols.

The ASP.NET script collects these parameters and, along with any other image creator properties that are hard coded in the file, these are passed to an instance of image creator. A barcode image is then obtained as a MemoryStream object using the Pic(0) method, eg.

```
Session("ms")=m_barcode.Pic(0)
```

which returns a Windows Metafile image.

If a Portable Network Graphic (png) image is required this may be obtained using the Pic(ires) method, eg.

```
Session("ms")=m_barcode.Pic(300)
```

which generates a Portable Network Graphic image with a pixel resolution of 300 dpi.

Finally the ASP.NET script returns the image to the report. The content type must be set to “application/octet-stream” using

```
Response.ContentType =  "application/octet-stream"
```

And the image is returned using:

```
Session("ms").WriteTo(Response.OutputStream)
```

1D Font barcodes

Font barcodes may be incorporated into any version of Crystal Reports using the barcode fonts supplied with Barcodes for Crystal Reports and the accompanying CRUFL_DL1D (User defined function library). . (The older UFL U25UNIF.DLL is provided for compatibility with previous releases – see The Legacy UFL).

For 1D barcodes the fonts are UNAxx, UNBxx, UNCxx, UNDxx, UNExx and UNFxx, where the xx refers to the font weight. W fonts produce the full bar and space sizes and these fonts are installed by the setup program, while the R, N and T fonts produce bars in which the bar width has been reduced by 8, 16 and 20% respectively. These reduced width fonts are not installed but may be found in the fonts folder in the Barcodes for Crystal Reports installation directory. They may be installed using the Fonts applet in the Windows Control Panel is required.

The sample report included with the kit is "Order Packing CRUFL1D.rpt" and will be found in the relevant CR version subdirectory of the kits installation directory. This is a modified version of the "Order Packing List.rpt" report included with the Crystal Reports Xtreme Mountain Bike Inc samples. If you do not have the samples installed on your system then the modified sample report will not work and you should skip to the section "Creating a barcode on a report" below. The report displays barcodes using one of the barcode fonts installed on your system; if for any reason that font cannot be found then the report will not display a barcode - but a collection of apparently meaningless characters. (You can change the font used to display the barcode as described below, and the barcode will then display correctly.)

The "Order Packing xxxx.rpt" sample may be run on a machine that contains a full Crystal Reports installation just by double clicking on the report file. A barcode is included on the report with its data taken from the one of the data fields.

Switching the report to Design view, Right-clicking on the barcode and selecting Edit Field Object from the drop-down menu displays the Function (DL1DD1UflBC for 1D barcodes) and parameters used to generate the barcode. The first parameter is the barcode data (a string), the second parameter is an integer that specifies the barcode type (eg. code type 8 is Code 39 for the 1D system), and the remaining parameter (normally 0) can be used to specify extra barcode features as described in the Reference section.

Four additional fonts are installed for the creation of “cosmetic” GS1 barcodes – these barcodes have the human readable text distributed under the bars, and have elongated start, centre and stop bars. These fonts are DBxE EANUPC and DCx EANUPC, where x is W for the full width bars font or R for the reduced width bars font. These fonts can be used to create cosmetic EAN, UPC and ISBN barcodes when the GS1 flag parameter is set.

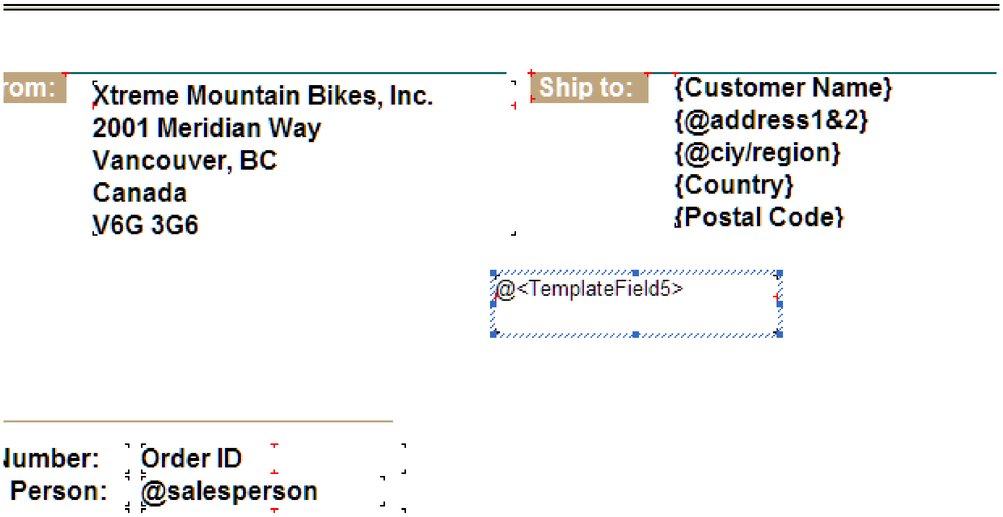
The sample report included with the kit is "Order Packing GS1CRUFL1D.rpt" and will be found in the installation directory. This uses the DCRE EANPC font to show cosmetic EAN-13 barcodes.

Creating a 1D font barcode on a report

To create your own barcode on a report follow the steps below; the instructions are provided for creating a 1D barcode on Crystal Reports 9 and 10, with the minor difference for v 8.5 mentioned in brackets:

Step 1

From Insert menu choose Template Field Object; ensure that the field is wide enough to hold the entire barcode with clipping off the right hand side!



rom:
Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to:
{Customer Name}
{@address1&2}
{@ciy/region}
{Country}
{Postal Code}

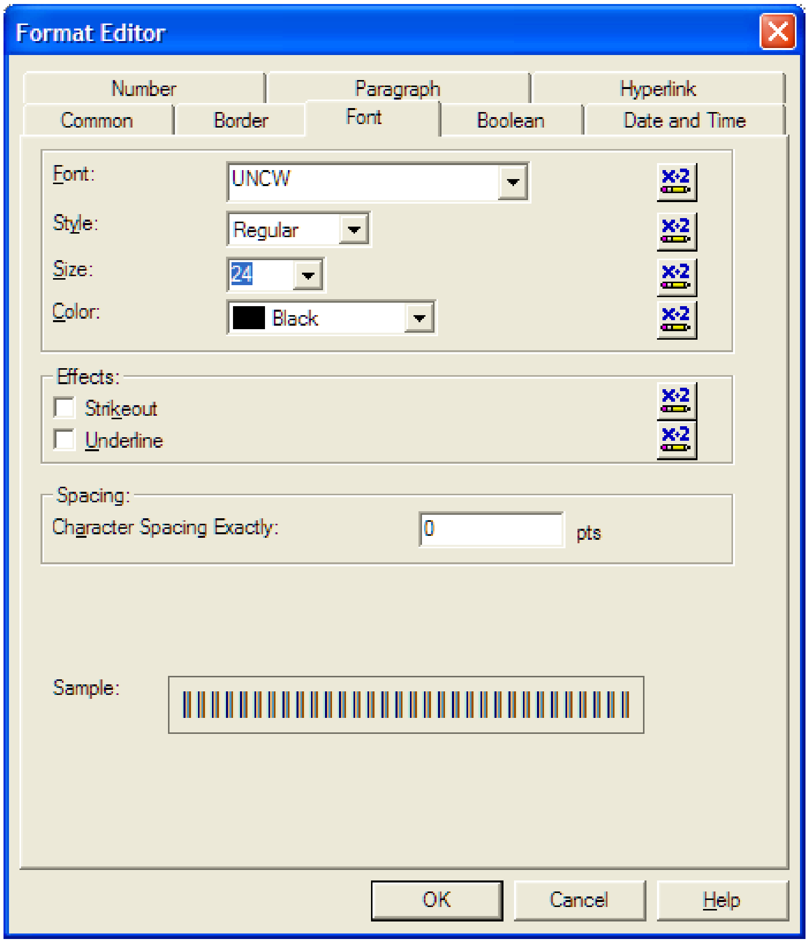
@<TemplateField5>

lumber: Order ID
Person: @salesperson

Step 2

Right click on the object and choose Format Template Field.
Check the checkbox labeled Can Grow

Select the Font tab from the Format Editor dialog and select a suitable font, such as UNCW 24 point. Then push the OK button



Font: UNCW

Style: Regular

Size: 24

Color: Black

Effects:

□ Strikeout

□ Underline

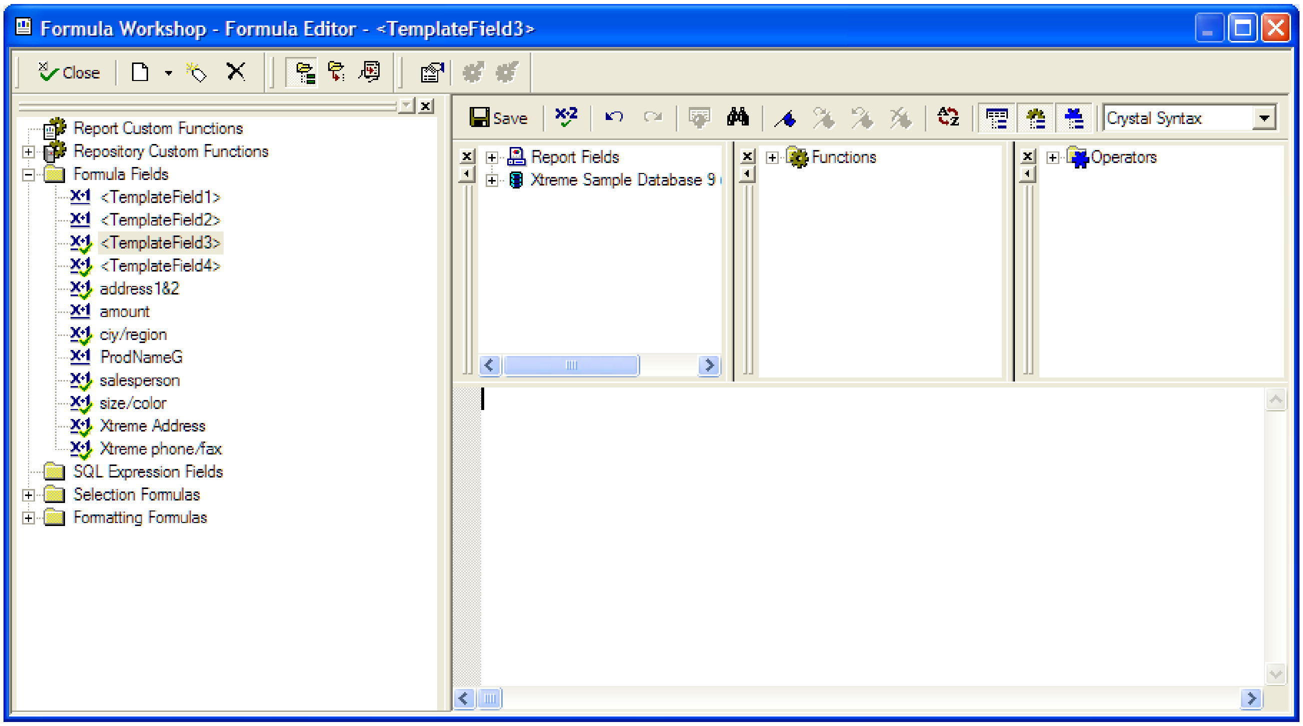
Spacing:

Character Spacing Exactly: 0 pts

Sample:

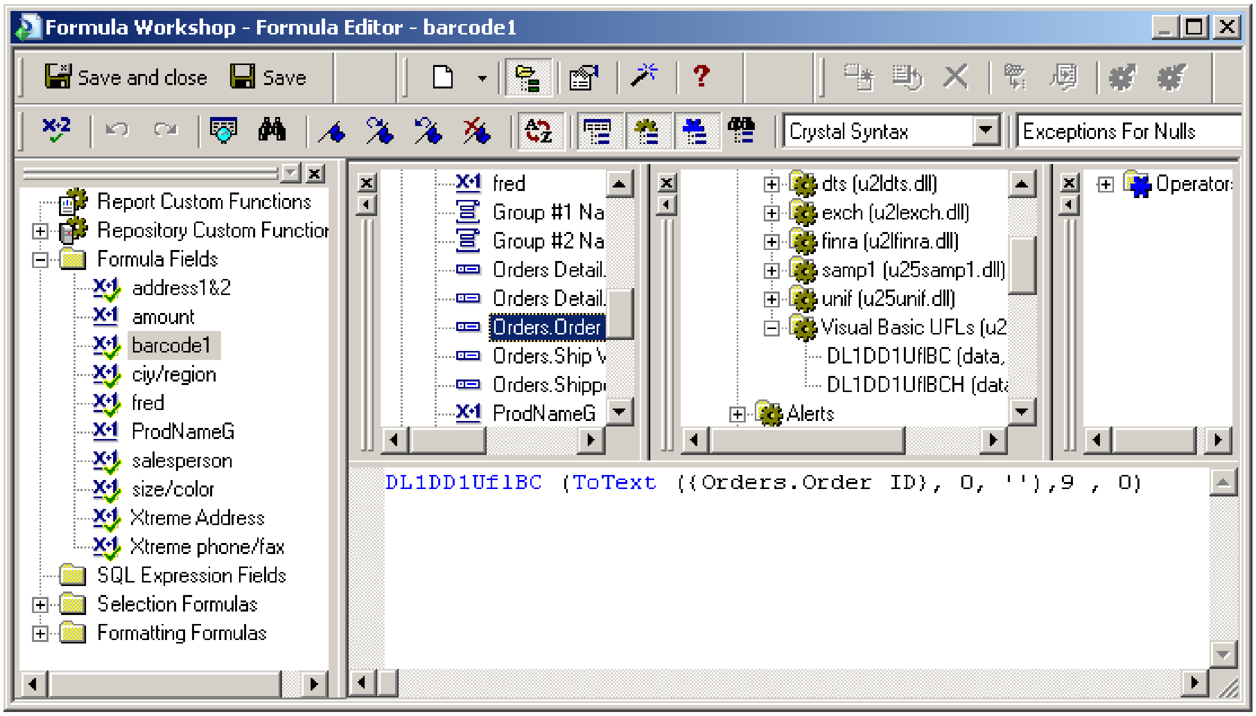
Step 3

With the Field Object selected, from the Report menu select Formula Workshop. Expand the Formula Fields list and double click on the field you have inserted



Step 4

Now enter the formula, such as
DL1DD1UflBC({Customer.Postal Code}, 9 , 0)



```
DL1DD1UflBC (ToText ({Orders.Order ID}, 0, ''),9 , 0)
```

Note that if the data is numeric it should be converted to text using the ToText() function as illustrated above.

The first parameter in the DL1DD1UflBC function must be a string containing the barcode data. This can be a literal string (ie. data enclosed in quotation marks, such as "1234"), or field data. If field data is to be used it must be text data - so if the required field actually contains numeric data this must be converted into text data.

If a required data field contains text data just double click on the field name in the list of Report Fields, and the field name enclosed in curly brackets will be copied to the function's first parameter position. eg.

DL1DD1UflBC({Customer.Region},,)

If a required field contains numeric data then expand the Strings item in the list of functions and the expand the ToText function; Now select the required function - which will usually be ToText(x,y,z) where the x represent the number to be used as data, y represent the number of decimal places (typically 0) and z is a character used to separate thousands from hundreds etc (which unfortunately defaults to a comma and is generally not wanted in a barcode). Double click on the required version of the ToText function and this will be copied to the first parameter position of the DL1DD1UflBCfunction, with the cursor now placed in the first parameter position of the ToText function, ie.

DL1DD1UflBC(ToText (, ,),,)

Now double click on the required data source field in the list of Report fields, eg.

DL1DD1UflBC(ToText ({Orders.Order ID}, ,),,)

and fill in the other two ToText parameters with a 0 (the number of decimal places) and a NULL character (two single quotes) respectively, ie.

DL1DD1UflBC(ToText ({Orders.Order ID},0 ,'’),,)

The second parameter of the DL1DD1UflBCfunction represents the type of the barcode to be created. This value (shown in the Barcode Types Table in the UFL Reference) must be an integer..

The final parameter is a flags parameter (normally a 0), also an integer, eg.

DL1DD1UflBC(ToText ({Orders.Order ID},0 ,''),8,0)

The Flags parameter is a number made up by adding values for the following barcode properties:

Property	*value*
Automatic check digit	1
Wide bars	2
GS1 flag	8 (using this flag requires the use of one of the DxxE EANUPC fonts
Extra 1	16
Extra 2	32

The uses of the Extra 1 and Extra 2 parameters are described in the Barcodes Help file for individual barcode types.

Step 5

Finally push the Save button, then the Close button (or just the Save and Close button in CR 10).
Return to the report's Preview display and you should have a barcode. If you are using data from a database as the data source for the barcode, then the barcode will change as you navigate around the record source.

Note that font barcodes do not include human readable text under the bars. If you wish to include human readable data you should add a text field under the barcode using data from the same source as the barcode.

To make changes to the barcode formula

Open the report in Design view, right click on the barcode (or its' empty box) and select Edit Formula [Edit Field Object in v 8.5] from the drop down menu.

Follow the procedure above to replace the formula.

Note that invalid barcodes will not be visible. Some barcode types support only digits, other support only digits and upper case letters. Barcode that support only digits do not support spaces!!

Note that for font barcodes to be visible on reports viewed through the Report Application Server the font must be installed on the client computer.

DL1DD1UflBCH ()

The DL1DD1UflBCH() function takes the same parameters as the DL1DD1UflBC() function, but returns the human readable text form of the barcode – including check digit if the Flags parameter contains 1. This is useful for obtaining the human readable form complete with check digit when the available data does not contain a check digit.

The 1D CRUFL parameters

The User Function Library generates a string of characters from data, and when those characters are displayed in the correct font they appear as a barcode. The UFL is called from a Template Field Object placed on a report when its formula has been specified in the Formula Workshop.

A typical formula would be,

DL1DD1UflBC(ToText ({Orders.Order ID},0 ,' '), 8, 0)

The formula has three parameters.

The first is the source of the text data for the barcode. If the source actually contains numeric data then the Crystal Reports ToText() function must be used to convert the data into text (as illustrated above).

The second parameter is the barcode type number. The available barcode types are shown in the table below.

The final parameter is the Flags parameter, which is a number made up by adding values for the following barcode properties:

Property	*value*
Automatic check digit	1
Wide bars	2
GS1 flag	8 (using this flag requires the use of one of the DxxE EANUPC fonts
Extra 1	16
Extra 2	32

The uses of the Extra 1 and Extra 2 parameters are described in the Barcodes Help file for individual barcode types.

Barcode Types Table for the 1D CRUFL

Barcode type	CodeTypeValue
Code 39	0
Extended Code 39	1
Codabar	2
2 of 5	3
Interleaved 2 of 5	4
Matrix 2 of 5	5
Code 93	6
Extended Code 93	7
Code 128	8
GS1 128	9

SSCC	10
EAN 14	11
EAN 13	12
EAN 13 + 2 digits	13
EAN 13 + 5 digits	14
EAN 8	15
EAN 8 + 2 digits	16
EAN 8 + 5 digits	17
UPC-A	18
UPC-A + 2 digits	19
UPC-A + 5 digits	20
UPC-E	21
MSI/Plessey	22
Plessey	23
DeutschenPost	24
PostNet	25
Planet 12	26
Planet 14	27
Royal Mail	28
4-State	29
ISBN	30
ISSN	31
ISMN	32
TelePen Standard	33
TelePen ASCII	34
TelePen Numeric	35
Japan Postal code	36
Code 11	37
Code B	38
ITF-6	39
ITF-14	40
IATA 2 of 5	41
China Post	42
OneCode	43
Code 128 Type A	44
Code 128 Type B	45
UPC-E0	46
UPC-E1	47
Korean Postal Authority	48
Italia Post 2/5	49
Italia Post 3/9	50
Australia Post	51
InfoMail A	52
PZN	53
Databar Omnidirectional	54
Databar Truncated	55
Databar Limited	56
Databar Expanded	57
HIBC LIC Code 128	58
HIBC LIC Code 39	59
IM^{TM} Package Barcode	60

2D Picture barcodes in CR XI or later

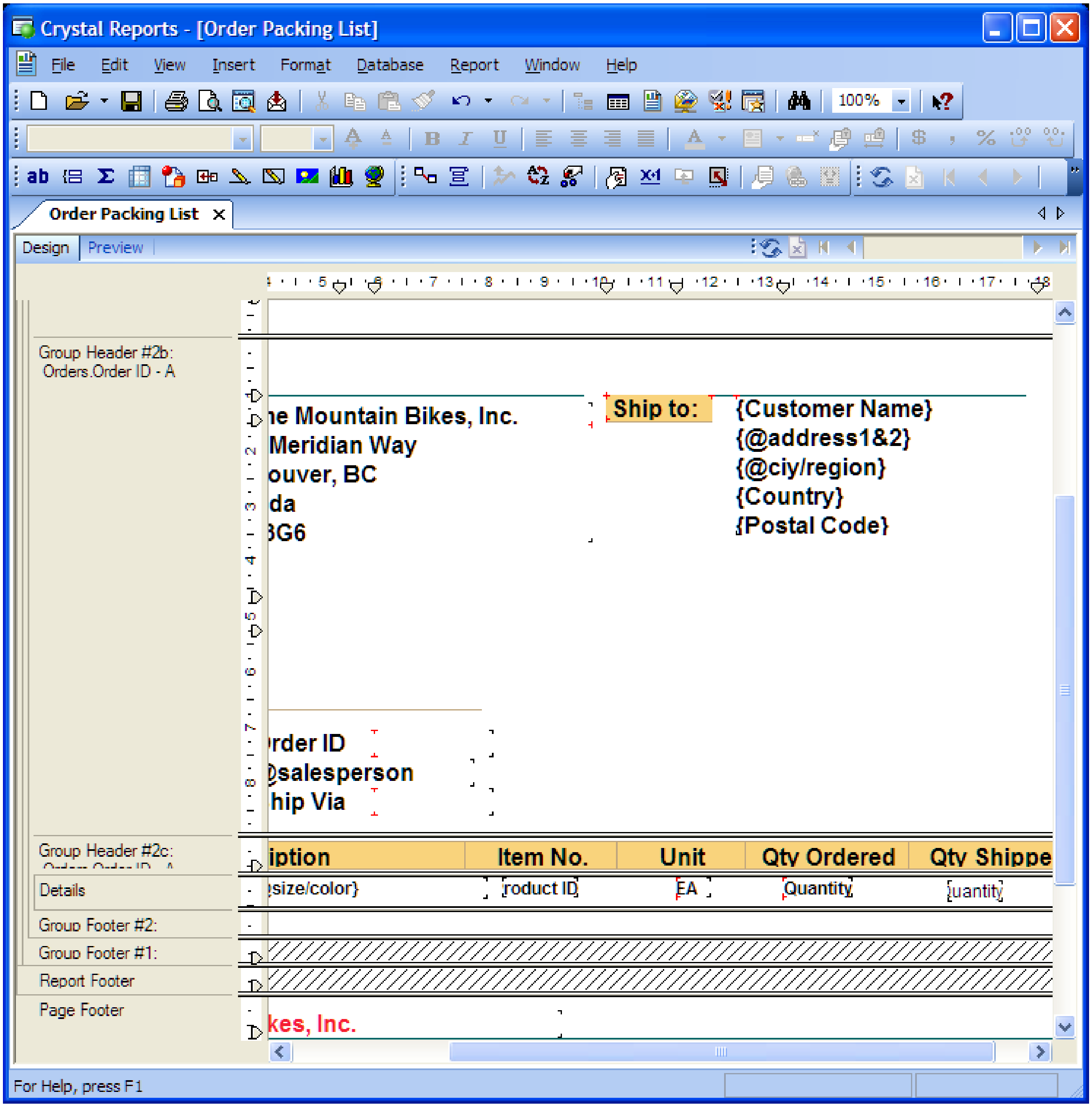
Picture barcodes may be embedded into reports created with Crystal Reports XI or later. Picture barcodes are Windows Metafile (wmf) images by default. Details on selecting alternative image types are in the IIS servlet section.

Creating embedded 2D barcode images

To place embedded barcode images into an existing report follow these steps.

Step 1.

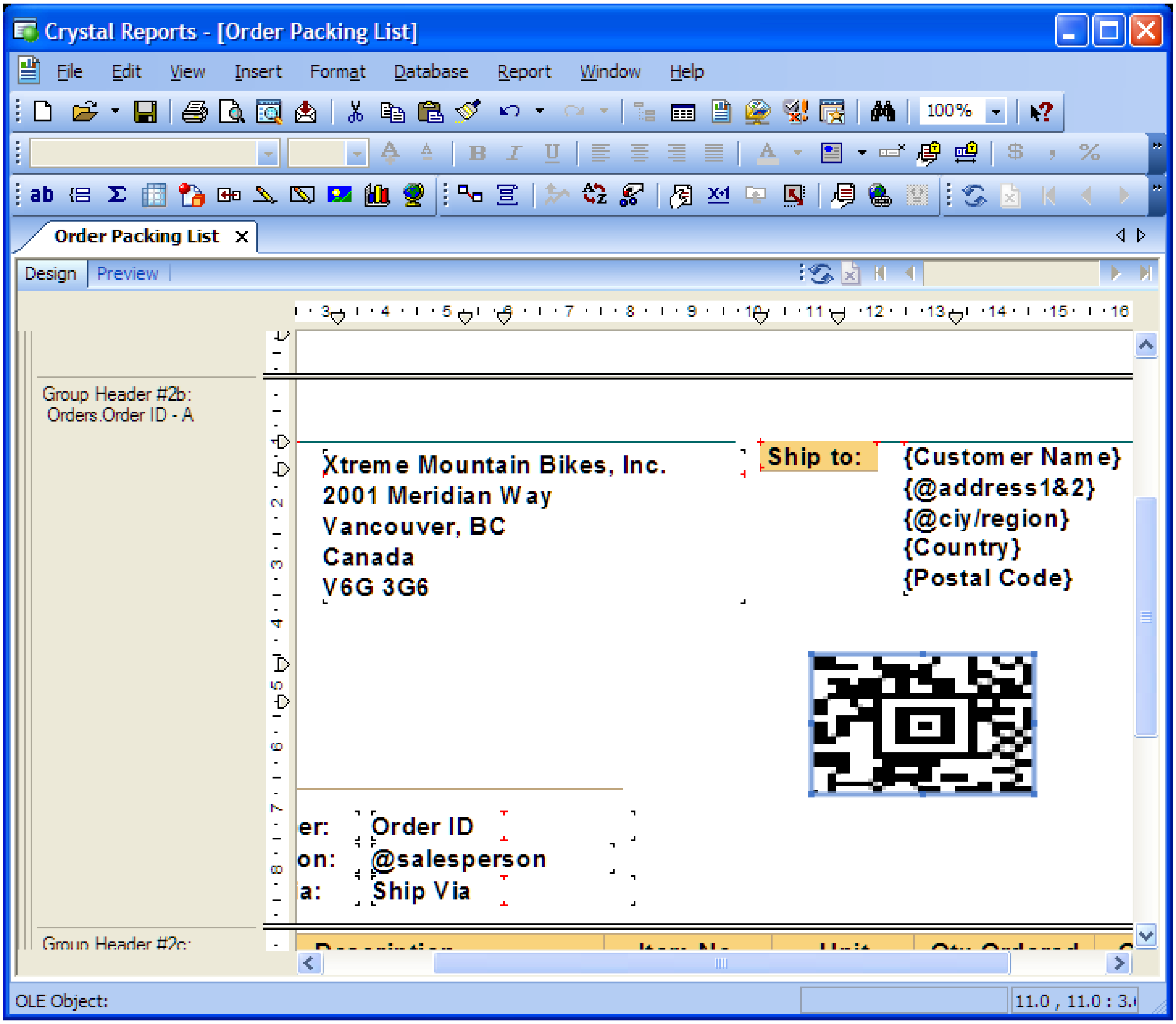
Open the report to which you wish to add barcodes



Step 2.

From the edit menu choose Insert and then choose Picture. Navigate to the 2D Barcode Tools for Crystal Reports installation directory and choose one of the dummy2D barcode image files, then push the Open button.

A movable rectangle appears. Position this at the point at which you wish the barcode to appear then click the left mouse button. Do not change the size of the rectangle. The rectangle containing a dummy barcode image appears on the report.



Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

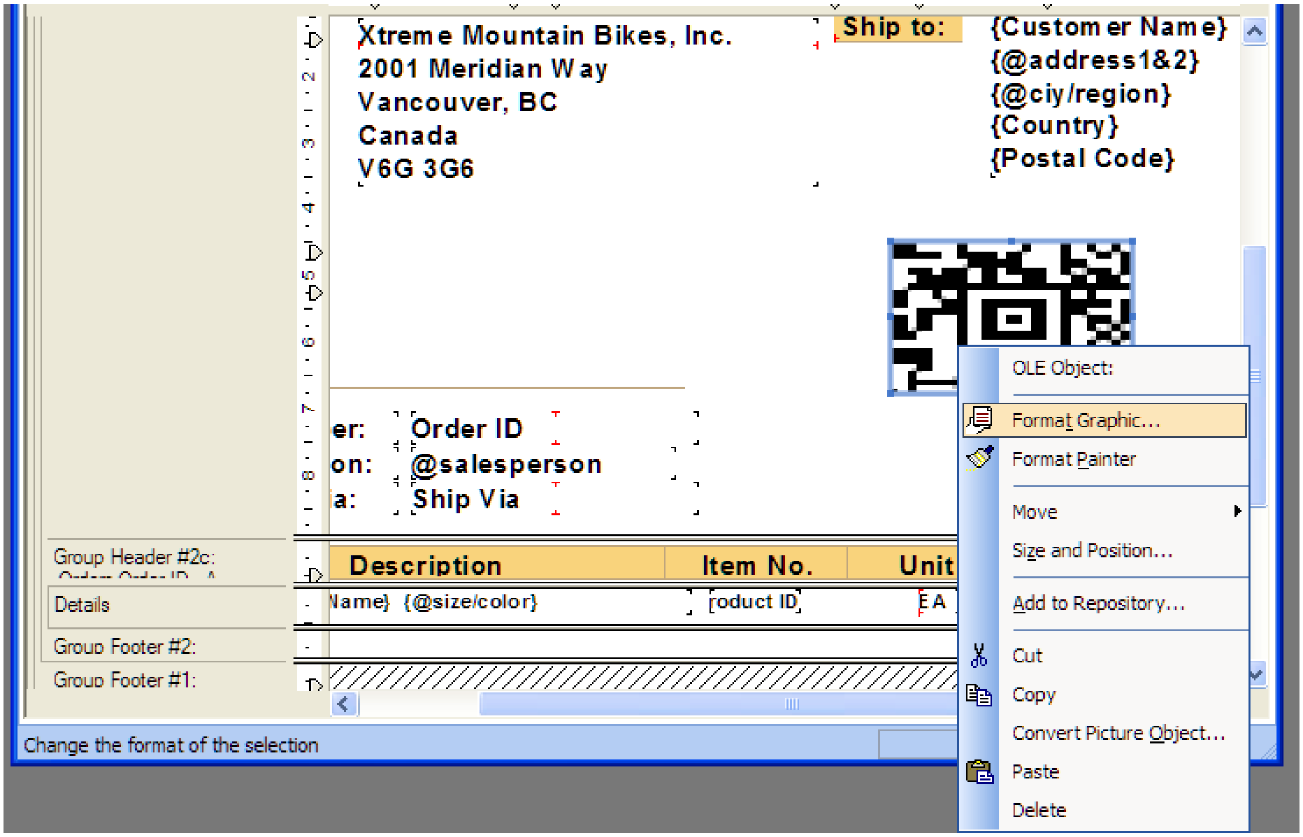
Ship to:

{Customer Name}
{@address1&2}
{@ciy/region}
{Country}
{Postal Code}

er: Order ID
on: @salesperson
a: Ship Via

Step 3.

Right-click on the selected barcode image and choose Format Graphic from the pop-up menu that appears.



Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to:

{Customer Name}
{@address1&2}
{@ciy/region}
{Country}
{Postal Code}

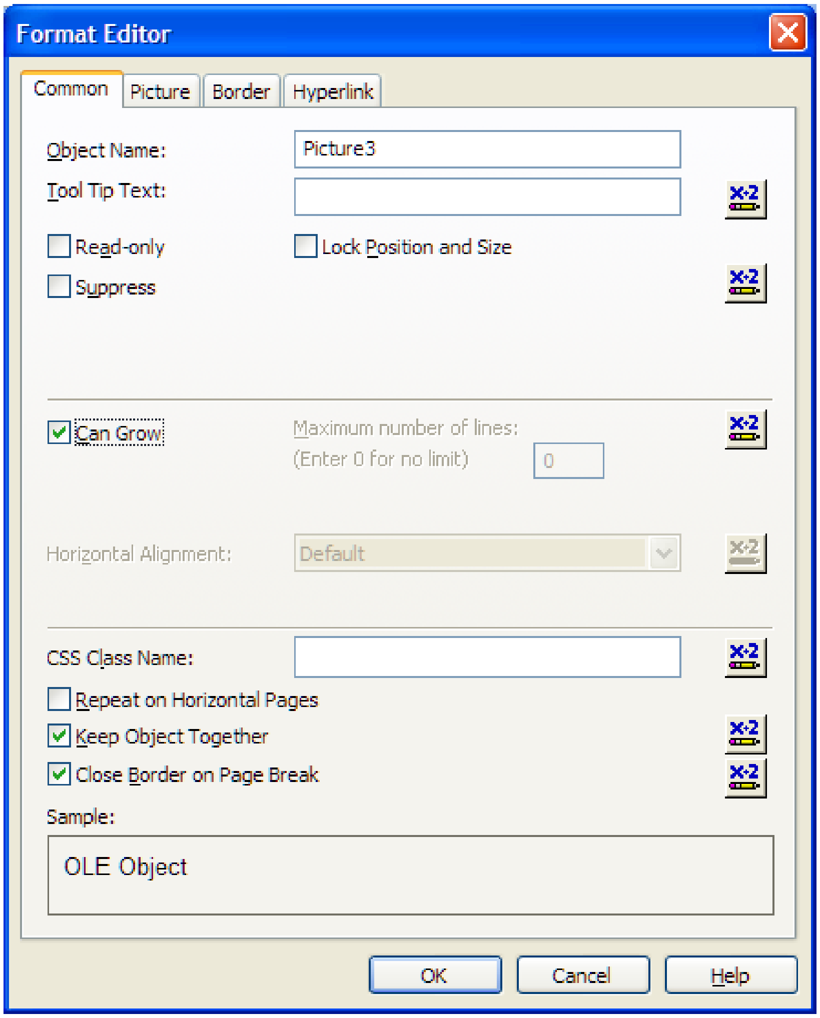
er: Order ID
on: @salesperson
a: Ship Via

Description	Item No.	Unit
Name} {@size/color}	roduct ID	EA

Change the format of the selection

Step 4.

In the Format editor dialog that appears ensure that the Can Grow check box is checked then select the Picture tab.



Object Name: Picture3

Tool Tip Text:

□ Read-only

□ Lock Position and Size

□ Suppress

☑ Can Grow

Maximum number of lines:
(Enter 0 for no limit) 0

Horizontal Alignment: Default

CSS Class Name:

□ Repeat on Horizontal Pages

☑ Keep Object Together

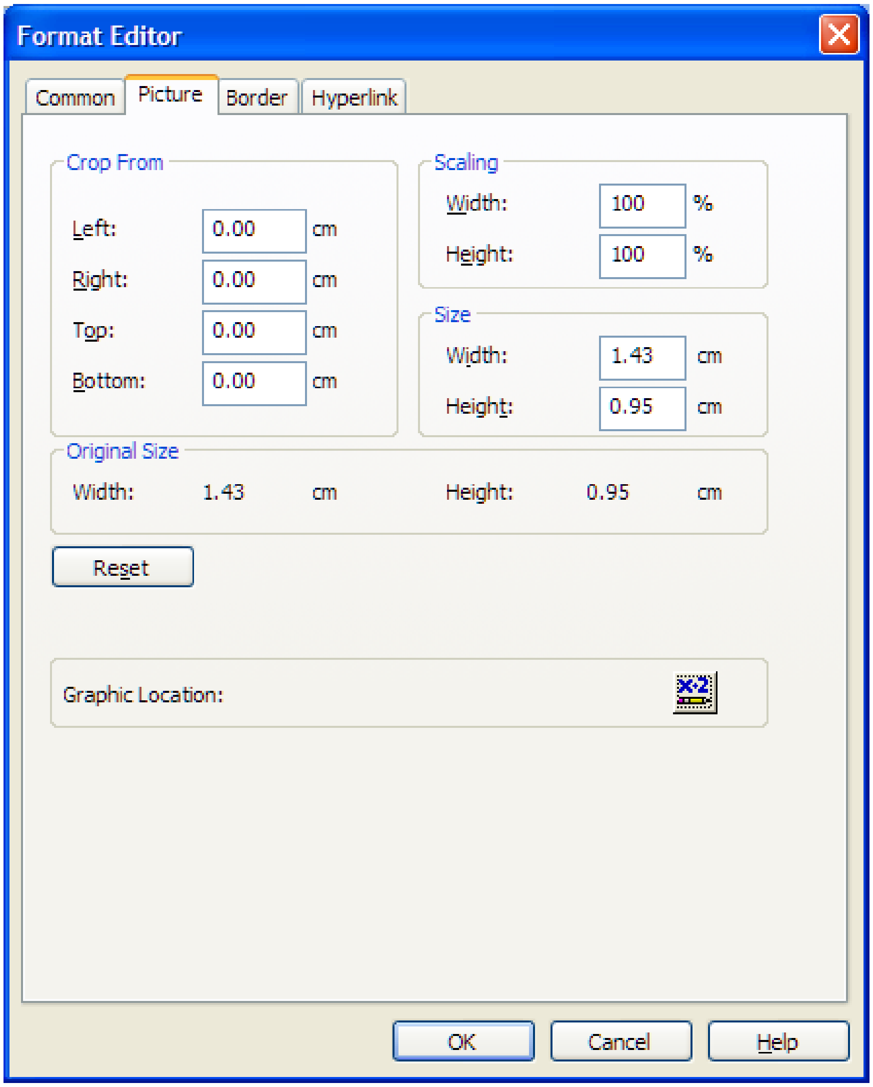
☑ Close Border on Page Break

Sample:

OLE Object

Step 5.

Push the icon that appears alongside the Graphic Location label



Crop From

Left: 0.00 cm

Right: 0.00 cm

Top: 0.00 cm

Bottom: 0.00 cm

Scaling

Width: 100 %

Height: 100 %

Size

Width: 1.43 cm

Height: 0.95 cm

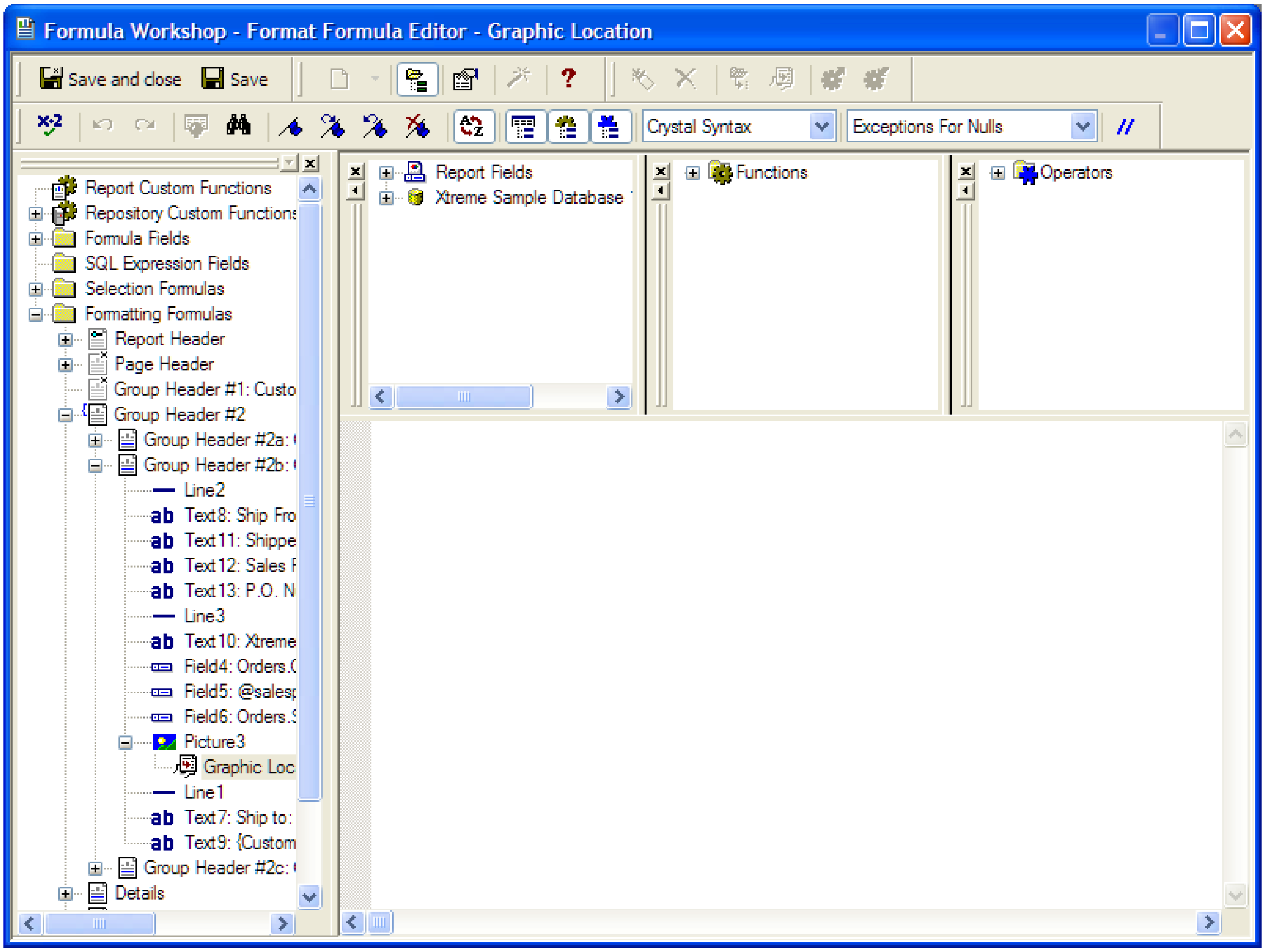
Original Size

Width: 1.43 cm Height: 0.95 cm

Graphic Location:

Step 6.

The Formula Workshop dialog appears. If the Graphic Location item for your picture is not already selected, expand the entries under Formatting Formulas and select the Graphic Location item.



Step 7.

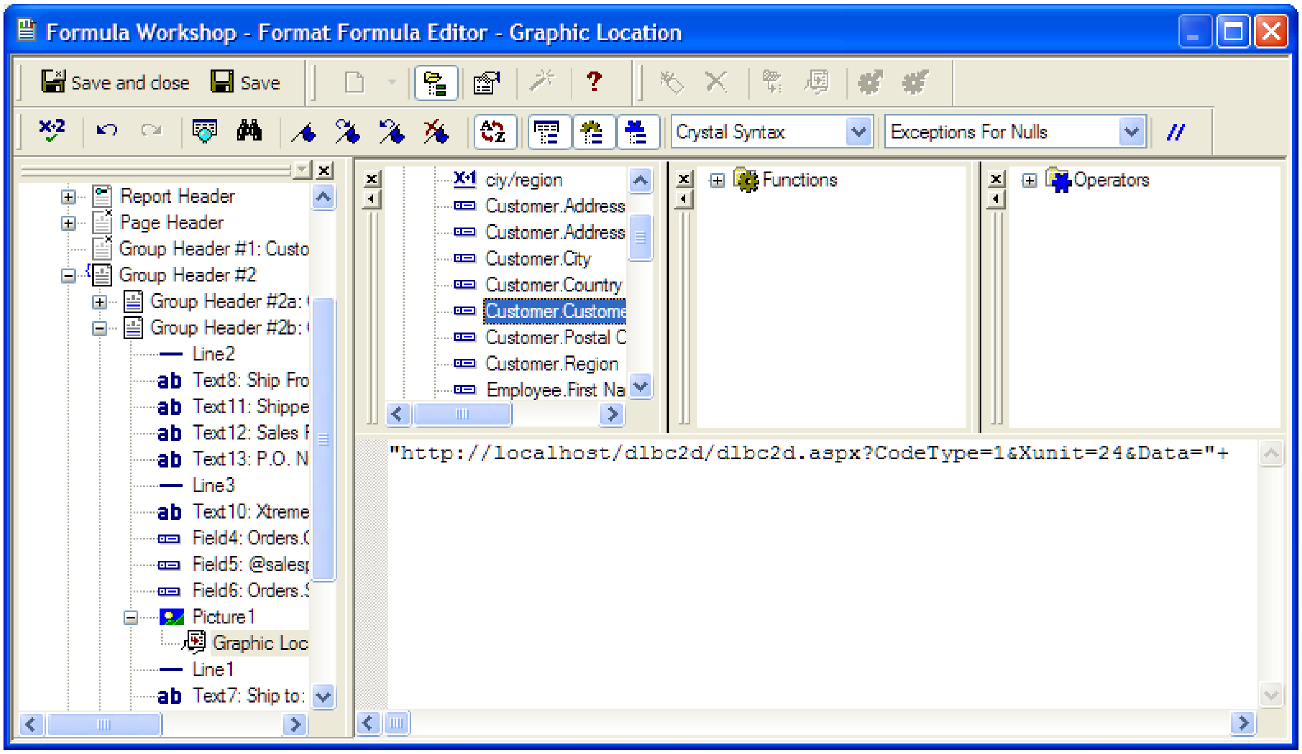
In the formula space enter the first part of the formula, such as

"http://localhost/dlbc2d/dlbc2d.aspx?CodeType=1&Xunit=24&Data="+

including the quotation marks and the + sign at the end. Click once just after the + sign to ensure that the insertion point cursor is flashing.

The last item within the quotation marks is "Data=" which is pointing to the data that will be used to create the barcode.

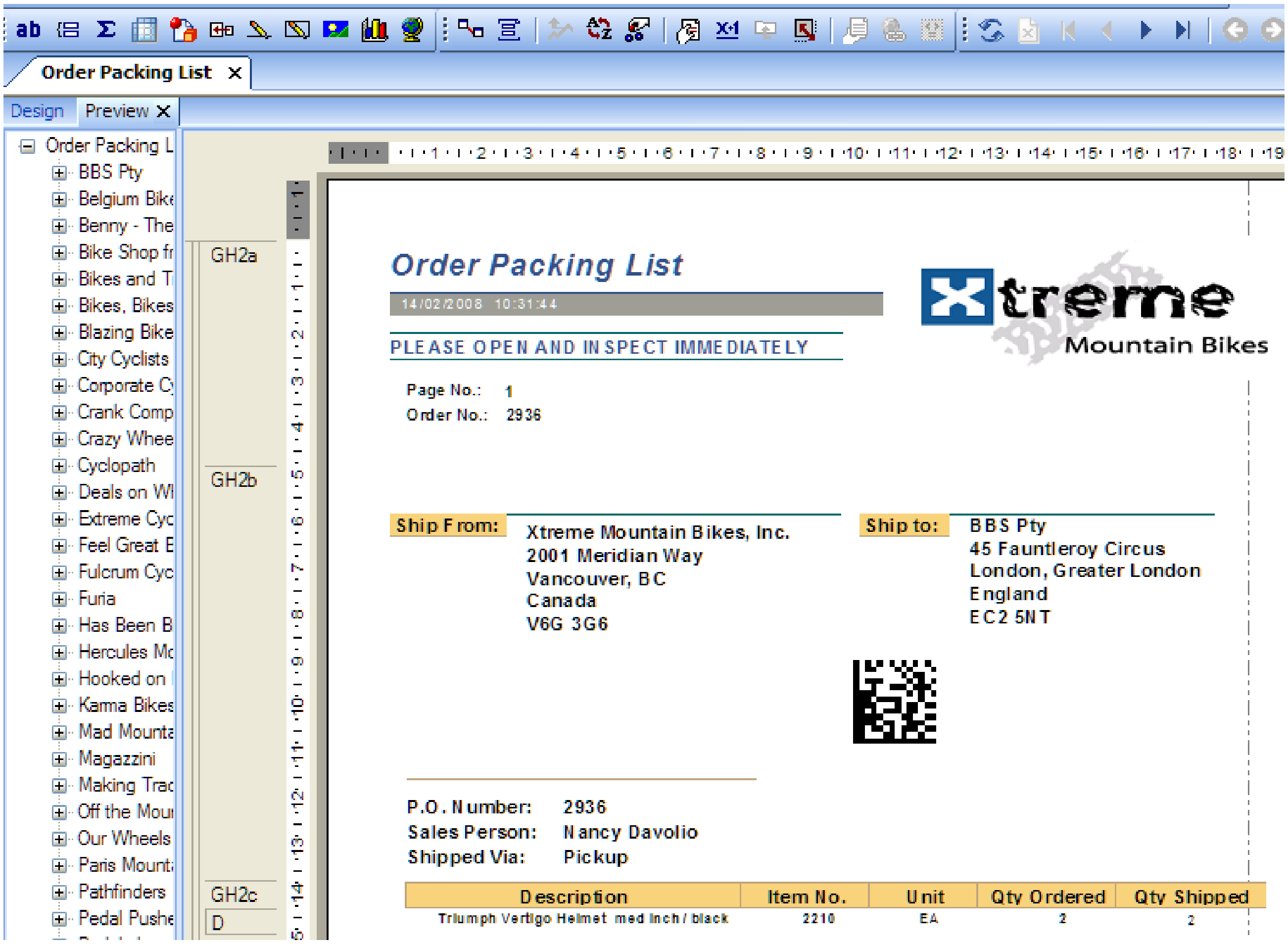
Expand the Report Fields tree and locate the field you wish to use as the data source (the Customer.Postal Code field in this example). Double-click on the field name and the name (complete with braces) will be copied to the end of the formula.



```
"http://localhost/dlbc2d/dlbc2d.aspx?CodeType=1&Xunit=24&Data="+
```

Step 8.

Push the Save and Close button, then push OK button in the Format Editor to return to the main window in Design view. Switch to Preview and the report complete with barcodes will be visible



Order Packing List

14/02/2008 10:31:44

PLEASE OPEN AND INSPECT IMMEDIATELY

Page No.: 1
Order No.: 2936

Ship From: Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to: BBS Pty
45 Fauntleroy Circus
London, Greater London
England
EC2 5NT

P.O. Number: 2936
Sales Person: Nancy Davolio
Shipped Via: Pickup

Description	Item No.	Unit	Qty Ordered	Qty Shipped
Triumph Vertigo Helmet med Inch / black	2210	EA	2	2

If you Save the report complete with data then the barcode images become embedded within the report. If the data changes the report should be refreshed by selecting Refresh Report Data from the Report menu.

Creating the 2D formula

While every requirement will need a different formula there are some general points that should be taken into account when creating the formula to generate the barcode symbols. Every formula will need to specify a CodeType and we recommend the use of the Xunit parameter to specify the basic element width for the symbol, so every symbol will start with code of the form:

"http://localhost/dlbc2d/dlbc2d.aspx?CodeType=1&Xunit=24

Many barcode types have a variety of Modes – for example Databar symbols can be generated in any one of seven modes – so you may need to check whether the barcode type you wish to use has any mode requirements, If it does then code such as

&Mode=1

may need to be included.

PDF417, MicroPDF417 and all Databar symbols are made of bars which are not square but have a height which is a multiple of the width of the narrowest bar (the Xunit value). The multiple is specified using the Ysize parameter. For PDF417 symbols this can range from 2 – 5, while for Databar symbols it can be in the range 7 – 50. So to generate one of these symbols the formula must contain code of the form

&Ysize=20

Many other optional parameters are available and are listed in the Reference section.

Finally every formula needs to end with code that specifies the Data, using code of the form &Data=12345 or, much more likely, &Data= followed by the source of the data, for example

http://localhost/dlbc2d/dlbc2d.aspx?CodeType=7&Xunit=12&Mode=6&Columns=3&Ysize=30&Data=+{fieldname}

for a Databar Expanded-Stacked symbol with an Xunit value of 12 Mils, a bar height of 30* Xunit, 3 codeword pairs in the first row, and the data taken from a field name fieldname.

Concatenating data fields

Data from multiple fields may be included in a barcode by concatenating the fields in the Data section of the source formula. For example, two fields may be concatenated by specifying the fields with a + symbol between the, eg.

Data="+{Customer.Postal Code}+{Customer.Country}

which sets the data equal to the content on the Customer Postal Code field followed immediately by the content of the Customer Country field.

However, it is more likely that multiple fields need to be separated, typically by tabs or newlines. This may be achieved by concatenating characters between the fields using escape sequences; for example:

Data="+{Customer.Postal Code}}+"^p"+{Customer.Country}

inserts a new paragraph (carriage return + line feed) between the postal code data and the country data.

The following escape sequences are defined for the image creators:

^t	Horizontal Tab character
^r	Carriage Return character
^n	Newline character
^p	new paragraph (Carriage Return followed by Linefeed)
^s	space character

So to produce a barcode containing a multi-line name and address the following formula could be used:

"http://localhost/dlbc2d/dlbc2d.aspx?CodeType=1&Xunit=16&Data="+
{Customer.Customer Name}+"^p"+{@address1&2}+"^p"+
{@ciy/region}+"^p"+{Customer.Postal Code}+"^p"+{Customer.Country}

Note that the escape character (^ in the above) may be modified using the Esc property.

Note also that many barcode scanners may be programmed to modify scanned characters – for example, by replacing a carriage return character by the carriage return – linefeed pair.

Using Crystal Reports Export function

The Crystal Reports Export to PDF function does not handle vector graphics - which is the default format in which barcode images are created. So if you need to Export reports (rather than Print to a PDF capable driver - such as Adobe Acrobat) it is necessary to modify the edit the servlet file to generate PNG (Portable Network Graphic) images of the barcodes. For full details see Pic() methods in the IIS 2D Server reference below.

To do this edit the file dlbc.aspx (eg using Notepad) and change the line

Session("ms")=m_barcode.Pic(0)

to

Session("ms")=m_barcode.Pic(300)

The 300 represents the graphics resolution you require. Here 300 dpi is used as an example, but if you intend the items to be printed on a printer with a higher resolution (eg 600 dpi) you could use that value. Note that the size of the image will change with resolution as Barcode Tools adjust the bar size to ensure that each bar in an integer number of dots.

IIS 2D Servlet reference

The IIS servlet is used for Crystal Reports XI and later. The servlet consists of an ASPX file (dlbc2d.aspx) and an associated library.

The image created by the servlet may be obtained within Crystal Reports by setting the source of a Picture element to the servlet's Pic ires() method.

Pic() methods

The pic() methods return a graphic image. These methods may be used by the APS.NET routine for loading an image into a CR XI or later report.

Note: The physical size of a Portable Network Graphic image is determined from the number or pixels per row (or column) and the resolution in pixels per inch. The PNG graphics generated contain the pHYs data block (which contains the size relevant information), but although this is supported by Crystal Reports, not all programs support this and so may display the image at a different size by assuming that there are 96 pixels per inch.

Pic()

returns a png format image using the current resolution setting as set by the DefResolution parameter.

Pic1()

returns a monochrome (1 bpp) png format image using the current resolution setting as set by the DefResolution parameter.

Pic(ires)

Where ires is an integer representing the resolution in dpi of the required image or 0.

If ires is 0 then the image is returned as a Windows Metafile, which is the ideal method for Crystal Reports running on Windows as Metafile images are small vector images.

If ires is greater than 90 then the image is returned as a Portable Network Graphic image with a pixel resolution specified by the value of ires. Typical value for ires would be 300 or 600 (in dpi), and would normally be chosen to suit the printer on which the barcodes will be printed. Lower values are not recommended as these will not generate bar widths that are sufficiently accurate, and higher values generate images that require large amounts of memory.

Pic1(ires)

As Pic(ires) except that the image returned when ires > 90 is a monochrome (1 bpp) Portable Network Graphic image.

Pic(type, imono)

Where type is a string representing the required graphics type, which may be any of:

"bmp" = ImageFormat.Bmp

"gif" = ImageFormat.Gif

"jpg" = ImageFormat.Jpeg

"png" = ImageFormat.Png

"tif" = ImageFormat.Tiff

"wmf" = ImageFormat.Wmf

"eps" = Encapsulated Postscript format

If imono is 0 then the image is returned as a 24 bpp colour graphic for bitmap formats, while if imono is not 0 a monochrome (1 bpp) image is returned.

This function returns an image with the resolution specified by the DefResolution parameter.

Summary of Properties

The image created depends on a number of Properties summarized below and described in more detail in the section following.

The properties that can be set in dlbc2d.aspx:

BackColor	the colour behind the bars
BarRatio	specifies the ratio of wide/narrow bar for some barcode types
BorderWidth	specifies the thickness of any border around the image
Caption	specifies the characters that make up the code
CodeTypeValue	specifies the barcode type required
DefResolution	set the default resolution in dpi for the graphic images.
ForeColor	the colour of the bars and any text under the bars
Orientation	specifies the orientation angle of the barcode image
Reduction	specifies the percentage reduction in bar thickness (useful for allowing for ink spread in wet-ink printing processes.
StartMode	the starting mode for the 2D barcode
Xunit	specifies the thickness of each barcode element in mils (1/1000 inches)

Some barcode types have the following additional properties:

Columns	the target number of columns for Datamatrix, PDF417 or Databar symbols
SecurityLevel	the security or level of the barcode (generally related the the fraction of the barcode devoted to error correction elements
Ymultiplier	the height of a barcode element specified in Xunits

Additional special properties may be set in the dlbc2d.aspx file and are described in the detailed section below.

Properties that can be set in the http call to the server:

BarRatio	specifies the ratio of wide/narrow bar for some barcode types
BorderWidth	specifies the thickness of any border around the image
CodeType	specifies the barcode type required (may be abbreviated to code)
Data	specifies the characters that make up the code
Orientation	specifies the orientation angle of the barcode image

Reduction specifies the percentage reduction in bar thickness (useful for allowing for ink spread in wet-ink printing processes.

Resolution set the resolution in dpi for the graphic image.

StartMode the starting mode for the 2D barcode (may be abbreviated to mode)

Xunit specifies the thickness of each barcode element in mils (1/1000 inches)

Some barcode types have the following additional properties:

Columns the target number of columns for Datamatrix, PDF417 or Databar symbols

SecurityLevel the security or level of the barcode (generally related the the fraction of the barcode devoted to error correction elements

Ymultiplier the height of a barcode element specified in Xunits

Many of these properties have default values, so do not require changing if you can make do with the default values. The properties that must be set for you to obtain a barcode are

CodeType specifies the barcode type required

Data specifies the characters that make up the code

Barcode types table for 2D IIS servlet

The barcode type may be specified by setting the CodeTypeValue to one of the following:

Barcode Type	**CodeTypeValue**
Aztec	0
Datamatrix	1
Maxicode	2
PDF417	3
Micro_PDF	4
Databar	5
QR Code	6
Truncated PDF417	7
Micro QR Code	8
DMRE 8x48	9
DMRE 8x64	10
DMRE 12x64	11
DMRE 16x64	12
DMRE 24x32	13
DMRE 24x36	14
DMRE 24x48	15
DMRE 24x64	16

DMRE 26x32	17
DMRE 26x40	18
DMRE 26x48	19
DMRE 26x64	20

2D Properties

AspectFlag

Type: int

Default: 0

Allowed values 0,1

Determines the interpretation of the Columns property for PDF417 barcode symbols. A value of 1 interprets the Columns number as the number of Rows required in the symbol.

BackColor

Type: Color

Default: Color.White

Allowed values: any allowed Color value.

Sets the colour of the image background. This value may be over-ridden by the Transparent property.

BorderWidth

Type: float

Default: 0.0 (no border)

Allowed values: any

Specifies the size (in units determined by the setting of the PageUnit property of the Graphics object passed into the component) of a border placed around the barcode. The border area is created using the background color set by the BackColor property.

Caption

Type: string

Default: "dBarcode.NET"

Allowed values: Any text string.

Note: only text strings recognised as valid barcodes will result in a barcode picture. An Illegal character in the text string will cause an Error value to be set.

CodeType

The CodeType should be provided using the CodeTypeValue parameter in this release.

CodeTypeValue

Type: integer

Default: 0

Allowed values: The ranges of values defined for the individual components are shown in the barcode type tables below.

The barcode type can be set in http calls using either the CodeType or CodeTypeValue property

The possible values for CodeType and CodeTypeValue are shown in the table above.

Columns

Type: integer

Default: 0

Allowed values: 0 – maximum size of symbol

Applies to PDF417, MicroPDF, Datamatrix, Databar (expanded stacked) and Codablock barcodes only

For PDF417, MicroPDF, Databar and Codablock “Columns” refers to the number of datawords in a row – NOT the number of bars. Note that requirements of the symbology mean that not all values are possible. For example, PDF417 requires a minimu of 3 rows irrespective of the number of columns.

For Datamatrix symbols “Columns” refers to the number of square elements in a row. Only specific symbol sizes are permitted in Datamatrix. If an invalid number of columns is specified the next valid number will be used. Columns should be set to zero if you wish the symbol size to be determined by the security level setting.

DeutschePost

Type: Boolean

Default: False

Applies to Datamatrix barcode type only.

When set to true, forces Damatrix symbols to be created using Base256 encoding as required in some DeutschePost applications.

DefResolution

Type: integer

Default: 96

Sets the default resolution for barcode images.

Esc

Type: string

Default: null

If the Esc string is set to a value then the first character of that string is used as an escape character, which allows specific strings to be inserted into a SetParam() data string using an escape sequence.

For example if the Esc string is set to “^”, then the string “ABC^tDEF” is changed so that ^t is replaced by a Tab character.

The escape sequences available are:

^t	horizontal tab
^n	newline
^r	carriage return
^p	carriage return + linefeed

In each case the ^ character may be any character as set by the Esc property.

Note: this facility is provided to allow concatenation of fields to generate barcodes using the Pic() method. Escape sequences are NOT converted during any other calls.

Flag

Type: Boolean

Default: False

Sets the encoding mode. When true symbols are encoded in Unicode (UTF8). When false encoding is ANSI. So when symbols need to contain non-ANSI characters, such as Arabic or Chinese, this property should be set to true.

ForeColor

Type: Color

Default: Color.Black

Allowed values: any valid Color

Sets the colour of the image foreground, i.e. the bars and text colour.

GS1_Flag

Type: bool

Default: false

Sets the GS1 encoding flag on or off for Datamatrix symbols.

Mirror

Type: BOOL

Default: False

A value of True will cause the barcode image to be created as a mirror image. This facility is designed for use with printing technologies that require a mirror image. Mirror image barcodes may not be scannable unless printed correctly.

Orientation

Type: integer

Default: 0 The value of this parameter determines the orientation of the barcode image created.

Allowed values: 0 - 360 degrees

Note that the rotation of text is only supported by TrueType and other rotatable fonts. Note also that some applications do not correctly handle metafiles that contain rotated text.

PairsPerRow

Type: float

Default: 1.0

Allowed values: 0.1 – 10.0

Applies to Databar Expanded-Stacked barcodes only

For Databar Expanded Stacked symbols the value specifies the number of segment pairs in the first row. Fractional values are rounded to the nearest integer. The final row of an Expanded Stacked symbol may contain a smaller number of segments.

PostFlag

Type: bool

Default: false

Applies to Datamatrix barcode type only.

When set true the symbol is created as a Deutsche Post PostMatrix symbol, with two additional vertical bars.

Reduction

Type: float

Default: 0 the thickness of each line drawn on the barcode image is reduced by this percentage amount. This property may be used to compensate for ink spreading during wet-ink printing.

Allowed values: 0 - 50 (%)

A positive reduces the thickness of bars by the calculated amount of the thinnest bar. A negative value reduces the thickness of bars by the percentage applied to the bar thickness (so a bar three units wide is reduced by an absolute amount that is three times greater than that which would apply to a one unit bar)

SecurityLevel

Type: integer

Default: 0

This parameter specifies the security level - i.e. the amount of redundancy built in to the barcode image to allow for errors to be corrected. The greater the security level the larger the barcode image. See 2D-Barcodes Help for details.

Allowed values are 0 - 8 for PDF417.

Allowed values for Aztec codes depend on the mode, and are 0-99 for Normal mode, 1-4 for Compact mode, and 1-32 for Full range mode (the value is ignored for Runes). Note that for Compact and Full range the security level is actually the required number of Aztec layers in the symbol.

For Datamatrix symbols the allowed values are 0 – 23 for Square symbols, or 0 – 6 for Rectangular symbols.

For QR Codes the allowed security levels are 0 – 3. For Micro QR Code the values are 0 – 2.

For other barcode types this parameter should be 0.

StartMode

Type: integer

Default: 0

Some 2D barcode types support different modes of encoding data. This parameter allows the initial mode to be specified, according to the tables below. For other barcode types the mode is determined automatically and the StartMode parameter must be 0.

Allowed values:

PDF417 (see 2D Barcodes Help for details)

StartMode parameter	Mode
0	EXC Alpha
1	EXC Lower
2	EXC Mixed
3	EXC Punctuation.
4	Binary/ASCII Plus
5	Numeric mode

For most applications this parameter should be 0 for PDF417, the algorithm within the DLL will adjust to the most appropriate mode.

Aztec (see 2D Barcodes Help for details)

StartMode parameter	Mode
0	Normal
1	Compact
2	Full range
3	Rune

For most applications a value of 0 is recommended, along with a security level of 0 which provides the default value of 23% + 3 codewords.

For DataMatrix

StartMode value	Datamatrix type
0	Automatic – Square
1	Automatic – Rectangular
2	ASCII encoding – Square

3	ASCII encoding – Rectangular
4	C40 encoding – Square
5	C40 encoding – Rectangular
6	Text encoding – Square
7	Text encoding – Rectangular
8	X12 encoding – Square
9	X12 encoding – Rectangular
10	EDIFACT encoding – Square
11	EDIFACT encoding – Rectangular
12	Base 256 encoding – Square
13	Base 256 encoding – Rectangular

MaxiCode (see 2D Barcodes Help for details)

StartMode Value	MaxiCode Mode
0	obsolete, not supported
1	obsolete, not supported
2	Mode 2 (Structured Carrier Message – Numeric Postal Code)
3	Mode 3 (Structured Carrier Message – Alphanumeric Postal Code)
4	Mode 4 – Standard Symbol (SEC)
5	Mode 5 – Full EEC Symbol
6	Mode 6 – Reader Program (SEC)

Databar codes.

StartMode value	Databar version
0	Omnidirection
1	Truncated
2	Limited
3	Stacked
4	Stacked Omnidirectional
5	Expanded
6	Expanded Stacked

QR Code & Micro QR Code

StartMode value	QR mode
0	Automatic
1	Alphanumeric

2	Byte
3	Kanji*

*Set the Flag property when using Kanji mode. Automatic mode does not detect Kanji; Kanji mode must be selected for Kanji encoding.

Unit

Type GraphicsUnit

Default: Millimeter

The scaling mode applied to values of the BorderWidth property:

Allowed values:

Millimeter
Inch

When passed as an http parameter the value must be passed as a string of "IN" or "MM".

Xunit

Type: float

Default: 15.0

Allowed range: 8.0 – 255.0

The Xunit property may be used to specify the width (in Mils) of the smallest element in the barcode.

Note that setting this property causes the image to resize itself to a size calculated from the number of X units in the barcode

Using values smaller than 8 will produce a barcode image, but that image will not meet standard specifications and may not scan.

For Maxicodes this property is used to specify the L dimension (the height of the image in mm) and defaults to 25.50 mm.

Ymultiplier

Type: float

Default: 3.0

Allowed range: 2.0 – 5.0 for PDF417, 1.0 – 255.0 for Codablock F, Code 49, and Databar

The Ymultiplier property may be used to specify the height (in Xunits) of the elements in the barcode.

Note that setting this property causes the image to resize itself to a size calculated from the number of X units in the barcode

Using values outside the allowed range will generate a barcode that may not scan. Some Databar barcode types have standard values for this property and will ignore the setting of this property.

This property is available only for Codablock F, Code 49, PDF417 and Databar components.

2D Information properties

Note that the barcode symbol is determined only when this Pic() method is called, so the information properties are not available until this method has been called.

Error

Type: integer

Returns a value representing the error code if a valid barcode image cannot be created. Otherwise returns 0.

Read only. Do not set this property.

The error codes and corresponding Status property values are shown below:

Status

Type: string

Returns a string interpretting the value of the Error property.

The error codes and corresponding Status property values are shown below:

Error	Status
0	OK
1	Illegal character in data
2	Wrong data length
3	Error in barcode data
4	Cannot generate barcode
9	Graphics drawing error

Read only. Do not set this property.

2D ASP.NET interface

The ASP.NET interface is provided by the files contained in the DLBC folder, consisting of:

DLBC2D.ASPX – the script pointed to by reports creating a barcode image
GLOBAL.ASAX – generic Run at server file
GLOBAL.VB – file for creating session state variables

The sample supplied create an instance of the image creator each time the script is accessed and then sets the properties most commonly used. Additional properties may be set in this code if required by adding lines to set the values of any of the properties defined in the IIS Servlet Properties section above. Multiple scripts may be created by copying the dlb2dc.aspx file to different files names where multiple default barcode setups are required.

The properties that are likely to be changed in a report are passed to the ASP.NET script as parameters immediately after the page URL, eg.

http://localhost/dlbc2d/dlbc2d.aspx?CodeType=0&Mode=0&Xunit=20&BorderSize=4&Caption=12345

The first parameter is preceded by a ? symbol and subsequent parameters are separated from one another by & symbols.

The ASP.NET script collects these parameters and, along with any other image creator properties that are hard coded in the file, these are passed to an instance of image creator. A barcode image is then obtained as a MemoryStream object using the Pic(0) method, eg.

```
Session(“ms”)=m_barcode.Pic(0)
```

which returns a Windows Metafile image.

If a Portable Network Graphic (png) image is required this may be obtained using the Pic(ires) method, eg.

```
Session(“ms”)=m_barcode.Pic(300)
```

which generates a Portable Network Graphic image with a pixel resolution of 300 dpi.

Finally the ASP.NET script returns the image to the report. The content type must be set to “application/octet-stream” using

```
Response.ContentType =  “application/octet-stream”
```

And the image is returned using:

```
Session("ms").WriteTo(Response.OutputStream)
```

2D Font barcodes

Font barcodes may be incorporated into any version of Crystal Reports using the barcode fonts supplied with 2D Barcode Tools for Crystal Reports and either the accompanying CRUFL (User defined function library) CRUFL_DL2D.DLL. (The older UFL U25U2DF.DLL is provided for compatibility with previous releases – see Appendix).

The fonts are AZTxx, (for Aztec, Datamatrix and QR Code symbols), and PD2xx, PD3xx, PD4xx and PD5xx (for PDF 417 and Micro PDF417), where the xx refers to the font weight and the number (2 – 5) refers to the height/width ratio of the individual elements. W fonts produce the full bar and space sizes and these fonts are installed by the setup program, while the R and N fonts produce bars in which the bar width has been reduced by 8 and 16% respectively. These reduced width fonts are not installed but may be found in the fonts folder in the 2D Barcode Tools for Crystal Reports installation directory. They may be installed using the Fonts applet in the Windows Control Panel is required.

The sample report included with the kit is "Order Packing CRUFL2D.rpt" and will be found in the kits installation directory. This will open in Crystal Reports version 9 or later (the report suitable for Crystal Reports version 8.5 is "Order Packing CRUFL2D85.rpt" in the cr85 folder). This is a modified version of the "Order Packing List.rpt" report included with the Crystal Reports Xtreme Mountain Bike Inc samples. If you do not have the samples installed on your system then the modified sample report will not work and you should skip to the section "Creating a barcode on a report" below. The report displays barcodes using one of the AZTxx barcode fonts installed on your system; if for any reason that font cannot be found then the report will not display a barcode - but a collection of characters. (You can change the font used to display the barcode as described below, and the barcode will then display correctly.)

The "Order Packing CRUFL2D.rpt" sample may be run on a machine that contains a full Crystal Reports installation just by double clicking on the report file. A barcode is included on the report with its data taken from the one of the data fields.

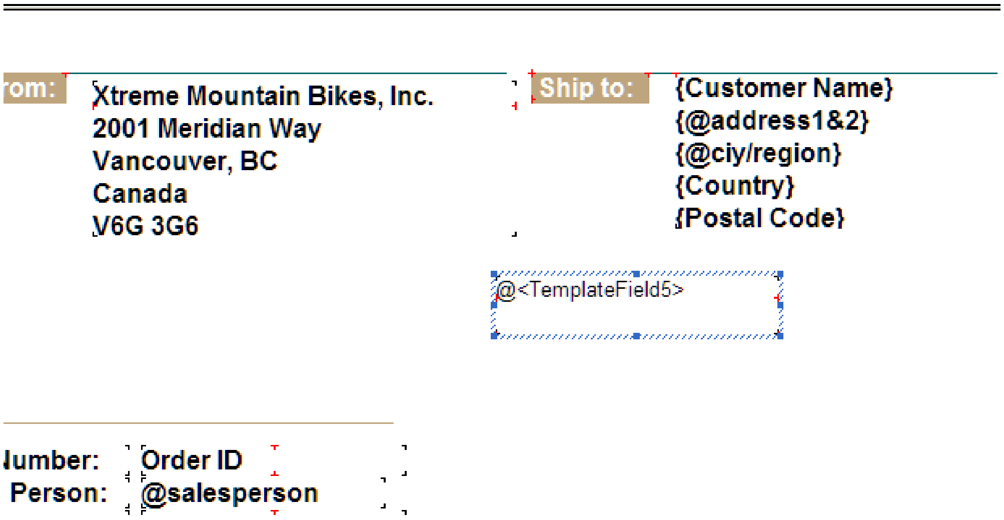
Switching the report to Design view, Right-clicking on the barcode and selecting Edit Field Object from the drop-down menu displays the Function (DL2DDUflBC) and parameters used to generate the barcode. The first parameter is the barcode data (a string), the second parameter is an integer that specifies the barcode type (code type 1 is Datamatrix); the third parameter may be used to specify the number of columns in the symbol, but may be 0 – in which case the size is automatically determined; the fourth parameter is for specifying a security level and may be 0 - in which case the security level is automatically determined; the fifth parameter is the barcode Mode (for Datamatrix that's 0 for Square or 1 for Rectangular), and the final parameter is an integer representing any special flags required for the barcode (usually 0).

Creating a 2D font barcode on a report

To create your own barcode on a report follow the steps below; the instructions provided for Crystal Reports 9 and 10, with the minor difference for other versions mentioned in brackets. These instructions are for the CRUFL; differences for the older UFL are described in the Appendix.

Step 1

From Insert menu choose Template Field Object; ensure that the field is wide enough to hold the entire barcode with clipping off the right hand side!



rom:

Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to:

{Customer Name}
{@address1&2}
{@ciy/region}
{Country}
{Postal Code}

@<TemplateField5>

lumber: Order ID
Person: @salesperson

[For Crystal Reports 8.5

1. Open the report in Design view

2. Select Field Object from the Insert menu. The Field Explorer opens.

3. Right click on Formula Fields in the Field Explorer and choose New from the pop-up menu that appears.

4. Create a suitable name for your formula field – such as barcode, then push the OK button.

5. The Formula Field Editor will open, but for now click on its "Save and Close" icon (top left).

6. The Field Explorer is still open. Select the new field object you have created and drag it to a suitable place on your report. Then Close the Field Explorer.

7. Right-click on the field you have placed on the report and select Edit Field Object from the pop-up menu. The Formula Editor appears.

8. From the list of Functions (normally the middle list) scroll down to the Additional Functions item; then expand this item by click on the + symbol alongside the Additional Items name.

9. Select the DL2DDUflBC function from the list, then either double-click on it or press the Enter key; the function then appears in the formula box below the lists, complete with its parameter brackets and commas, and the cursor in the position of the first parameter. ie.

DL2DDUflBC (,,,,,)

10. Then go to Step 5 below to complete the formula

]

Step 2

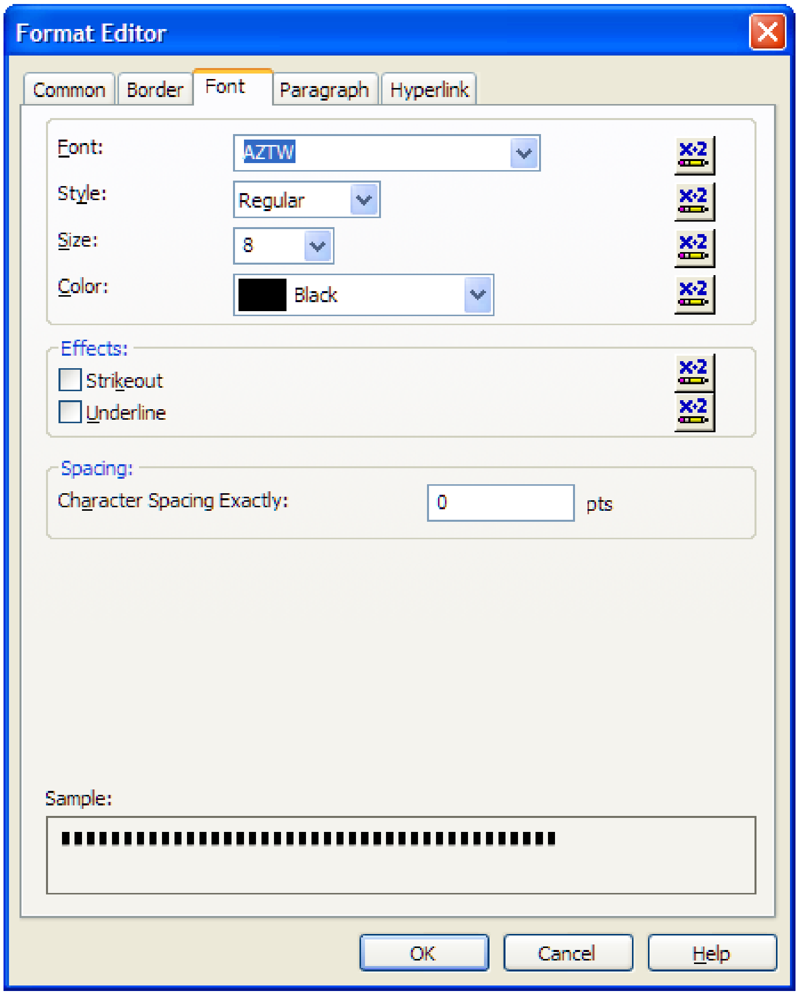
Right click on the object and choose Format Field.
Check the checkbox labeled Can Grow

Select the Font tab from the Format Editor dialog and select a suitable font, such as AZTW 8 point. Then push the OK button.

For Aztec, Datamatrix, DMRE, Micro QR Code or QR Code types select one of the AZT fonts.

For PDF417, Micro PDF417 or Truncated PDF417 select one of the PD2, PD3, PD4 or PD5 fonts (the number specifies the height/width ratio for the thinnest bar).

Note: you can specify smaller font sizes by entering the number into the size box.



Font: AZTW

Style: Regular

Size: 8

Color: Black

□ Strikeout

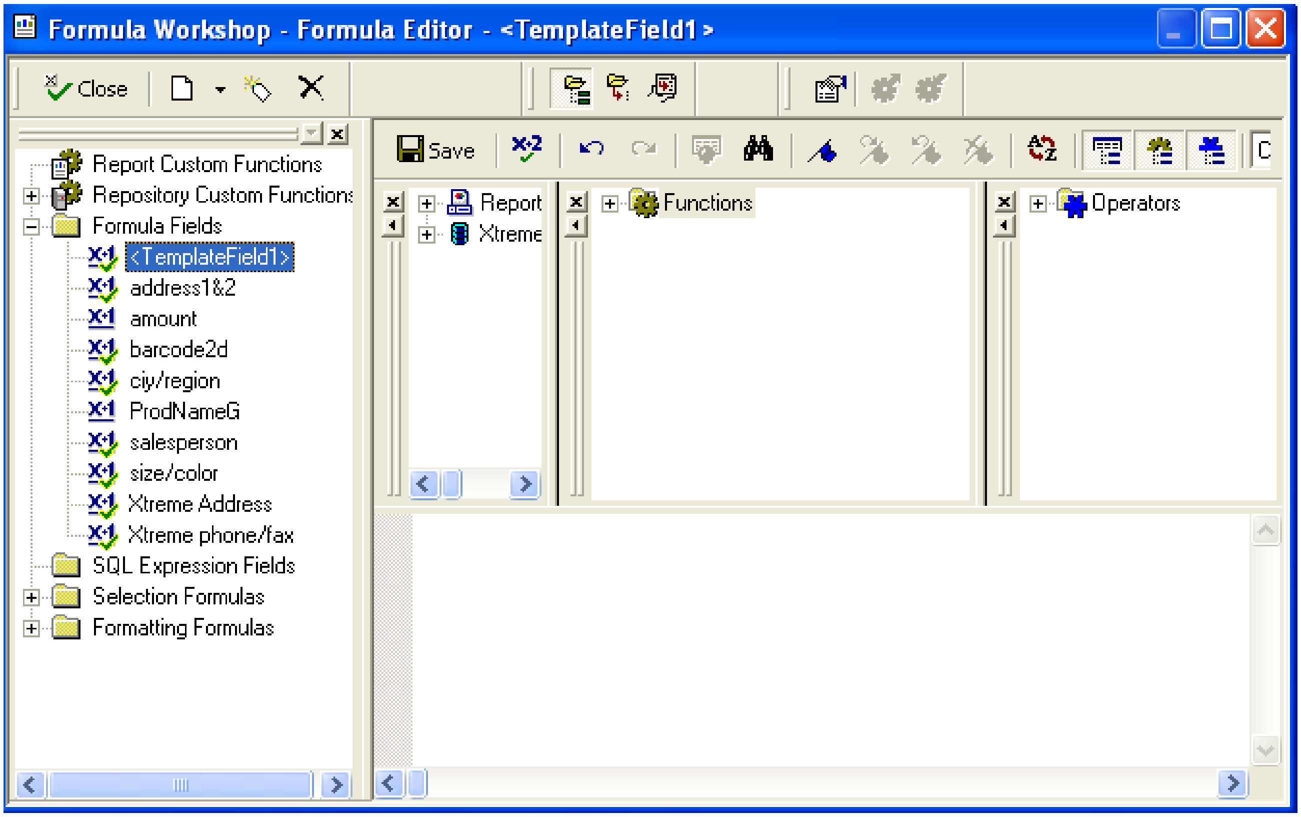
□ Underline

Character Spacing Exactly: 0 pts

Step 3

With the Field Object selected, from the Report menu select Formula Workshop.
Expand the Formula Fields list and click on the field you have inserted.

Note that you can rename the Formula field from, say, <TemplateField1> to something more useful, such as barcode2d, by right-clicking on the field name and choosing Rename from the pop-up menu.

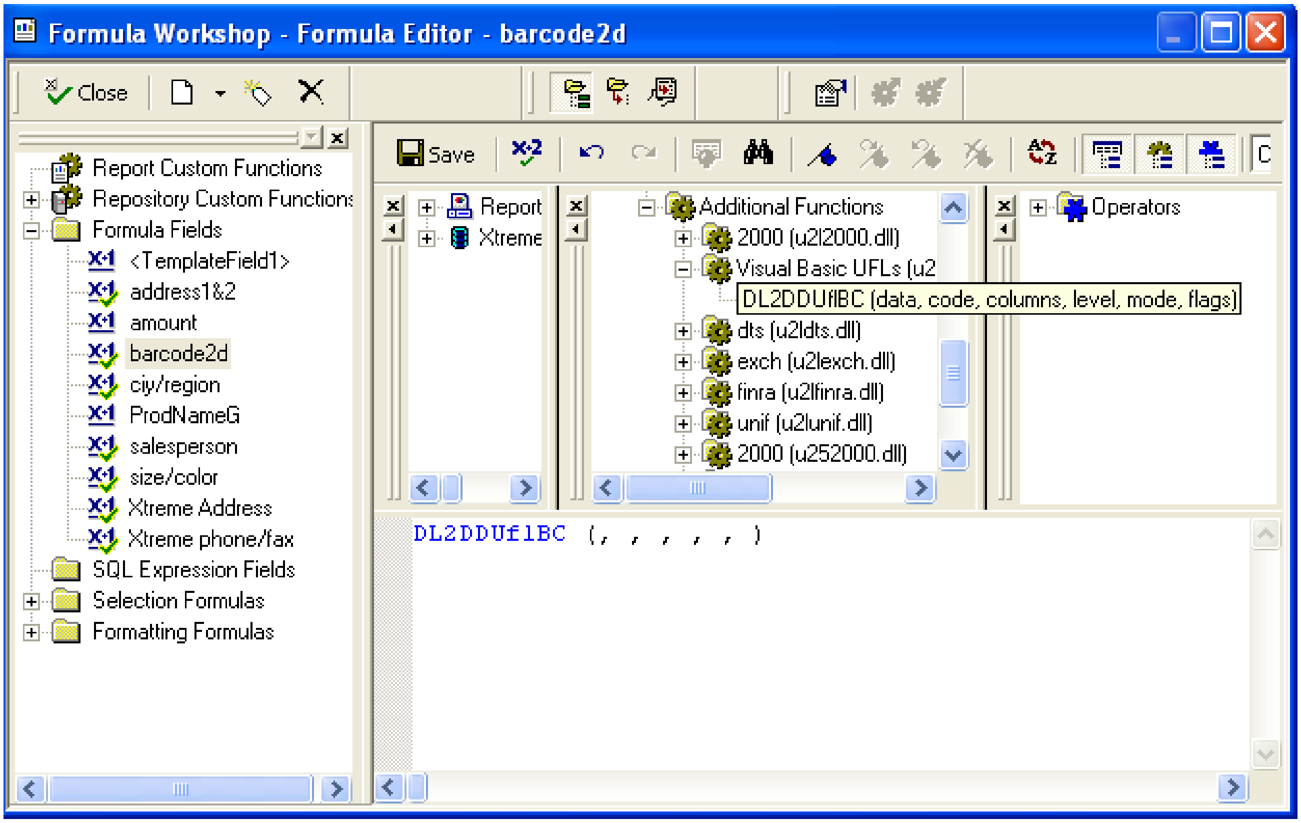


Step 4

Now to create the formula.

Expand the Functions list in the Formula Workshop. The User Function provided in Barcode 2D Tools is DL2DDUflBC. This will normally be located in the Additional Functions section of the Functions list. The precise location will depend on the version of Crystal Reports you are using. For version 9 – 2008 the function will appear under the “Visual Basic UFLs” item, but for later versions it will appear under “COM and .NET UFLs”.

Double-click on the DL2DDUflBC item and it will be placed in the formula box:

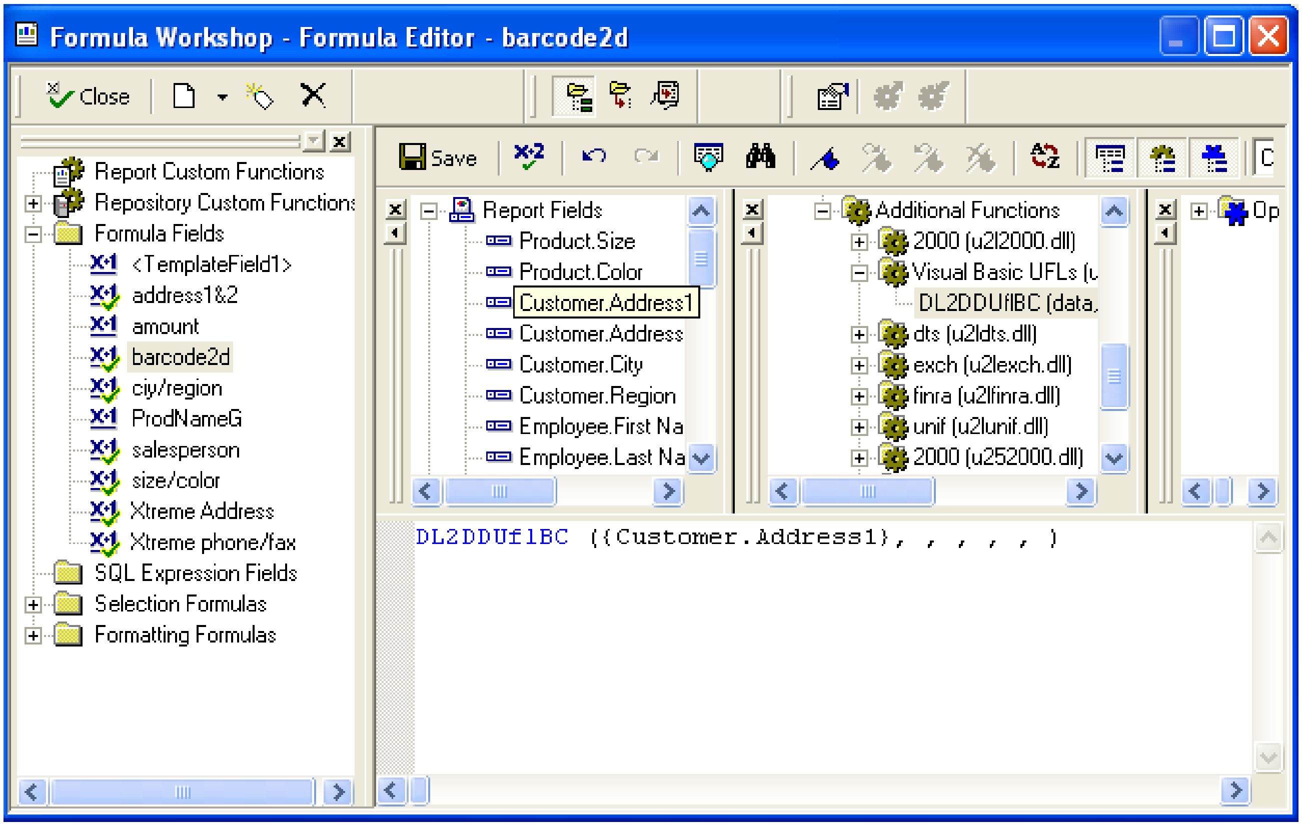


DL2DDUflBC (data, code, columns, level, mode, flags)

```
DL2DDUf1BC (, , , , , )
```

Step 5

To complete the formula we need to provide the parameters, the first of which is the data for the barcode. As an example we will take this from one of the Report Fields. So place the cursor in the position of the first parameter (ie before the first comma) then expand the list of Report Fields. Double-click on the required fieldname and the name is inserted into the formula.



```
DL2DDUf1BC ({Customer.Address1}, , , , , )
```

The data used for a barcode must be text. If a required field contains numeric data then expand the Strings item in the list of functions and the expand the ToText function; Now select the required function - which will usually be ToText(x) where the x represent the number to be used as data (note that there are several ToText functions which permit formatting of the number). Double click on the required version of the ToText function and this will be copied to the first parameter position of the DL2DDUflBC function, with the cursor now placed in the first parameter position of the ToText function, ie.

DL2DDUflBC (ToText (),,,,)

Now double click on the required data source field in the list of Report fields, eg.

DL2DDUflBC (ToText ({Orders.Order ID}),,,,)

To complete the formula we now add numbers representing the remaining parameters:

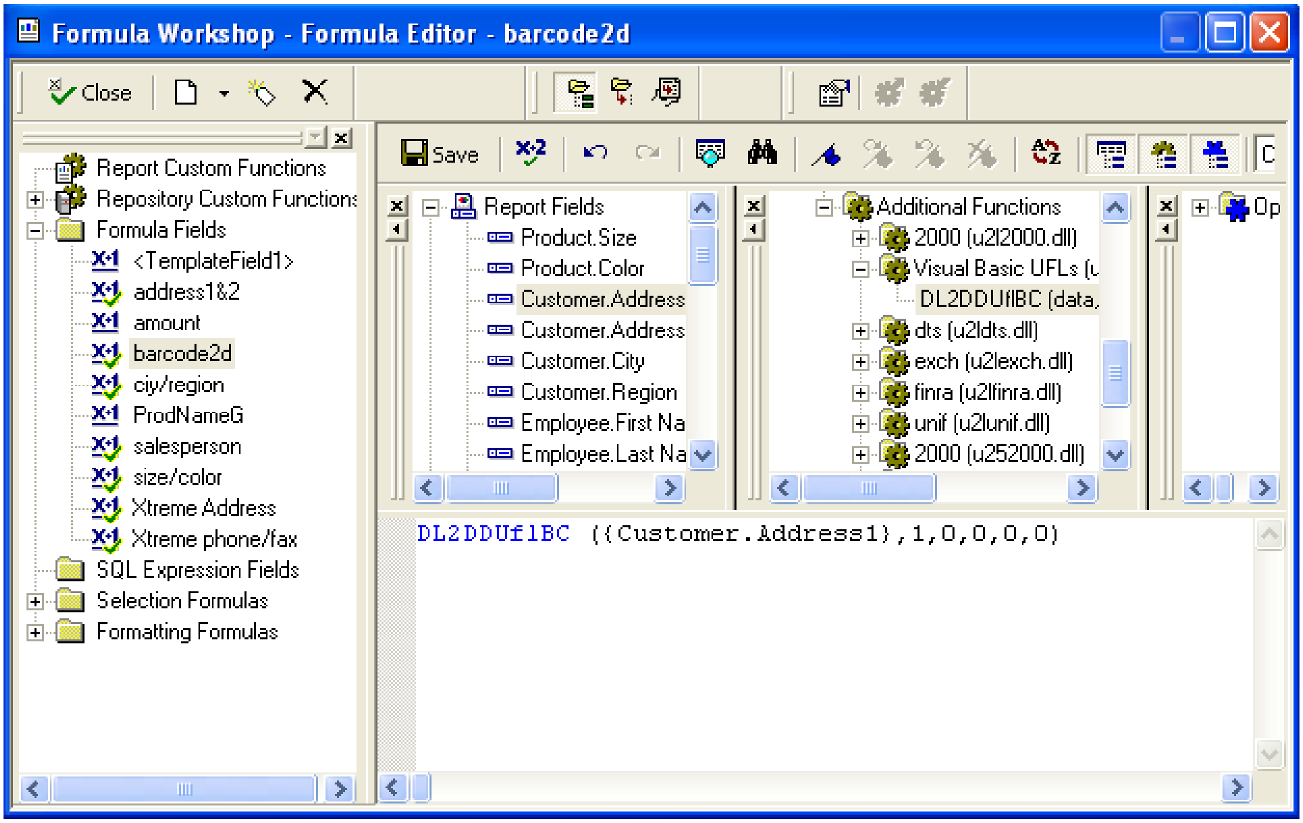
DL2DDUflBC (data, codetype, columns, security level, start mode, flags)

For illustration we will enter the remaining formula parameters as follows: the second parameter is an integer that specifies the barcode type (code type 1 is Datamatrix – see below); the third parameter may be used to specify the number of columns in the symbol, but may be 0 – in which case the size is automatically determined; the fourth parameter is for specifying a security level and may be 0 - in which case the security level is automatically determined; the fifth parameter is the barcode Mode (for Datamatrix that’s 0 for Automatic Square or 1 for Automatic Rectangular, etc. Full details are provided in the Barcodes Help file), and the final parameter is the flags parameter (normally 0). For example,

DL2DDUflBC ({Customer.Address1},1,0,0,0,0)

The flags parameter is normally 0, but for Datamatrix symbols a value of 1 causes GS1 encoding (with automatic AI handling) and a value of 4 produces the Deutschepost Postmatrix bars in front of the symbol.

To include non-ANSI characters the symbol must be encoded in Unicode, which can be accomplished by setting the flag to a value of 8.



```
DL2DDUf1BC ({Customer.Address1},1,0,0,0,0)
```

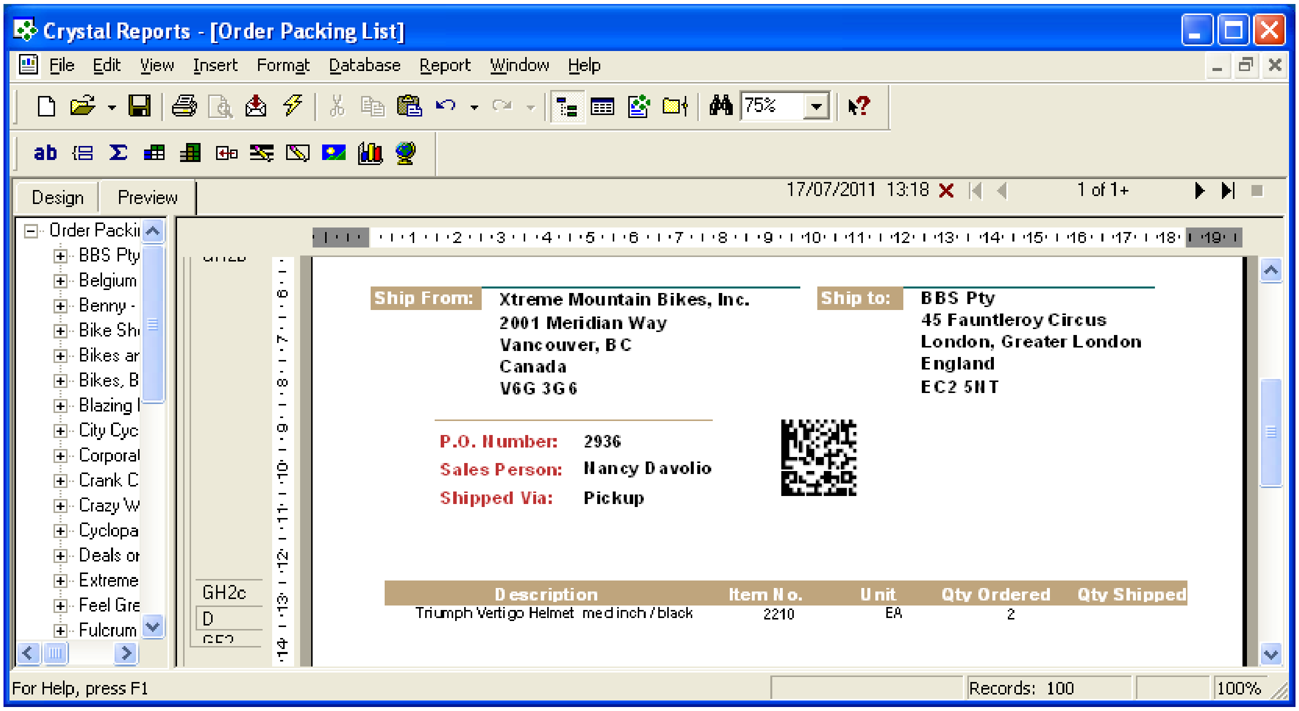
Step 6

Finally push the Save button, then the Close button (or just the Save and Close button in CR 10 or later).
Return to the report's Preview display and you should have a barcode. If you are using data from a database as the data source for the barcode, then the barcode will change as you navigate around the record source.

[For Crystal Reports 8.5

Complete Step 2 above to set the font used for showing the barcode.

]



Ship From: Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to: BBS Pty
45 Fauntleroy Circus
London, Greater London
England
EC2 5NT

P.O. Number: 2936
Sales Person: Nancy Davolio
Shipped Via: Pickup

Description	Item No.	Unit	Qty Ordered	Qty Shipped
Triumph Vertigo Helmet med inch / black	2210	EA	2	

To make changes to the barcode formula

Open the report in Design view, right click on the barcode (or its' empty box) and select Edit Formula from the drop down menu.

Follow the procedure above to replace the formula.

Note that invalid barcodes will not be visible.

Note that for font barcodes to appear as a barcode on reports viewed through the Report Application Server the font must be installed on the client computer. If the font is not available the barcode will appear either blank or as a series of meaningless characters.

Barcode types table for 2D Font barcodes

The barcode type parameter may be set to one of the following values in a formula:

Barcode Type	CodeTypeValue
Aztec	0
Datamatrix	1
PDF417	2
Micro_PDF	3
Truncated PDF417	4
QR Code	5
Micro QR Code	6

DMRE 8x48	7
DMRE 8x64	8
DMRE 12x64	9
DMRE 16x64	10
DMRE 24x32	11
DMRE 24x36	12
DMRE 24x48	13
DMRE 24x64	14
DMRE 26x32	15
DMRE 26x40	16
DMRE 26x48	17
DMRE 26x64	18

Appendix

Remote ASP.NET use

The IIS servlet and ASP.NET script supplied with Server/Developer versions may be installed on a remote server computer by copying the dlbc and/or dlbc2d folders of the Barcodes for Crystal Reports installation directory to a suitable location on the remote IIS server. The dlbc/dlbc2d folders on the remote server may be set as a web site or as applications running under an existing web site. If it is desirable to use a non-standard port number this may be done by setting the port number in the IIS Administrator properties and modifying all pointer in reports to use the port by placing the port number after a colon in the URL, e.g.

"http://www1.dlsoft.com:3001/dlbc/dlbc.aspx?CodeType=9&ImageHeight=12&Xunit=12&Data="+

The only additional system required on the IIS server is the .NET Framework version 2.0 or later (either the 32 or 64 bit version of the Framework may be used) for the ASP.NET script. The server does require the presence of Server Extensions on the web hosting the script, and the script must be assigned at least “Script only” execute permission.

The ASP.NET interface included with Single computer licensed versions may not be installed on a remote computer.

Legacy UFLs

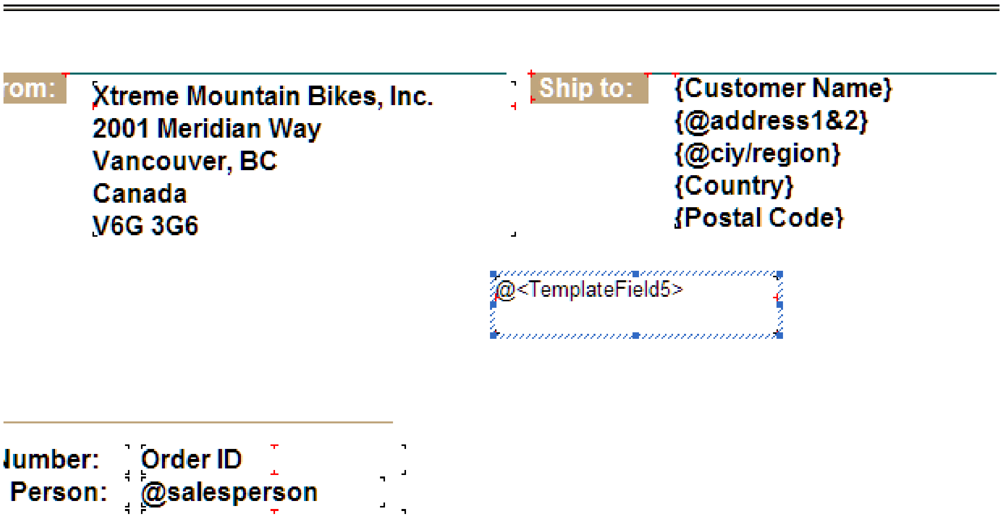
The old UFLs are provided for compatibility with previous versions of dLSoft Barcode products for Crystal Reports. For new implementation we recommend the use of the CRUFLs.

Creating a font barcode with the legacy 1D UFL

To create your own barcode on a report follow the steps below; the instructions are provided for creating a 1D barcode on Crystal Reports 9 and 10, with the minor difference for v 8.5 mentioned in brackets:

Step 1

From Insert menu choose Template Field Object; ensure that the field is wide enough to hold the entire barcode with clipping off the right hand side!



om:

Xtreme Mountain Bikes, Inc.
2001 Meridian Way
Vancouver, BC
Canada
V6G 3G6

Ship to:

{Customer Name}
{@address1&2}
{@ciy/region}
{Country}
{Postal Code}

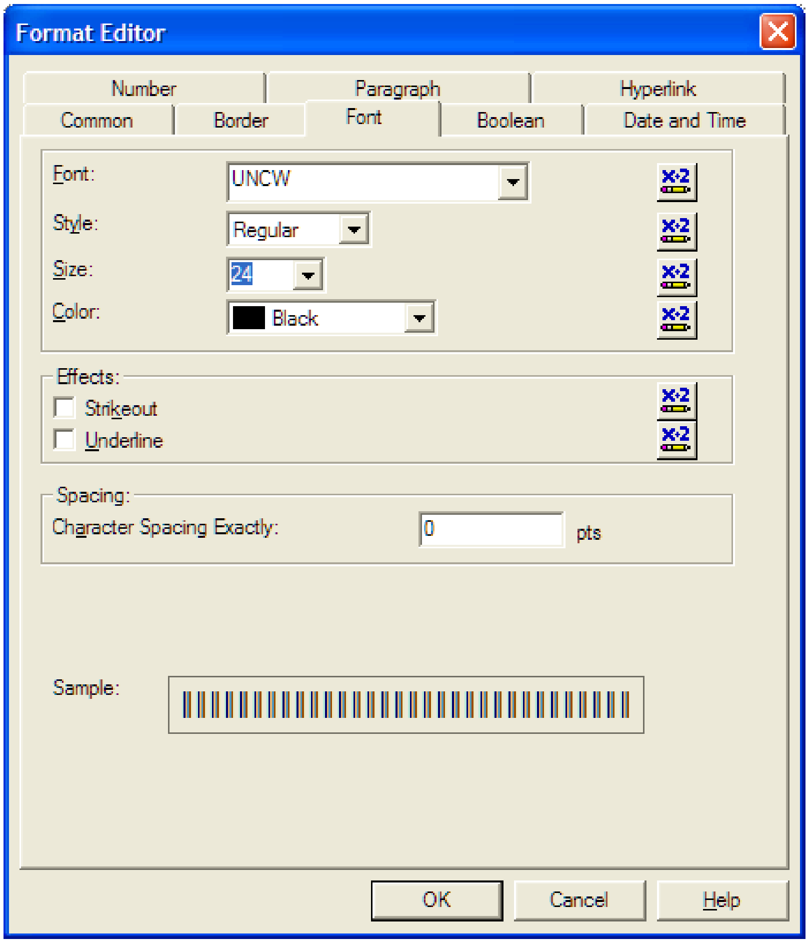
@<TemplateField5>

lumber: Order ID
Person: @salesperson

Step 2

Right click on the object and choose Format Template Field.
Check the checkbox labeled Can Grow

Select the Font tab from the Format Editor dialog and select a suitable font, such as UNCW 24 point. Then push the OK button



Font: UNCW

Style: Regular

Size: 24

Color: Black

Effects:

☐ Strikeout

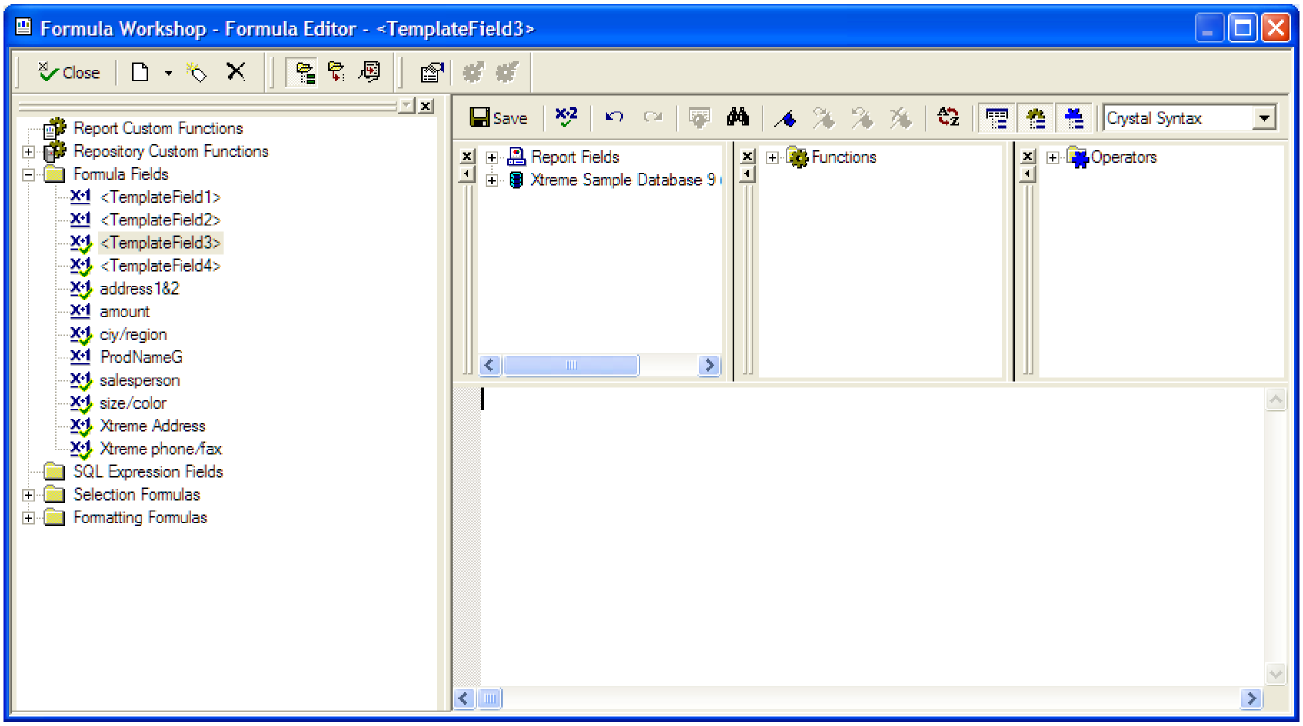
☐ Underline

Spacing:

Character Spacing Exactly: 0 pts

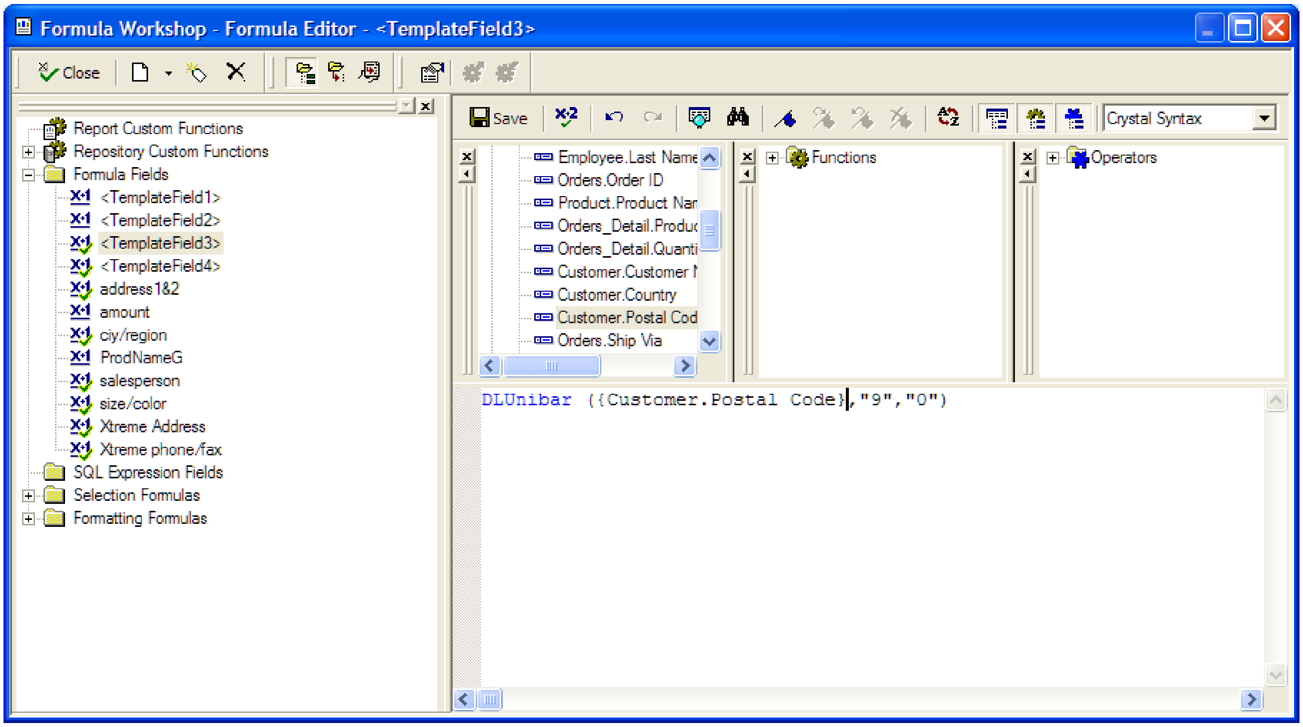
Step 3

With the Field Object selected, from the Report menu select Formula Workshop. Expand the Formula Fields list and double click on the field you have inserted



Step 4

Now enter the formula, such as
DLUnibar ({Customer.Postal Code},"9","0")



```
DLUnibar ({Customer.Postal Code},"9","0")
```

[For CR 8.5

1. Open the report in Design view

2. Select Field Object from the Insert menu. The Field Explorer opens.

3. Right click on Formula Fields in the Field Explorer and choose New from the pop-up menu that appears.

4. Create a suitable name for your formula filed – such as barcode, then push the OK button.

5. The Formula Field Editor will open, but for now click on its “Save and Close” icon (top left).

6. The Field Explorer is still open. Select the new field object you have created and drag it to a suitable place on your report. Then Close the Field Explorer.

7. Right-click on the field you have placed on the report and select Edit Field Object from the pop-up menu. The Formula Editor appears.

8. From the list of Functions (normally the middle list) scroll down to the Additional Functions item; then expand this item by click on the + symbol alongside the Additional Items name.

9. Select the DLUnibar function from the list, then either double-click on it or press the Enter key; the function then appears in the formula box below the lists, complete with its parameter brackets and commas, and the cursor in the position of the first parameter. ie.

DLUnibar (,,)

10. Complete the formula by changing it to
DLUnibar ({Customer.Postal Code},"9","0")

]

The first parameter in the DLUnibar function must be a string containing the barcode data. This can be a literal string (ie. data enclosed in quotation marks, such as "1234"), or field data. If field data is to be used it must be text data - so if the required field actually contains numeric data this must be converted into text data.

If a required data field contains text data just double click on the field name in the list of Report Fields, and the field name enclosed in curly brackets will be copied to the function's first parameter position. eg.

DLUnibar ({Customer.Region},,)

If a required field contains numeric data then expand the Strings item in the list of functions and the expand the ToText function; Now select the required function - which will usually be ToText(x,y,z) where the x represent the number to be used as data, y represent the number of decimal places (typically 0) and z is a character used to separate thousands from hundreds etc (which unfortunately defaults to a comma and is generally not wanted in a barcode). Double click on the required version of the ToText function and this will be copied to the first parameter position of the DLUnibar function, with the cursor now placed in the first parameter position of the ToText function, ie.

DLUnibar (ToText (, ,),,)

Now double click on the required data source field in the list of Report fields, eg.

DLUnibar (ToText ({Orders.Order ID}, ,),,)

and fill in the other two ToText parameters with a 0 (the number of decimal places) and a NULL character (two single quotes) respectively, ie.

DLUnibar (ToText ({Orders.Order ID},0 ,''),,)

The second parameter of the DLUnibar function represents the type of the barcode to be created. This value (shown in the Barcode Types Table in the UFL Reference) must be enclosed in double quotation marks.

The final parameter is a flags parameter (normally a 0), also within quotation marks eg.

DLUnibar (ToText ({Orders.Order ID},0 ,' '),"8","0")

The Flags parameter is a number made up by adding values for the following barcode properties:

Property	*value*
Automatic check digit	1
Wide bars	2
GS1 flag	8 (using this flag requires the use of one of the DxxE EANUPC fonts
Extra 1	16
Extra 2	32

The uses of the Extra 1 and Extra 2 parameters are described in the Barcodes Help file for individual barcode types.

Step 5

Finally push the Save button, then the Close button (or just the Save and Close button in CR 10).
Return to the report's Preview display and you should have a barcode. If you are using data from a database as the data source for the barcode, then the barcode will change as you navigate around the record source.

Note that font barcodes do not include human readable text under the bars. If you wish to include human readable data you should add a text field under the barcode using data from the same source as the barcode.

To make changes to the barcode formula

Open the report in Design view, right click on the barcode (or its' empty box) and select Edit Formula [Edit Field Object in v 8.5] from the drop down menu.

Follow the procedure above to replace the formula.

Note that invalid barcodes will not be visible. Some barcode types support only digits, other support only digits and upper case letters. Barcode that support only digits do not support spaces!!

Note that for font barcodes to be visible on reports viewed through the Report Application Server the font must be installed on the client computer.

DLUnibarh()

The DLUnibarh() function takes the same parameters as the DLUnibar() function, but returns the human readable text form of the barcode – including check digit if the Flags parameter contains 1. This is useful for obtaining the human readable form complete with check digit when the available data does not contain a check digit.

The 1D UFL parameters

The User Function Library generates a string of characters from data, and when those characters are displayed in the correct font they appear as a barcode. The UFL is called from a Template Field Object placed on a report when its formula has been specified in the Formula Workshop.

A typical formula would be,

DLUnibar (ToText ({Orders.Order ID},0 ,' '),"8","0")

The formula has three parameters.

The first is the source of the text data for the barcode. If the source actually contains numeric data then the Crystal Reports ToText() function must be used to convert the data into text (as illustrated above).

The second parameter is a text version of the barcode type number. The available barcode types are shown in the table below.

The final parameter is the Flags parameter, which is a text version of a number made up by adding values for the following barcode properties:

Property	*value*
Automatic check digit	1
Wide bars	2
GS1 flag	8 (using this flag requires the use of one of the DxxE EANUPC fonts
Extra 1	16
Extra 2	32

The uses of the Extra 1 and Extra 2 parameters are described in the Barcodes Help file for individual barcode types.

Barcode Types Table for the 1D UFL

code#	Code type	no. of characters	check digit
0	EAN13	13 numbers	1
1	EAN-8	8 numbers	
2	EAN13+2	15 numbers	1
3	EAN13+5	18 numbers	1
4	UPC-A	12 numbers	1
5	UPC-E	7 numbers	1
6	ITF-14	14 numbers	1 (EAN optional)
7	ITF-6	6 numbers	
8	Code 39	any	1 optional
9	Code 128	any*	automatic
10	GS1-128	any*	automatic
11	2 of 5	any numbers	
12	Interleaved-2 of 5	number pairs	1 optional
13	3 of 9	any	
14	Code B	any numbers	
15	Code 11	any	1 or 2
16	Codabar	any	
17	MSI	any numbers	1 or 2
18	Ext. Code 39	any (full ASCII)	1 optional
19	UPCA+2	14 numbers	1
20	UPCA+5	17 numbers	1
21	EAN8+2	10 numbers	1
22	EAN8+5	13 numbers	1
23	UPCE+2	9 numbers	1
24	UPCE+5	12 numbers	1
25	Telepen numeric	any	1 optional
26	Telepen ASCII	any	1 optional
27	Telepen begin numeric end	any	1 optional
28	PostNet type A	5 numbers	1
29	PostNet type C	9 numbers	1
30	PostNet type C'	11 numbers	1

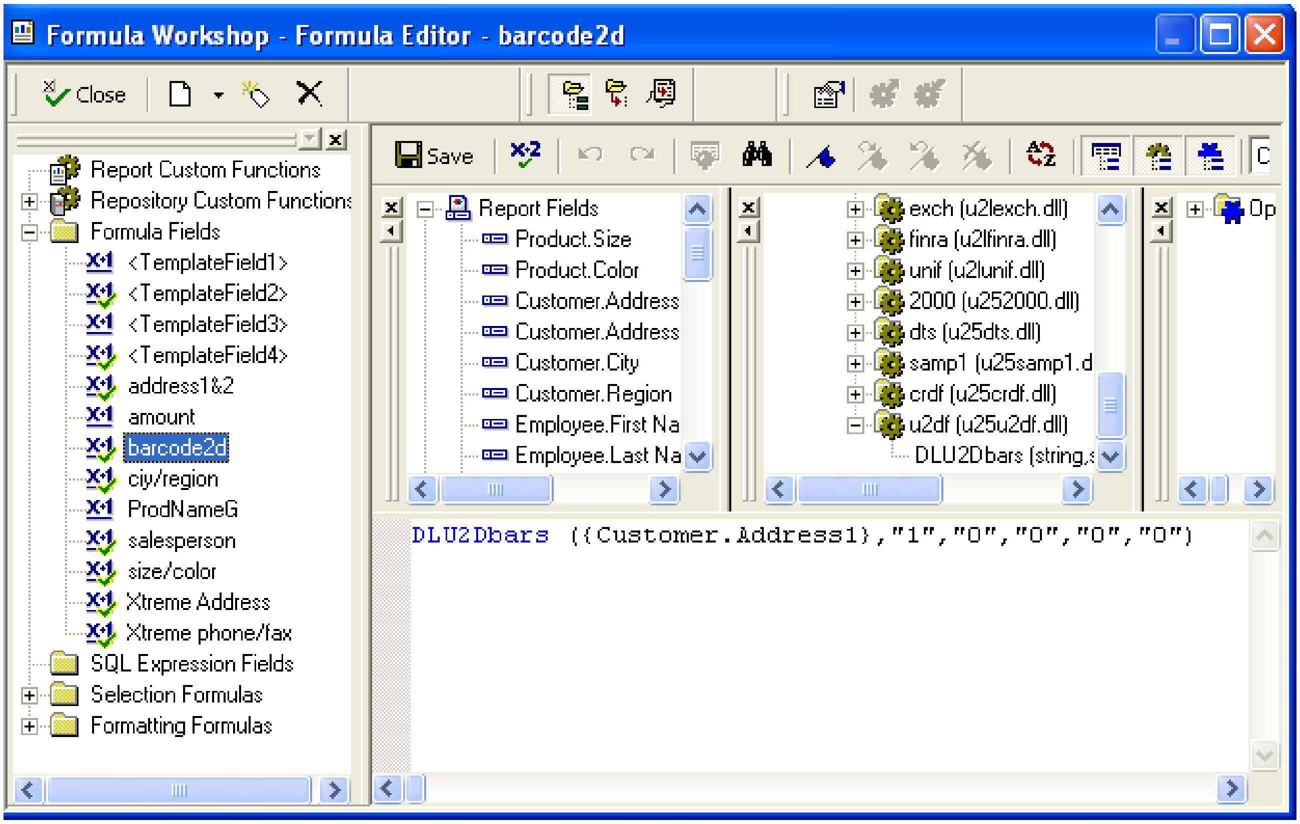
31	FIM A	fixed code	0
32	FIM B	fixed code	0
33	FIM C	fixed code	0
34	RM4SCC	any	1
35	4-State	any	1 optional
36	Code 93	any	2 optional
37	Ex Code 93	any (full ASCII)	2 optional
38	ISBN	9/10 digit ISBN	1 automatic
39	Matrix 2/5	numeric	1 optional
40	Plessey	numeric/some alpha	2
41	Australia Post	numeric	automatic
42	Swiss Post	fixed code	0
43	Deutsche Post	numeric	
44	SISAC	SICI codes	1
45	EAN-14	13	1
46	Planet Origin	9	1 automatic
47	Planet Destination	9	1 automatic
48	ISSN	9/10 digit ISSN	1 automatic
49	ISMN	8/9 digit ISMN	1 automatic
50	SSCC	17	1
51	Korean Postal Authority	6	1
52	Italian Postal 3/9	2+8+c+2	1
53	Italian Postal 2/5	12	1 automatic
54	ISBN+2 digit	Any / 2	1 automatic
55	ISBN+5 digit	Any / 5	1 automatic
56	ISSN+2 digit	Any / 2	1 automatic
57	ISSN+5 digit	Any / 5	1 automatic
58	Japan Post	Any	1 automatic
59	IATA 2 of 5	Any numeric	
60	China Postal Code	Any numeric	1 optional
61	OneCode	numeric	0
62	Code 128 Subtype A		1 automatic
63	Code 128 Subtype B		1 automatic
64	UPC-E0	12 numeric	1
65	UPC-E1	12 numeric	1
66	InfoMail Barcode A	numeric	0
67	PZN	numeric	1 automatic
68	Databar Omnidirectional	13 numeric	0
69	Databar Truncated	13 numeric	0
70	Databar Limited	13 numeric	0
71	Databar Expanded	Any	0
72	HIBC Code 39	various	1

73	HIBC Code 128	various	1
74	IM^{TM} Package Barcode	numeric	1
75	FIM D	Fixed code	0

Using the older 2D UFL

The U25U2DF library is installed automatically and may be used in all versions of Crystal Reports from 8.5. The size of the barcode generated by this library is restricted to 254 characters, which limits the amount of data the barcode can hold to around 70-80 characters. This library is included in for compatibility with previous versions of dLSoft Barcode products for Crystal Reports.

The UFL formula is created in the same way as described above for the CRUFL, except that the function appears in the Formula list as DLU2Dbars() under the "u2df (u25u2df.dll)" item of the Additional Functions list.



```
DLU2Dbars ({Customer.Address1},"1","0","0","0","0")
```

The function is:

DLU2Dbars (data, codetype, columns, security level, start mode, flags)

And in this case all parameters must be strings – so the values for codetype, etc. must be placed within quotation marks.

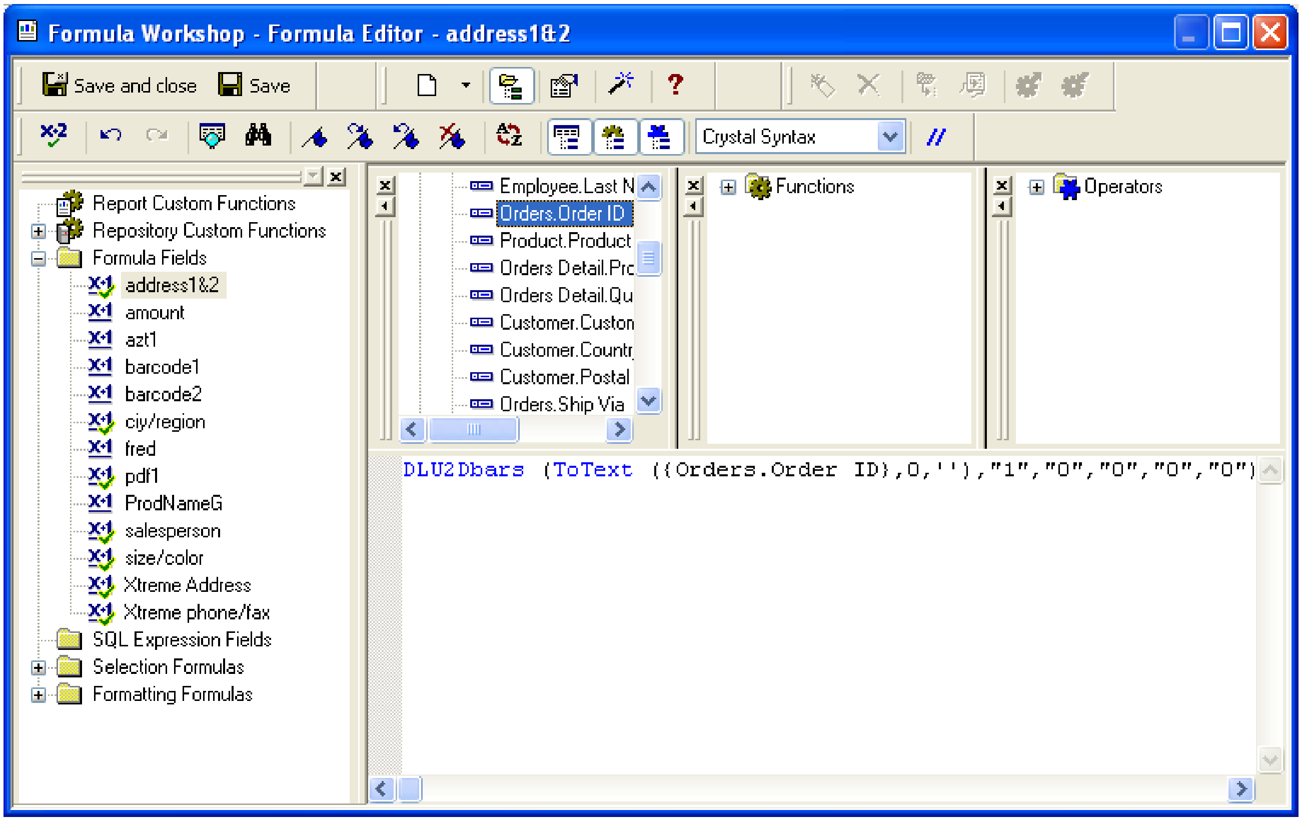
The first parameter is the barcode data (a string), the second parameter is an integer that specifies the barcode type (code type "1" is Datamatrix); the third parameter may be used to specify the number of columns in the symbol, but may be "0" – in which case the size is automatically determined; the fourth parameter is for specifying a security level and may be "0" - in which case the security level is automatically determined; and the fifth parameter is the barcode Mode (for Datamatrix that's 0 for Automatic Square or 1 for Automatic Rectangular, etc. Full details are provided in the Barcodes Help file), and the final parameter is the flags parameter (normally 0). For example:

DLU2Dbars ({Customer.Address1},"1","0","0","0","0")

The flags parameter may be made up by summing the following features:

For Datamatrix codes: GS1 encoding =2; for Postmatrix bars = 64; for DeutschePost encoding = 512.

For Aztec, Datamatrix, PDF417 and QR Code: Unicode encoding = 128.



```
DLU2Dbars (ToText ({Orders.Order ID},0,''),"1","0","0","0","0")
```

Note that invalid barcodes will not be visible.

Note that for font barcodes to be visible on reports viewed through the Report Application Server the font must be installed on the client computer.

Sample reports for font barcodes using the UFL are Order Packing UFL2D.rpt in the installation folder. [For Crystal Reports version 8.5 the sample is Order Packing UFL2D85.rpt in the cr85 folder of the installation folder]

Legacy 1D OLE server

The OLE server is used by the Report Designer Component in many versions of Crystal Reports. The server is dBarcodeCR.dll which is installed in the system folder of the Barcodes for Crystal Reports installation directory. The OLE server is self-registered on installation. If the dll is moved it must be re-registered with the machine's OLE system by running

Regsvr32 full_path\dBarcodeCR.dll

From a command prompt, where full_path is the full path to the directory containing the dll.

The image created by the OLE image creator may be obtained its Picture property.

Picture property

The Picture property returns an OLE picture which is generally used as the source of a picture in a Visual Basic PictureBox.

In this case the physical size of the picture must be obtained from the OLE image creator by retrieving the PictureWidth and PictureHeight properties once the image has been created. These values are reported in Windows HIMETRIC units, and so require scaling to the units used by the form containing the PictureBox – which in Crystal Reports is TWIPS (twentieths of a point per inch!).

The Picture property may be placed in a VB PictureBox using

```
Set PictureBox1.FormattedPicture = M_barcode.Picture
PictureBox1.Height = M_barcode.PictureHeight / twips
PictureBox1.Width = M_barcode.PictureWidth / twips
```

Where twips is the scaling factor between HIMETRIC and TWIPS units.

The Picture property is used in this way to generate barcode images in applications created using older versions of the Report Designer Component for VB6. [For the RDC of Crystal Reports 11 the use of the Pic() method is to be preferred]

Crystal Reports 1D Designer for VB6

The Crystal Reports Designer Component (RDC) was included with Crystal Reports 8, 8.5 and 9, and allows reports to be incorporated into programs that can (among other things) display the report. The RDC allows images to be created dynamically, and Barcodes for Crystal Reports includes an OLE server that can use this facility to create barcode images when a report record is displayed.

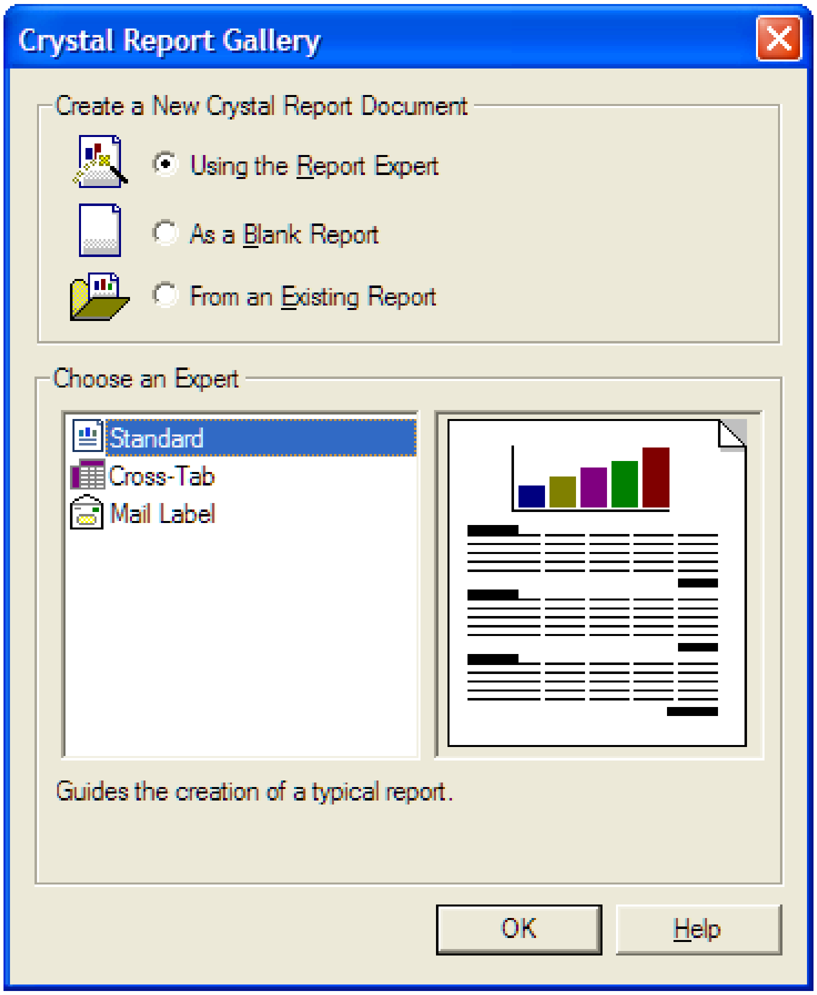
To create an application containing dynamically generated barcodes follow the steps below:

Step 1.

Open Visual Basic and create a new Windows Application (such as Standard Exe).

In the Project Explorer Window right click on the project title and choose Add

Select Crystal Report 9 [or 8.5] from the list of options. Select an existing report or create a blank report and connect to some data.



Crystal Report Gallery

Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

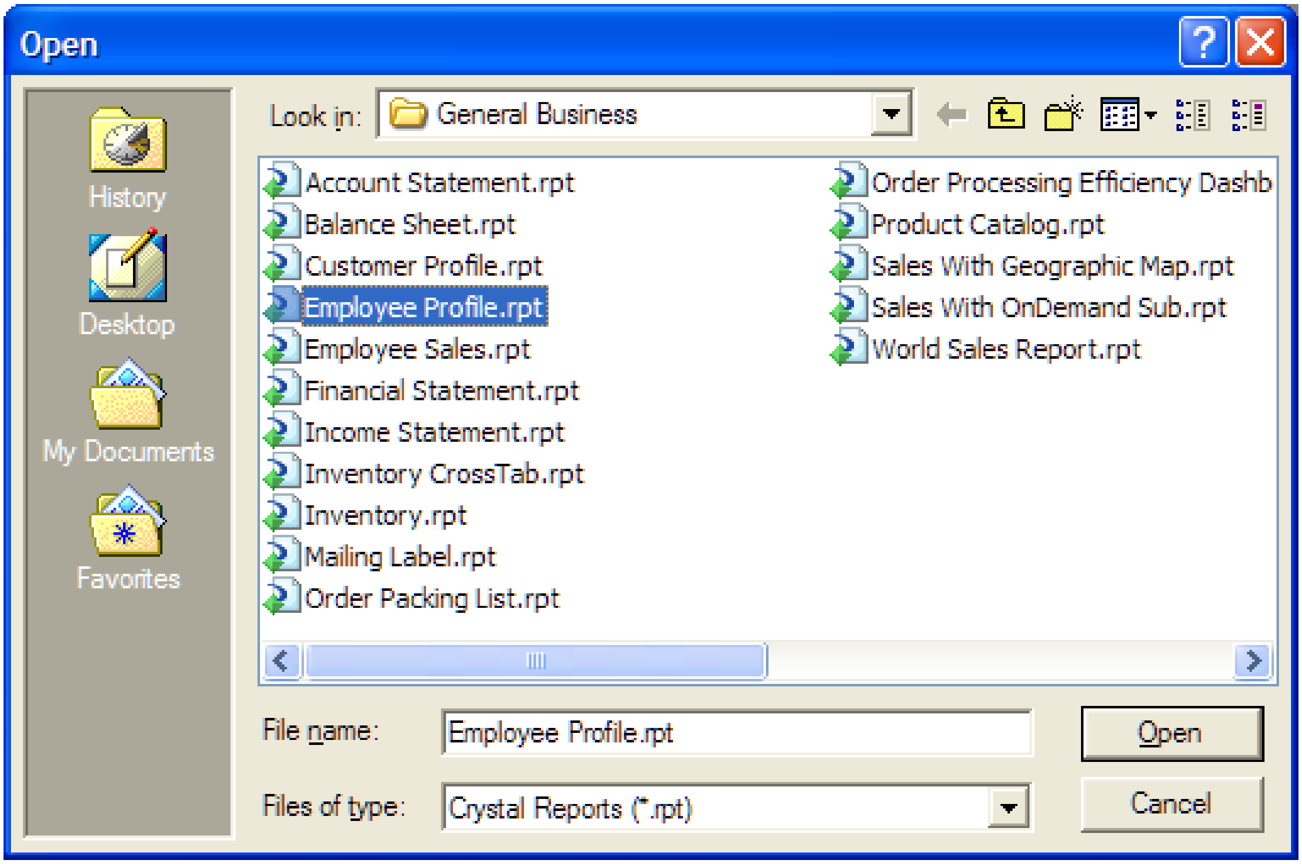
○ From an Existing Report

Choose an Expert

- Standard
- Cross-Tab
- Mail Label

Guides the creation of a typical report.

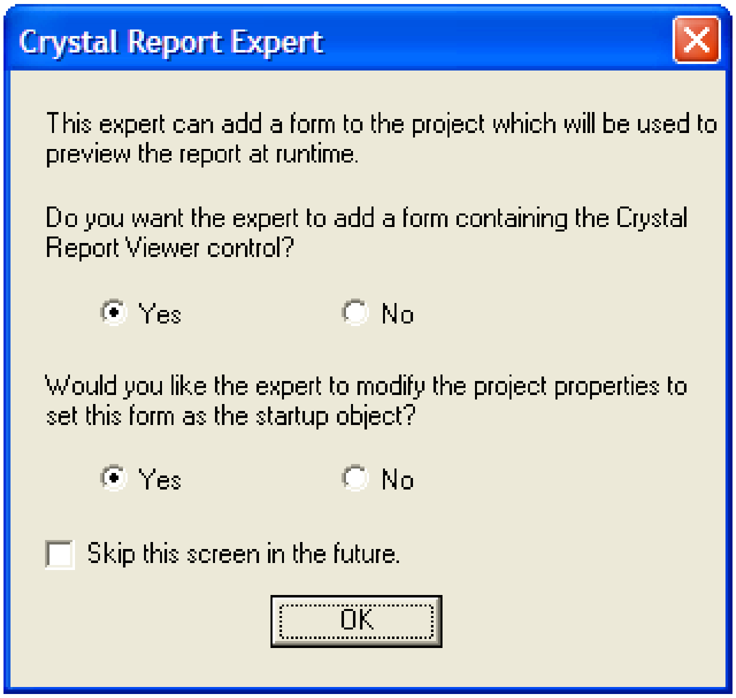
If you wish to add an existing report using the Report Expert you can navigate to the report, eg



File name: Employee Profile.rpt

Files of type: Crystal Reports (*.rpt)

And elect to have the expert create a form containing the layout of the report



This expert can add a form to the project which will be used to preview the report at runtime.

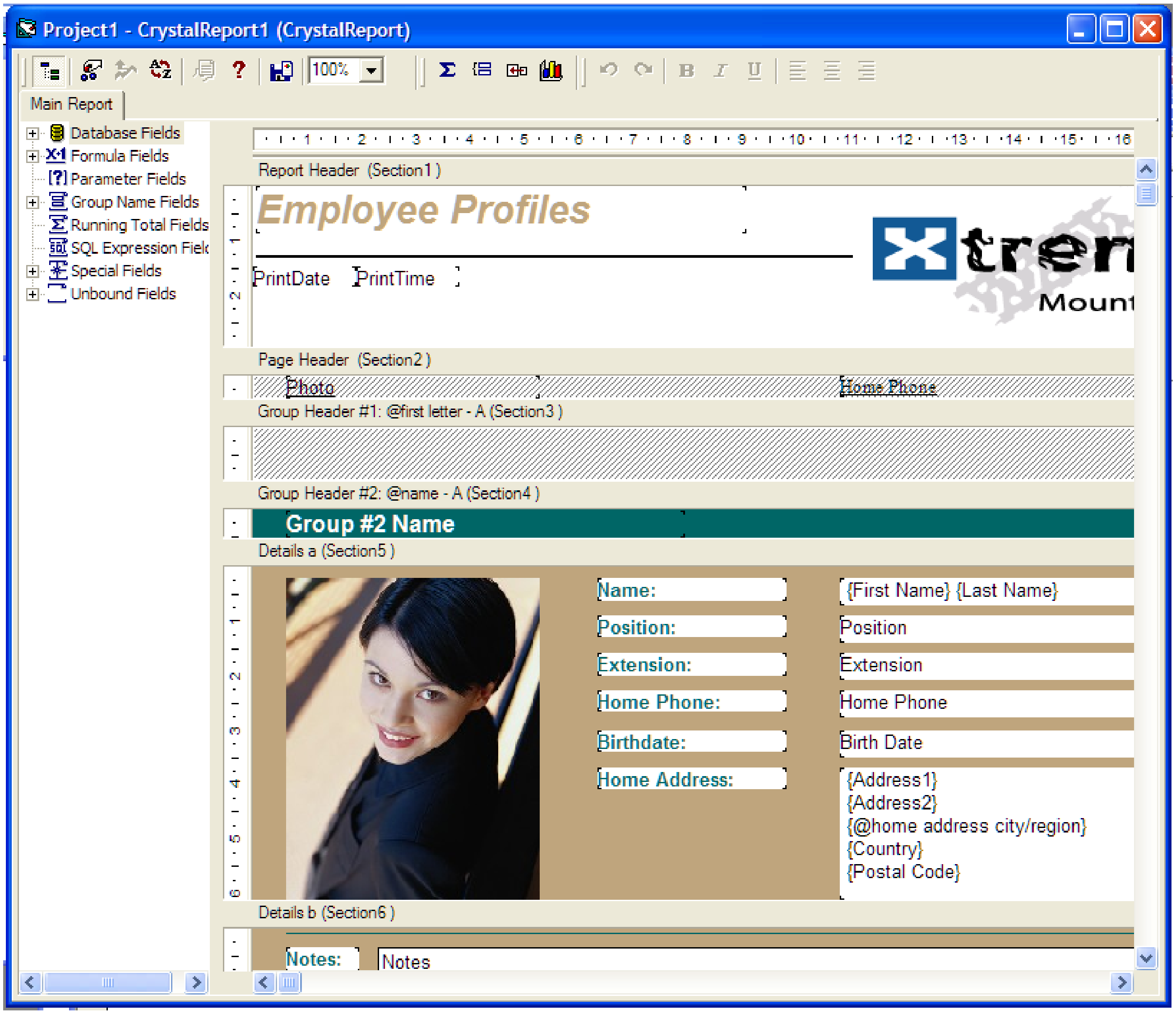
Do you want the expert to add a form containing the Crystal Report Viewer control?

● Yes ○ No

Would you like the expert to modify the project properties to set this form as the startup object?

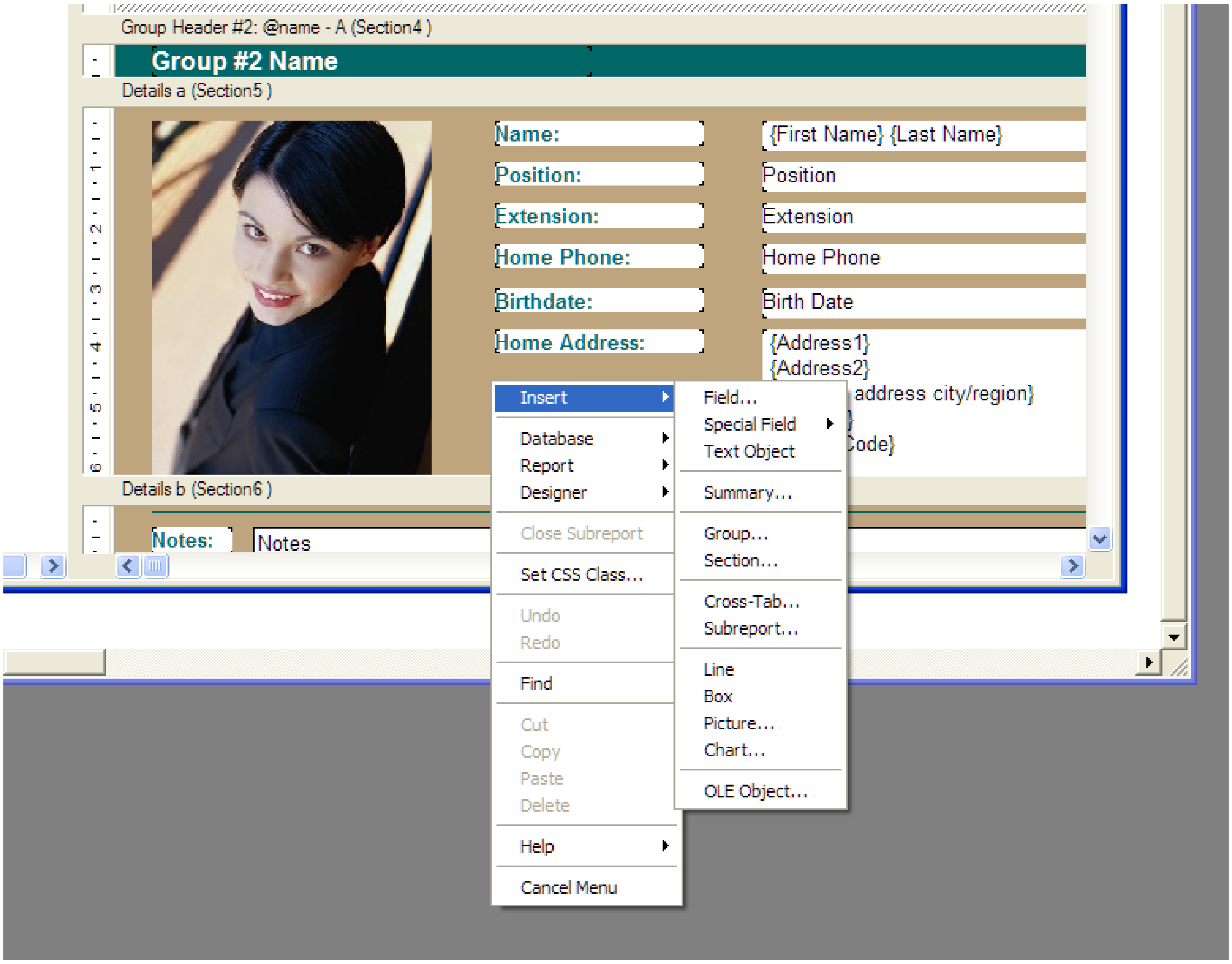
● Yes ○ No

□ Skip this screen in the future.

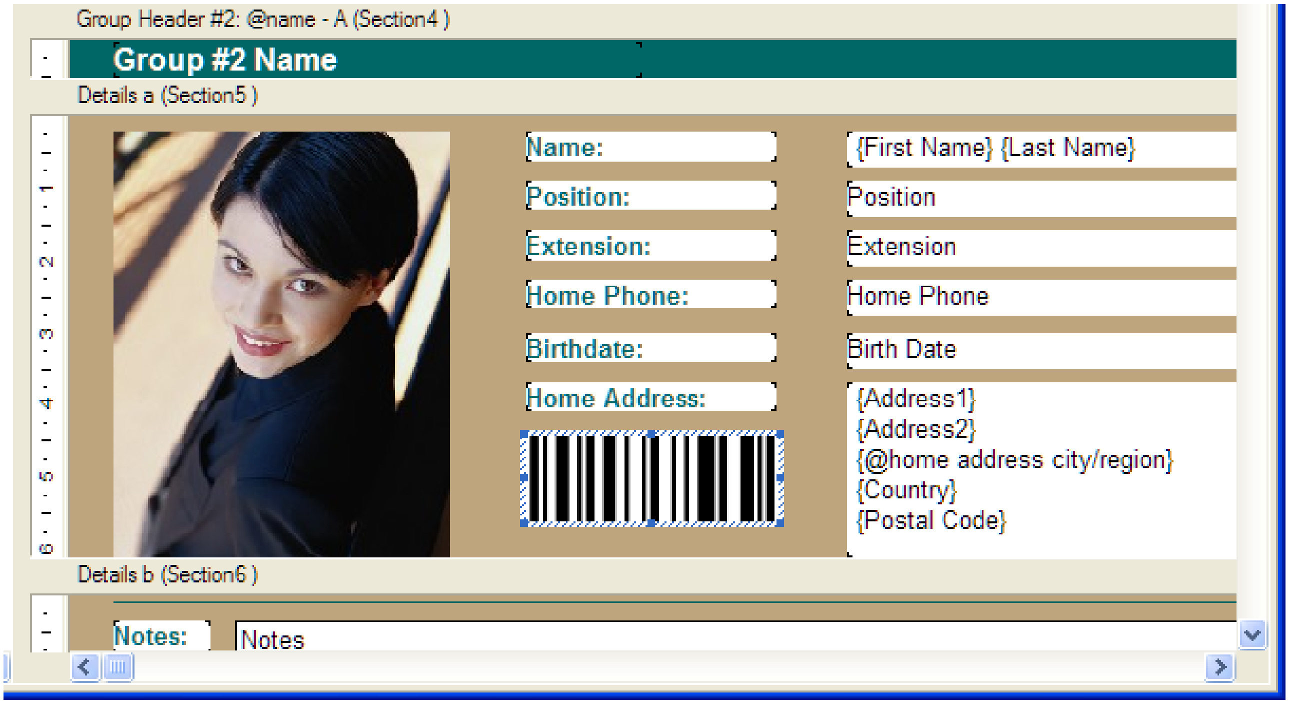


Step 2

Right-click on the area of the report's section at which you wish to place a barcode image and select Insert and Picture from the pop-up menus displayed

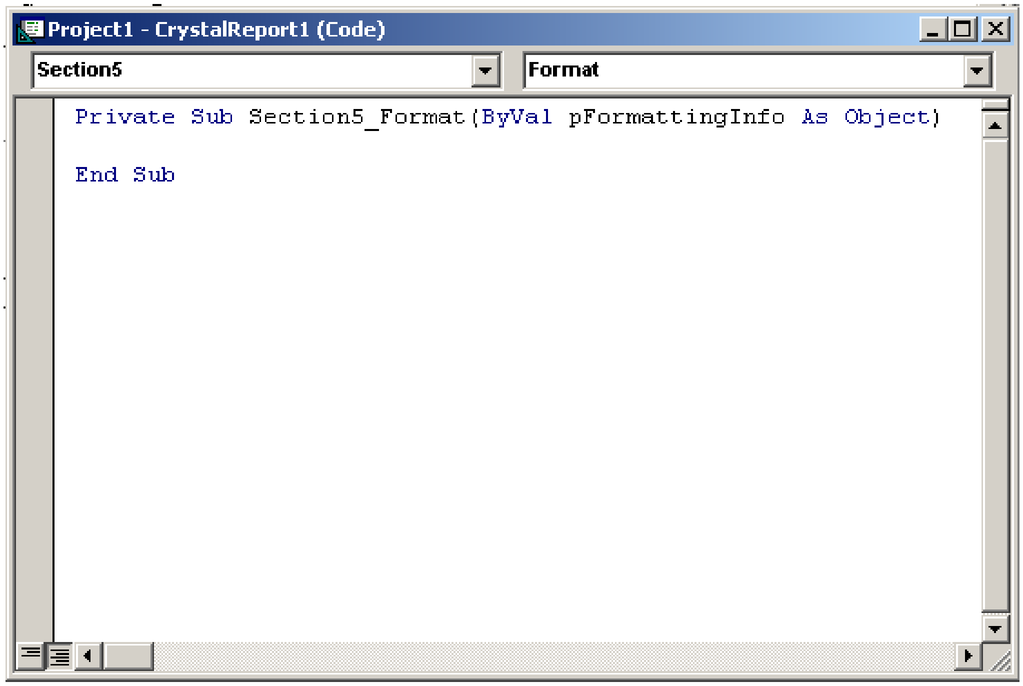


When the dialog appears for you to specify the source of the picture navigate to the Barcodes for Crystal Reports installation directory and choose the dummybarcode image. A barcode placeholder will then be displayed on the report.



Step 3

Double-click on the barcode image and a code window will open.



```
Private Sub Section5_Format(ByVal pFormattingInfo As Object)

End Sub
```

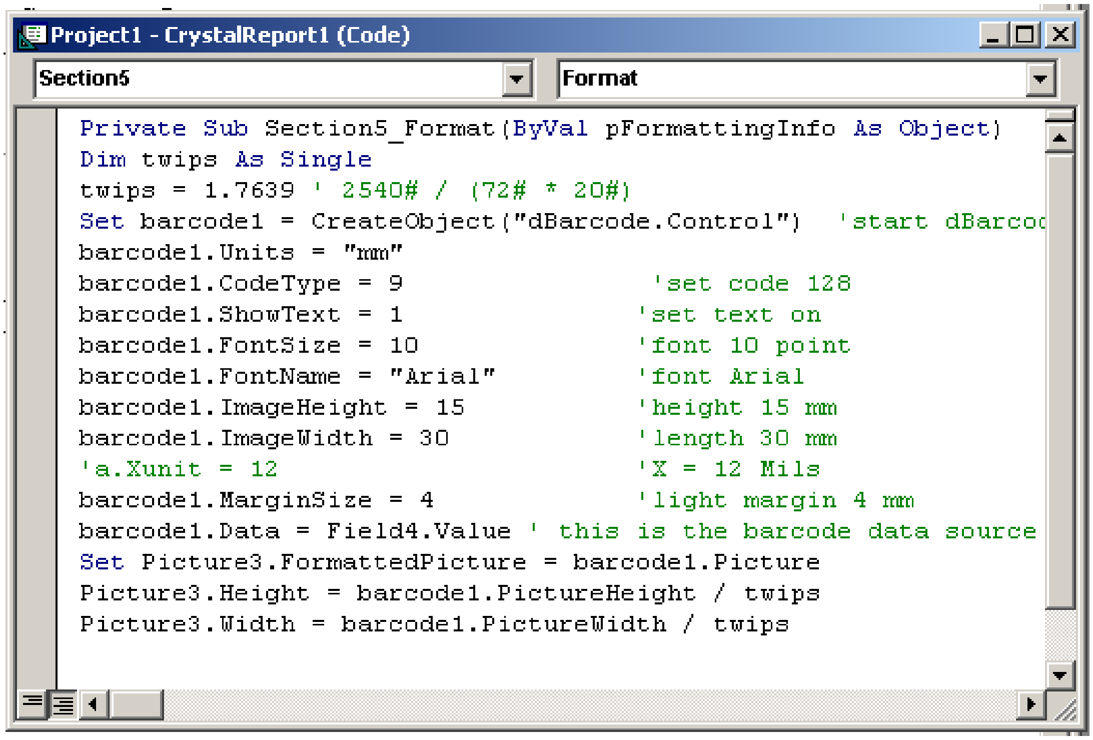
This will contain the code which is run each time the section you double-clicked on is “formatted”, i.e. prepared for display.

Step 4

Enter the code required to create the barcode from data in your chosen source field.

The code required here will frequently be very similar to that shown below, in fact if you have a field named Field5 on the report you can copy and paste this code directly into code window.

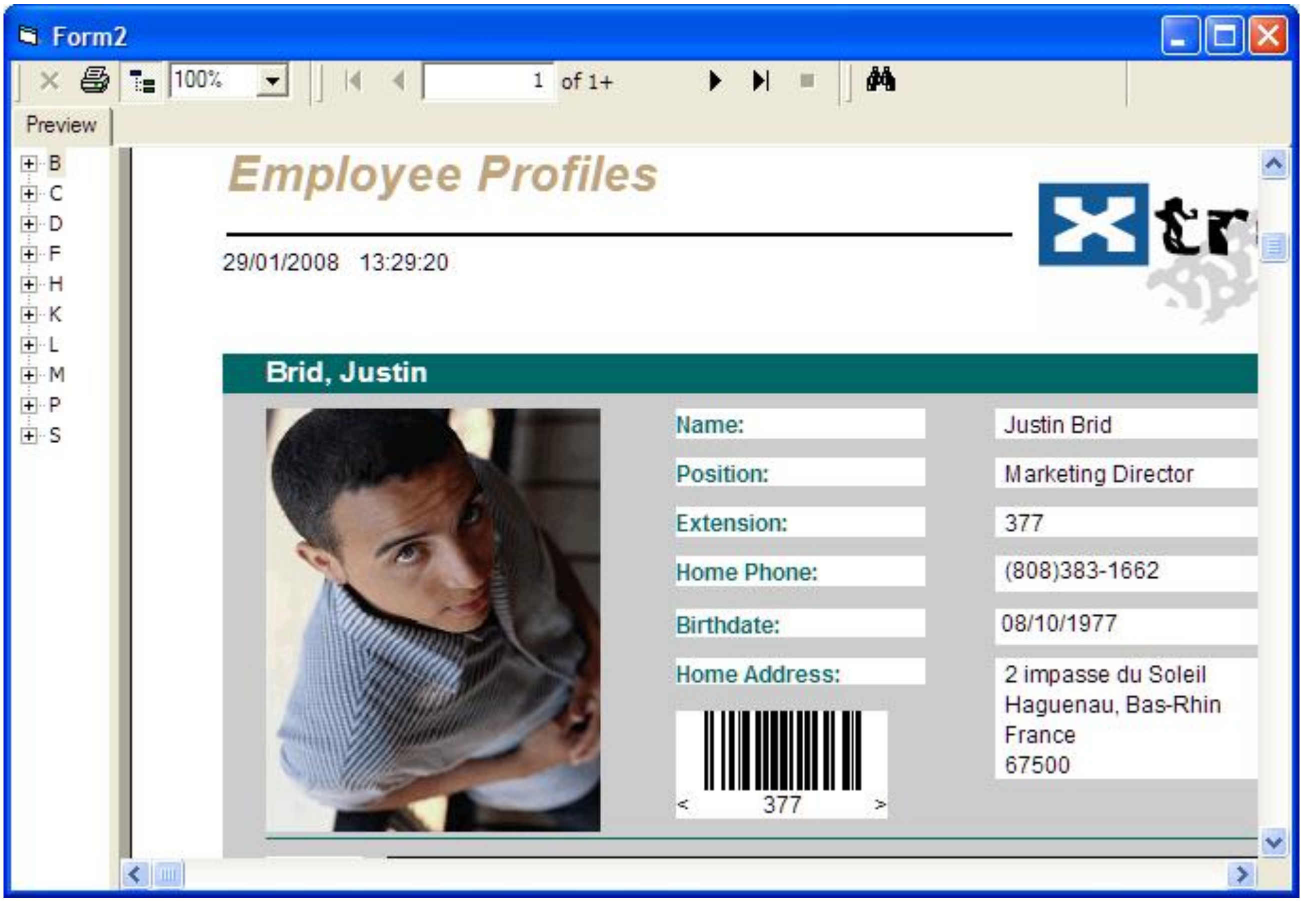
```
Private Sub scnDetail_Format(ByVal pFormattingInfo As Object)
Dim twips As Single
twips = 1.7639                  ' TWIPS scaling factor 2540# / (72# * 20#)
Set barcode1 = CreateObject("dBarcode.Control")  'start dBarcode
barcode1.Units = "mm"
barcode1.CodeType = 9              'set code 128
barcode1.ShowText = 1              'set text on
barcode1.FontSize = 10             'font 10 point
barcode1.FontName = "Arial"        'font Arial
barcode1.ImageHeight = 15          'height 15 mm
barcode1.ImageWidth = 30           'length 30 mm
'a.Xunit = 12                      'X = 12 Mils
barcode1.MarginSize = 4            'light margin 4 mm
barcode1.Data= Field5.Value          ' this is the barcode data source
Set Picture1.FormattedPicture = barcode1.Picture
Picture1.Height = barcode1.PictureHeight / twips
Picture1.Width = barcode1.PictureWidth / twips
End Sub
```



```
Private Sub Section5_Format(ByVal pFormattingInfo As Object)
Dim twips As Single
twips = 1.7639 ' 2540# / (72# * 20#)
Set barcode1 = CreateObject("dBarcode.Control")  'start dBarco
barcode1.Units = "mm"
barcode1.CodeType = 9                    'set code 128
barcode1.ShowText = 1                   'set text on
barcode1.FontSize = 10                  'font 10 point
barcode1.FontName = "Arial"             'font Arial
barcode1.ImageHeight = 15               'height 15 mm
barcode1.ImageWidth = 30                'length 30 mm
'a.Xunit = 12                           'X = 12 Mils
barcode1.MarginSize = 4                 'light margin 4 mm
barcode1.Data = Field4.Value ' this is the barcode data source
Set Picture3.FormattedPicture = barcode1.Picture
Picture3.Height = barcode1.PictureHeight / twips
Picture3.Width = barcode1.PictureWidth / twips
```

Step 5

Finally compile and run the program – which will now display the report complete with barcodes.



Employee Profiles

29/01/2008 13:29:20

Brid, Justin

Name:	Justin Brid
Position:	Marketing Director
Extension:	377
Home Phone:	(808)383-1662
Birthdate:	08/10/1977
Home Address:	2 impasse du Soleil Haguenau, Bas-Rhin France 67500

Note: If the application is transferred to another computer, then that machine must contain the dBarcode control to be able to reproduce the barcodes. See Server/Multi-user version and distribution.

1D OLE Server properties

The image created depends on a number of Properties summarized below and described in more detail in the section following.

Summary of Properties

The properties that can be set to generate an image

AutoCheck specifies whether any check digits are calculated automatically

BackColor the colour behind the bars*

BarRatio specifies the ratio of wide/narrow bar (times ten) for some barcode types

BarReduce specifies the percentage reduction in bar thickness (useful for allowing for ink spread in wet-ink printing processes.

BearerSize specifies the thickness of bearer bars for those barcodes that may have bearer bars

BorderSize specifies the thickness of any border around the image (in points)

CharSpacing the percentage of the barcode width over which the text under the barcode should be spread out.

CodeType specifies the barcode type required

Data specifies the characters that make up the code

ExtendBearers Allows bearer bars to extend into light margins.

ForeColor the colour of the bars and any text under the bars

FontSize, FontName specifies the font for human readable text

FontBold, FontItalic font characteristics for Windows

ImageHeight required target height of barcode image (in units determined by the setting of Units)

ImageWidth required target width of barcode image (in units determined by the setting of Units)

Indicators specifies whether light margin indicators should be displayed alongside human readable text for those barcode types that support these

Justify specifies the text justification for human readable text under the barcode

MarginSize specifies the size of the light margins (in units determined by the setting of Units)

NominalSize specifies a percentage of the barcode's standard size at which the image should be created. If specified the calculated value overrides the ImageHeight and ImageWidth parameters. (Windows only)

Orientation specifies the orientation of the barcode image

PictureHeight the height of barcode picture generated (in Pixels) (Windows only)

PictureWidth the width of barcode picture generated (in Pixels) (Windows only)

BothBearers determines whether both upper and lower bearer bars are displayed

ShowText specifies that the text content of the barcode should be displayed under the bars

ShowCheck specifies whether any automatic check digit is displayed (for those barcode type which permit this)

Units specifies the units of ImageHeight, ImageWidth, BorderSize and MarginSize parameters; default is mm (millimeter), also allowed are cm (centimeter) and in (inch)

Xunit specifies the thickness of each barcode element in Mils (1/1000 inch)

Many of these properties have default values (see the reference section), so do not require changing if you can use the default values. The properties that must be set to obtain a barcode are

CodeType specifies the barcode type required

Data specifies the characters that make up the code

AutoCheck

Type: BOOL

Default: FALSE

Allowed values: FALSE (checkdigit characters not calculated)

TRUE (checkdigit characters calculated and appended to code for appropriate code types)

BackColor

Type: LONG

Default: &H00FFFFFF& on Windows.

Allowed values: 0 (black)

to &H00FFFFFF& (white)

Sets the colour of the image background. This value may be over-ridden by the Transparent property for WMF images. (Ignored for PNG images in Windows).

BarRatio

Type: FLOAT

Default 2.5

Allowed values: 2.0 – 3.0

This setting allows some barcode types to have the Wide bar/Narrow bar ratio modified.

Applies mainly to Code 39 and Interleaved 2 of 5 barcodes.

BarReduce

Type: SHORT

Default: 0 the thickness of each line drawn on the barcode image is reduced by this percentage amount. This property may be used to compensate for ink spreading during wet-ink printing.

Allowed values: 0 - 50 (%)

BearerSize

Type: FLOAT

Default: 0

Allowed values: 0 - 10000

Allows the thickness of bearer bars for those barcode which support bearers to be set in units of the current value of the Units property.

Note that for most barcode types the number of bearer bars displayed depends on the setting of the ShowBearerBars property. If ShowBearerBars is True the bars are displayed above and below the barcode. If ShowBearerBars is False then only a single bearer bar is displayed above the barcode.

BorderSize

Type: FLOAT

Default: 0

Allowed values: 0 (no border)

to 255

BothBearers

Type: BOOL

Default: FALSE

Allowed values: TRUE or FALSE

When TRUE causes bearer bars to be draw above and below the barcode. When FALSE no bearer bars are drawn if the BearerSize property is zero. Otherwise a FALSE setting causing only a single bearer bar to be displayed above the barcode.

CharSpacing

Type: SHORT

Default: 0

Allowed values 0 and 10 - 100

The percentage of the barcode width that the text under the barcode will occupy. A value of 0 causes the characters to be displayed with their "normal" spacing. It is the user's responsibility to ensure that the value provided is sensible in comparison with the fontsize.

CodeType

Type: SHORT

Default: 8

Allowed values: The range of values shown in the barcode types table below.

Barcode Types Tables for OLE server

The barcode types supported by the OLE Server in this release of Barcodes for Crystal Reports are listed in the table below, arranged alphabetically using their most common barcode names

CodeType	Barcode	no. of characters	check digit
11	2 of 5	any numbers	
13	3 of 9	any	
35	4-State	any	1 optional
41	Australia Post*	numeric	1 automatic
60	China Postal	Any numeric	1 optional
16	Codabar/NW-7	any	
15	Code 11	any	1 or 2
9	Code 128	any	1 automatic
8	Code 39	any	1 optional
36	Code 93	any	2 optional
14	Code B	any numbers	none
62	Code128 Type A	any	1 automatic
63	Code128 Type B	any	1 automatic
68	Databar Omni	13 numeric	0
69	Databar Truncated	13 numeric	0
70	Databar Limited	13 numeric	0

71	Databar Expanded	any	0
43	Deutsche Post	numeric	
10	EAN/UCC-128	any	1 automatic
45	EAN/UCC-14	14 numbers	1
0	EAN13	13 numbers	1
2	EAN13+2	15 numbers	1
3	EAN13+5	18 numbers	1
1	EAN-8	8 numbers	
21	EAN8+2	10 numbers	1
22	EAN8+5	13 numbers	1
37	Ext. Code 93	any (full ASCII)	2 optional
18	Ext. Code 39	any (full ASCII)	1 optional
31	FIM A	fixed code	0
32	FIM B	fixed code	0
33	FIM C	fixed code	0
75	FIM D	fixed code	0
72	HIBC Code 39	various	1
73	HIBC Code 128	various	1
59	IATA 2 of 5	Any numeric	
66	InfoMail A*	21 numeric	
61	Intelligent Mail*	numeric	
74	IM Package Barcode	numeric	1
12	Interleaved-2 of 5	number pairs	1 optional
38	ISBN	10/13 digit ISBN	1 automatic
54	ISBN+2 digit	any / 2	1 automatic
55	ISBN+5 digit	any / 5	1 automatic
49	ISMN	8/9 digit ISMN	1 automatic
48	ISSN	9/10 digit ISSN	1 automatic
56	ISSN+2 digit	any / 2	1 automatic
57	ISSN+5 digit	any / 5	1 automatic
53	Italian Postal 2/5	12 numbers	1 automatic
52	Italian Postal 3/9	2+8+c+2	1
6	ITF-14	14 numbers	1 (EAN optional)
7	ITF-6	6 numbers	
58	Japan Post*	any	automatic
51	Korean Postal Authority*	6	1
39	Matrix 2/5	numeric	1 optional
17	MSI	any numbers	1 or 2
46	Planet 12*	12	1 automatic
47	Planet 14*	14	1 automatic
40	Plessey	numeric/some alpha	2
28	PostNet type A*	5 numbers	1
29	PostNet type C*	9 numbers	1
30	PostNet type C'*	11 numbers	1
67	PZN	6 numbers	1 automatic
34	RM4SCC*	any	1
44	SISAC	SICI codes	1
50	SSCC	17 numbers	1
42	Swiss Post	fixed code	0
26	Telepen ASCII	any	1 optional
27	Telepen numeric	any	1 optional

25	Telepen standard	any	1 optional
4	UPC-A	12 numbers	1
19	UPCA+2	14 numbers	1
20	UPCA+5	17 numbers	1
5	UPC-E	7 numbers	1
23	UPCE+2	9 numbers	1
24	UPCE+5	12 numbers	1
64	UPC-E0	12 reduced to 7	1
65	UPC-E1	12 reduced to 7	1

Data

Type: BSTR

Default: "12345"

Allowed values: Any text string.

Note: only text strings recognised as valid barcodes will result in a barcode picture. Illegal text string will cause an ErrorCode value to be set and not generate an image.

The Data should be the last property set - as setting this property causes the barcode image to be recreated.

Errorcode

The read-only Property M_barcode.ErrorCode AFTER the Pic(type) method has been called or a Picture image retrieved may contain an error code if the server was unable to generate a barcode image. The property value will be one of the following:

0	No error
1	Wrong code length
2	Unrecognised code type
3	Wrong add-on code length
4	Illegal character in code
5	Error in embedded code
6	Generated line width less than 1 unit
7	Invalid text font
8	Invalid device context
9	Invalid Data property

ExtendBearers

Type: BOOL

Default: FALSE

When TRUE allows bearer bars to extend into light margins. When FALSE bearer bars cover the bars only.

Extra1

Type: BOOL

Default FALSE

These additional properties are not normally used. However, they do provide additional functions for a limited number of specific barcode types. See Barcodes HELP for details.

Extra2

Type: BOOL

Default FALSE

These additional properties are not normally used. However, they do provide additional functions for a limited number of specific barcode types. See Barcodes HELP for details.

FontName

Type: BSTR

Default: Arial

Allowed values: Any accessible TrueType font installed ON THE SERVER.

FontSize

Type: short

Default: 10

Allowed values: Any allowed vlaue (typically 6 – 72)

FontBold

Type: BOOL

Default: FALSE or 0

Allowed values: FALSE (font is not bold) or TRUE (font is bold)

FontItalic

Type: BOOL

Default: FALSE or 0

Allowed values: FALSE (font is not italic) or TRUE (font is italic)

ForeColor

Type: LONG

Default: 0 on Windows

Allowed values: 0 (black)

to &H00FFFFFF& (white)

Set the colour of the image foreground, i.e. the bars and text colour. This property should be set in code and not as a parameter in a graphic location URL.

(Ignored for PNG images in Windows).

GapToText

Type: SHORT

Default: 0 A value given as a percentage of the text font point size by which text is raised within its bounding rectangle.

Allowed values: 0-100 (%)

Note: For most applications this value should be set to 0.

ImageHeight

Type: FLOAT

Default: 20.0 mm

Allowed values: 5.0 – 1000.0 mm

The target height of the barcode image. Note that this is the target size of the image. The font height will be as requested if the image is reproduced at this size. If the image is resized to say double this value then the font will also be increase in absolute size. Also this value will be placed in the metafile header of WMF images.

ImageWidth

Type: FLOAT

Default: 25 mm

Allowed values: 1.0 – 1000.0 mm

The target width of the barcode image. Note that this is the target size of the image. The font width will be as requested if the image is reproduced at this size. If the image is resized to say double this value then the font shape will change. Also this value will be placed in the metafile header of WMF images.

If the value of Xunit is NOT 0 then the value of ImageWidth will be recalculated based on the number of elements in the barcode and the value of Xunit.

Indicators

Type: BOOL

Default: FALSE A value of TRUE causes the light margin indicators to be displayed. For some EAN barcodes there are recommended ways for the margin indicators to be shown on the image, and the Prefix code and checkdigit for UPC-A and UPC-E codetype to be displayed in the light margins. A value of FALSE prevents the display of the light margin indicators.

Justify

Type: SHORT

Default: 0

Allowed values 0,1 and 2

Justify sets the justification of any text displayed under the barcode. A value of 0 centres the text, 1 gives left justification and 2 gives right justification. Justification is to the outside of the light margins, so no justification can be used when light margin indicators are displayed.

MarginSize

Type: FLOAT

Default: 0

Allowed values: 0 – 100 mm

The MarginSize property sets the Light Margin space on either side of a barcode image. The units are the units specified by the current value of the Units property.

NominalSize

Type: SHORT

Default: 0 When this property is set to a value between 50 and 200 the width of the image created will be scaled to generate that percentage of the barcodes Standard. When this property is 0 no scaling is applied.

Allowed values: 0 or 50-200

Note that this feature is relevant only for some EAN, UPC and ITF code types. It is ignored for barcode types that do not have a standard size.

Orientation

Type: SHORT

Default: 0 The value of this parameter determines the orientation of the barcode image created.

Allowed values: 0 = normal orientation

1 = image rotated left by 90 degrees (clockwise)

2 = image rotated right by 90 degrees (anti-clockwise)

3 = image inverted.

Note that the rotation of text is only supported by TrueType and other rotatable fonts. Note also that some applications do not correctly handle metafiles that contain rotated text.

ShowCheck

Type: BOOL

Default: 0 When set to 1 this property causes the any automatically calculated check digit to be included in the any text displayed along with the barcode. When set to 0 the check digit is not displayed.

Note that this property has no effect on those codetypes for which the checkdigit display is mandatory (including EAN and UPC codes).

ShowText

Type: BOOL

Default: TRUE

Allowed values FALSE (text version of the code is NOT included in the barcode image)

TRUE (text version of the code IS included in the barcode image)

Transparent

Type: BOOL

Default: FALSE

Allowed values TRUE or FALSE. When set to TRUE this property causes a barcode WMF image to be created on a transparent background (ie. and background colour is ignored). The property should be set to 1 only by users who are confident that the final barcode image will be produced on a background that does not interfere with barcode scanning.

Units

Type BSTR

Default: “mm”

The units used for ImageWidth, ImageHeight, MarginSize, BorderSize and BearerSize property values.

Allowed values:

“mm” - millimetres
“cm” – centimeters
“in” - Inches

Xunit

Type: SHORT

Default: 0

Allowed range: 10 - 255

The xunit property may be used to specify the width (in Mils) of the smallest element in the barcode.

Note that setting this property to a value other than 0 causes the control to resize itself to a width calculated from the number of X units in the barcodes. Using values smaller than 8 will produce a barcode image, but that image will not meet standard specifications and may not scan.

Legacy 2D OLE server

The OLE server is used by the Report Designer Component in many versions of Crystal Reports. The server is dBarcodeCR2D.dll which is installed in the system folder of the Barcode Tools for Crystal Reports installation directory. The OLE server is self-registered on installation. If the dll is moved it must be re-registered with the machine’s OLE system by running

Regsvr32 full_path\dBarcodeCR2D.dll

From a command prompt, where full_path is the full path to the directory containing the dll.

The image created by the image creator may be obtained from its Picture property.

Picture property

The Picture property returns an OLE picture which is generally used as the source of a picture in a Visual Basic PictureBox.

In this case the physical size of the picture must be obtained from the image creator by retrieving the PictureWidth and PictureHeight properties once the image has been created. These values are reported in Windows HIMETRIC units, and so require scaling to the units used by the form containing the PictureBox – which in Crystal Reports is TWIPS (twentieths of a point per inch!).

The Picture property may be placed in a VB PictureBox using

```
Set PictureBox1.FormattedPicture = M_barcode.Picture
PictureBox1.Height = M_barcode.PictureHeight / twips
PictureBox1.Width = M_barcode.PictureWidth / twips
```

Where twips is the scaling factor between HIMETRIC and TWIPS units.

The Picture property is used in this way to generate barcode images in applications created using older versions of the Report Designer Component for VB6. [For the RDC of Crystal Reports 11 the use of the Pic() method is to be preferred]

Crystal Reports 2D Designer for VB6

The Crystal Reports Designer Component (RDC) was included with Crystal Reports 8, 8.5 and 9, and allows reports to be incorporated into programs that can (among other things) display the report. The RDC allows images to be created dynamically, and 2D Barcode Tools for Crystal Reports includes an OLE server that can use this facility to create barcode images when a report record is displayed.

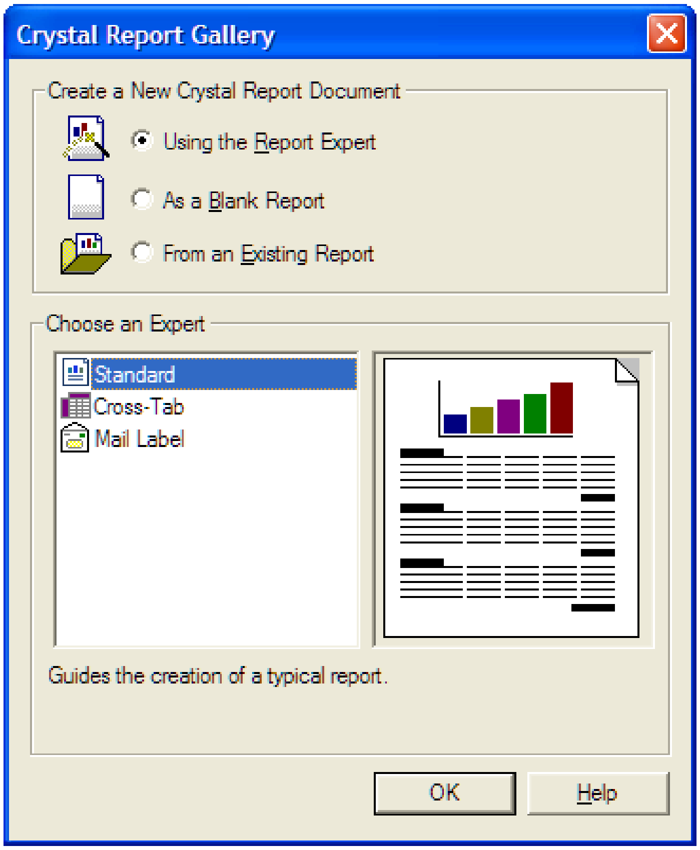
To create an application containing dynamically generated barcodes follow the steps below:

Step 1.

Open Visual Basic and create a new Windows Application (such as Standard Exe).

In the Project Explorer Window right click on the project title and choose Add

Select Crystal Report 9 [or 8.5] from the list of options. Select an existing report or create a blank report and connect to some data.



Create a New Crystal Report Document

● Using the Report Expert

○ As a Blank Report

○ From an Existing Report

Choose an Expert

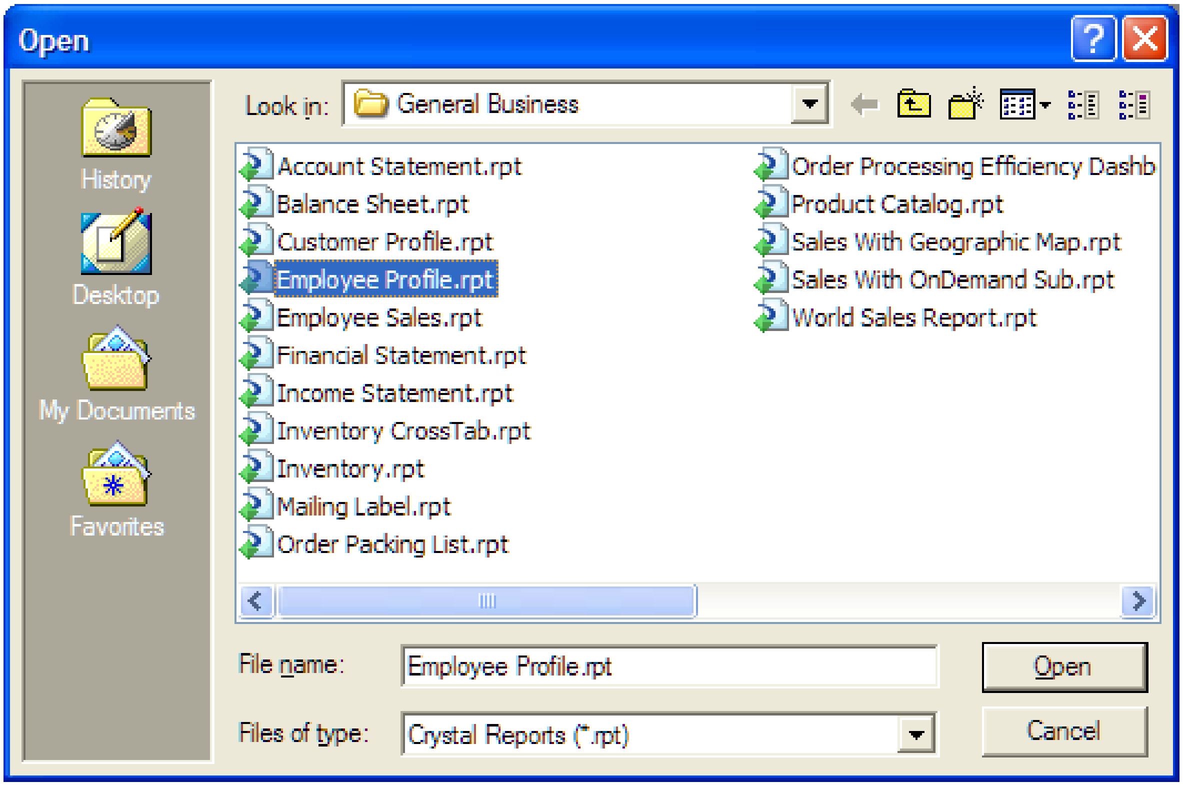
Standard

Cross-Tab

Mail Label

Guides the creation of a typical report.

If you wish to add an existing report using the Report Expert you can navigate to the report, eg

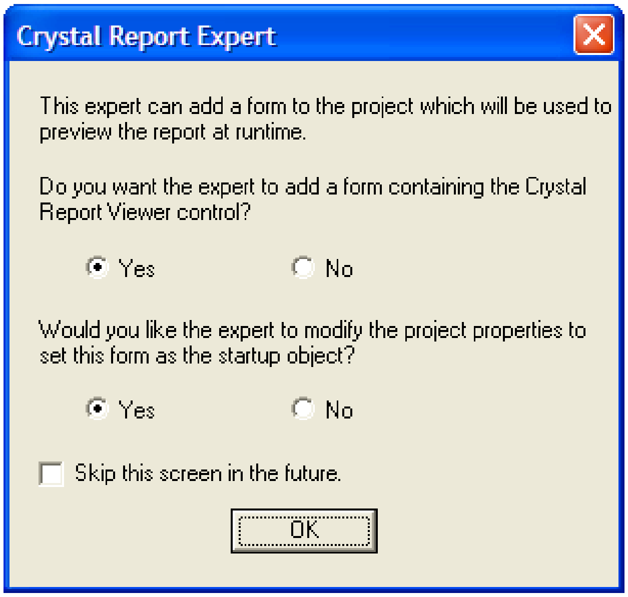


Look in: General Business

File name: Employee Profile.rpt

Files of type: Crystal Reports (*.rpt)

And elect to have the expert create a form containing the layout of the report



This expert can add a form to the project which will be used to preview the report at runtime.

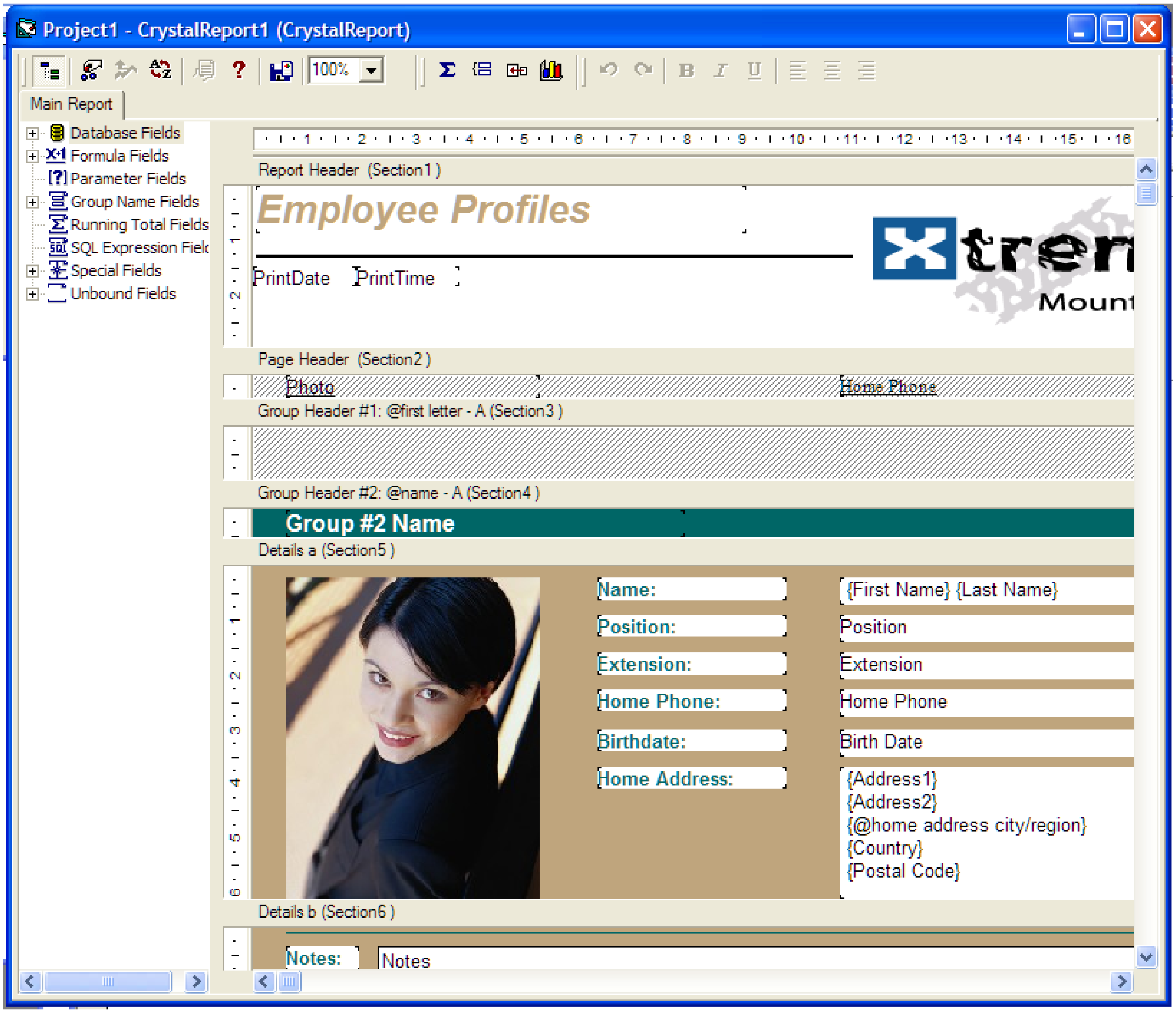
Do you want the expert to add a form containing the Crystal Report Viewer control?

● Yes ○ No

Would you like the expert to modify the project properties to set this form as the startup object?

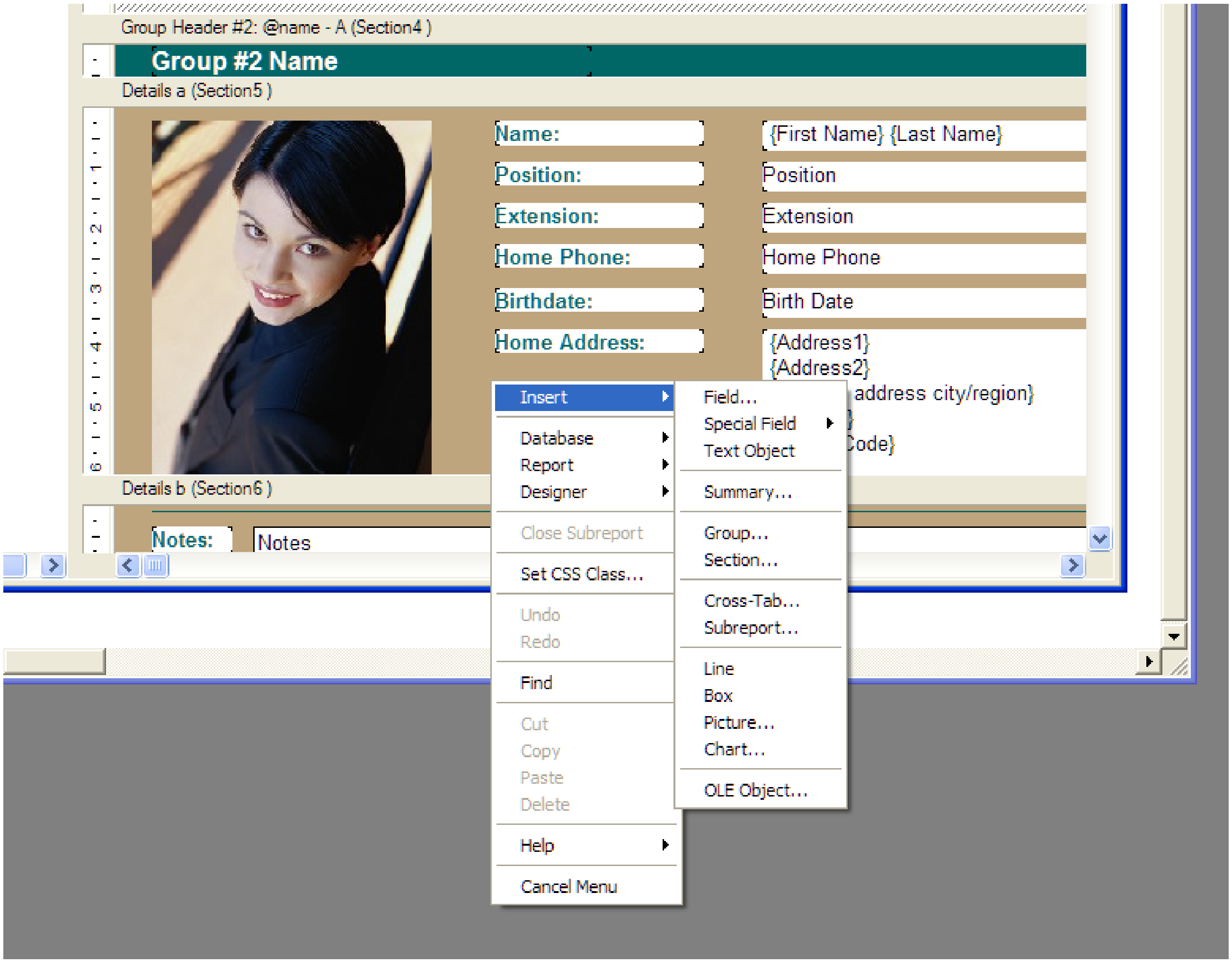
● Yes ○ No

□ Skip this screen in the future.

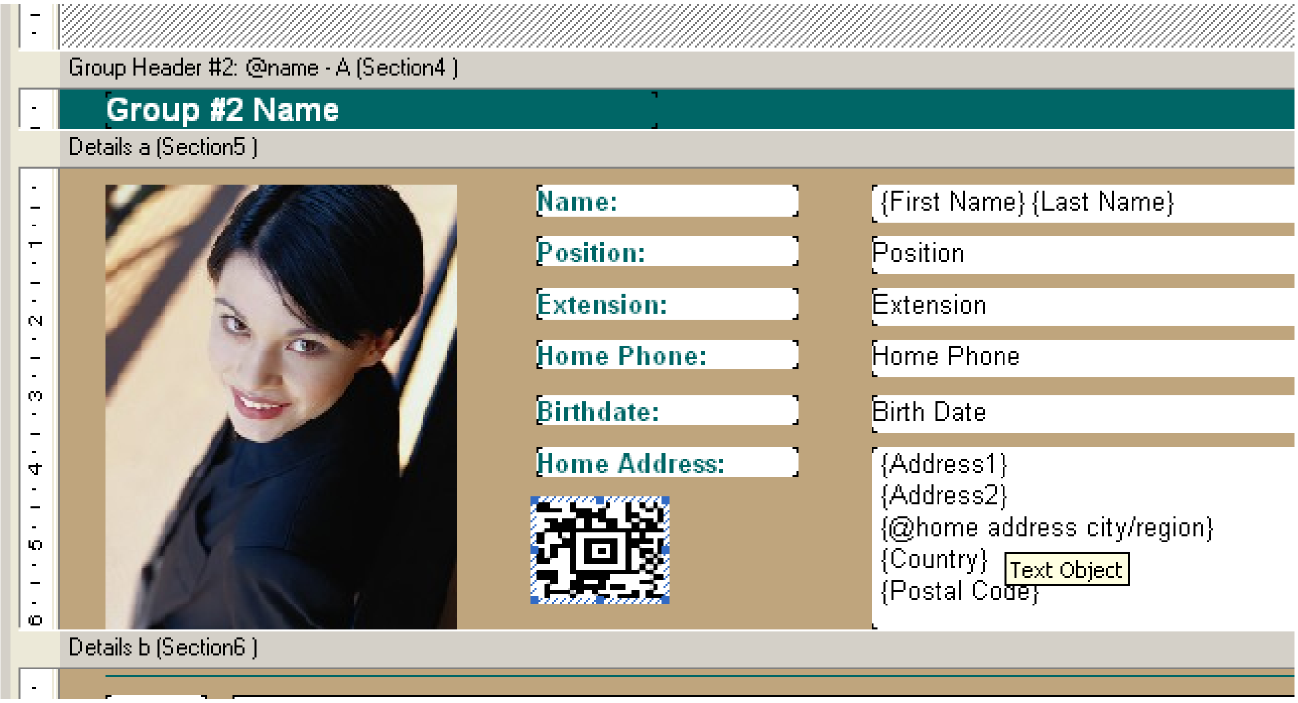


Step 2

Right-click on the area of the report’s section at which you wish to place a barcode image and select Insert and Picture from the pop-up menus displayed

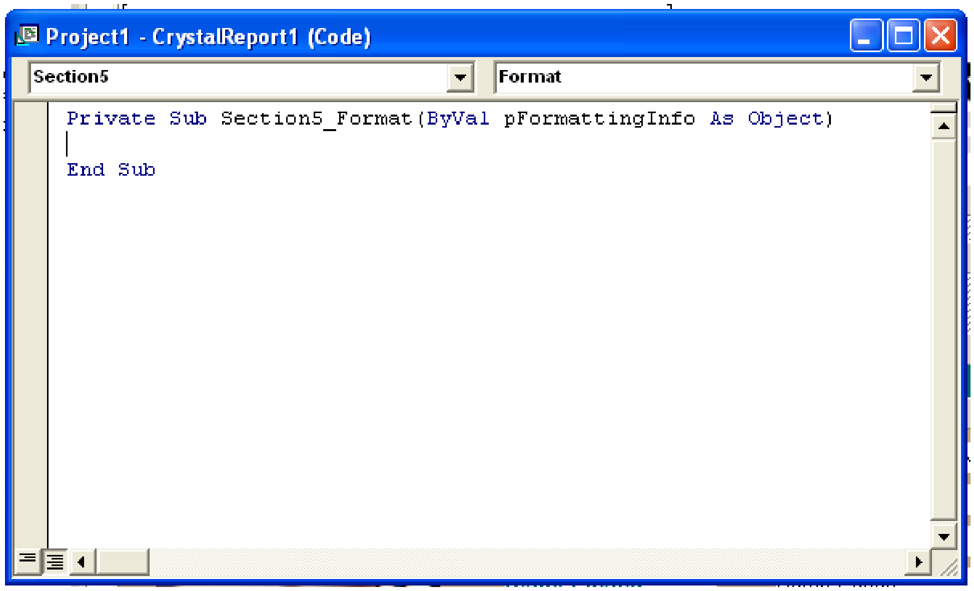


When the dialog appears for you to specify the source of the picture navigate to the 2D Barcode Tools for Crystal Reports installation directory and choose the dummy barcode image. A barcode placeholder will then be displayed on the report.



Step 3

Double-click on the barcode image and a code window will open.



```
Private Sub Section5_Format(ByVal pFormattingInfo As Object)

End Sub
```

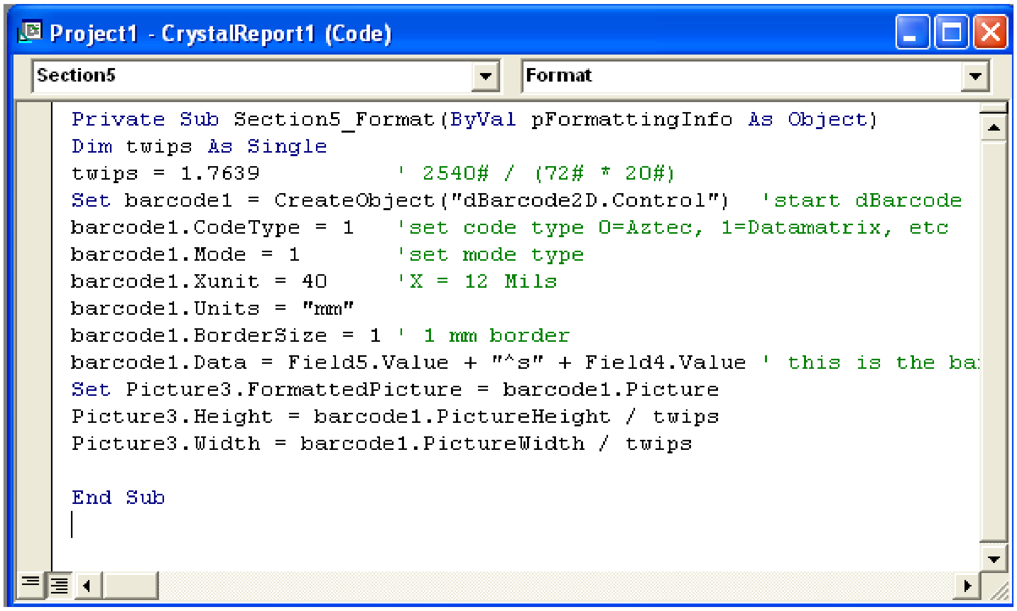
This will contain the code which is run each time the section you double-clicked on is “formatted”, i.e. prepared for display.

Step 4

Enter the code required to create the barcode from data in your chosen source field.

The code required here will frequently be very similar to that shown below, in fact if you have a field named Field5 on the report you can copy and paste this code directly into code window.

```
Private Sub scnDetail_Format(ByVal pFormattingInfo As Object)
Dim twips As Single
twips = 1.7639                    ' 2540# / (72# * 20#)
Set barcode1 = CreateObject("dBarcode2D.Control")  'start dBarcode
barcode1.CodeType = 1             'set code type 0=Aztec, 1=Datamatrix, etc
barcode1.Xunit = 20               'X = 12 Mils
barcode1.Units=”mm”
barcode1.BorderSize=1             ‘1 mm border
barcode1.Data = Field5.Value      ' this is the barcode data source
Set Picture3.FormattedPicture = barcode1.Picture
Picture3.Height = barcode1.PictureHeight / twips
Picture3.Width = barcode1.PictureWidth / twipsEnd Sub
```

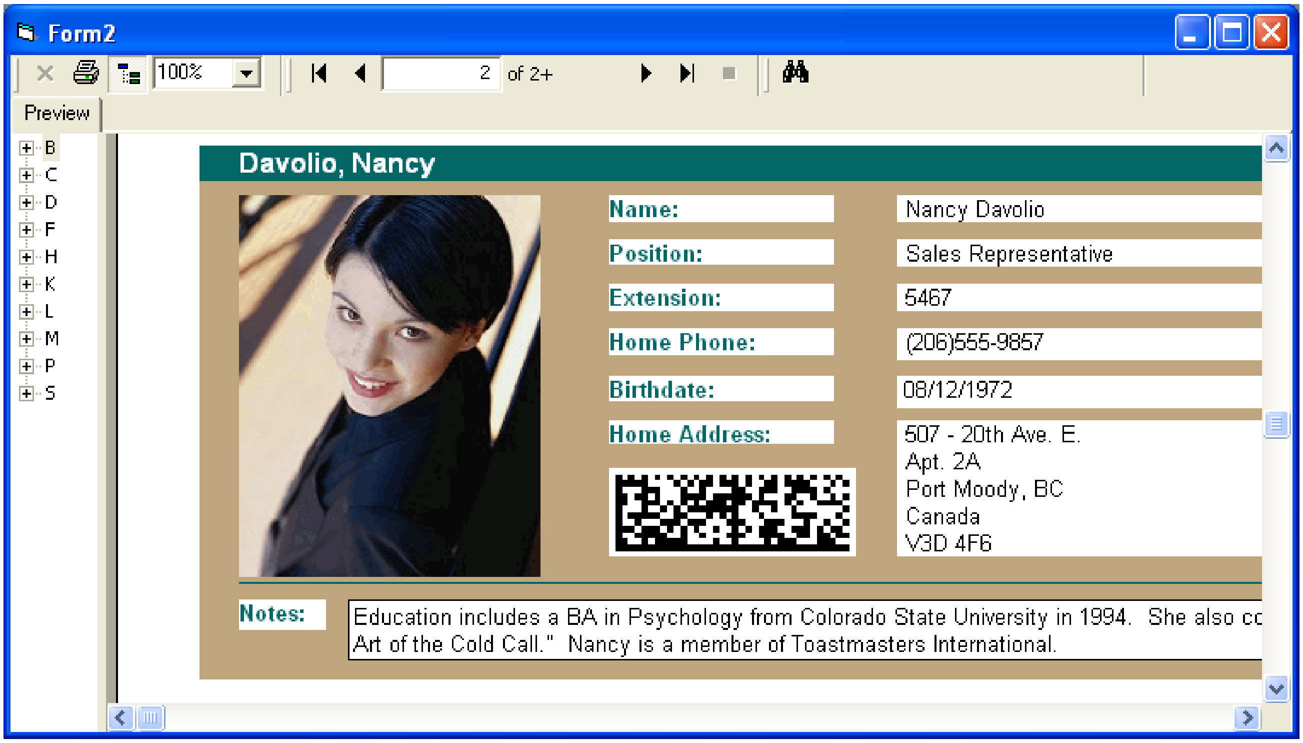


```
Private Sub Section5_Format(ByVal pFormattingInfo As Object)
Dim twips As Single
twips = 1.7639          ' 2540# / (72# * 20#)
Set barcode1 = CreateObject("dBarcode2D.Control")  'start dBarcode
barcode1.CodeType = 1    'set code type 0=Aztec, 1=Datamatrix, etc
barcode1.Mode = 1        'set mode type
barcode1.Xunit = 40      'X = 12 Mils
barcode1.Units = "mm"
barcode1.BorderSize = 1 ' 1 mm border
barcode1.Data = Field5.Value + "^s" + Field4.Value ' this is the ba
Set Picture3.FormattedPicture = barcode1.Picture
Picture3.Height = barcode1.PictureHeight / twips
Picture3.Width = barcode1.PictureWidth / twips

End Sub
```

Step 5

Finally compile and run the program – which will now display the report complete with barcodes.



Davolio, Nancy

Name: Nancy Davolio

Position: Sales Representative

Extension: 5467

Home Phone: (206)555-9857

Birthdate: 08/12/1972

Home Address: 507 - 20th Ave. E.
Apt. 2A
Port Moody, BC
Canada
V3D 4F6

Notes: Education includes a BA in Psychology from Colorado State University in 1994. She also cc
Art of the Cold Call." Nancy is a member of Toastmasters International.

Note: If the application is transferred to another computer, then that machine must contain the dBarcode control to be able to reproduce the barcodes. See Server/Multi-user version and distribution.

2D OLE Server properties

Whichever technique is used to obtain the barcode image, the image created depends on a number of Properties summarized below and described in more detail in the section following.

Summary of Properties

The properties that can be set to generate an image

BackColor the colour behind the bars*

BarReduce specifies the percentage reduction in bar thickness (useful for allowing for ink spread in wet-ink printing processes.

BorderSize specifies the thickness of any border around the image (in points)

CodeType specifies the barcode type required

Columns specifies the number of element or codeword columns in the symbol (depending on the barcode type)

Data specifies the characters that make up the code

ForeColor the colour of the bars and any text under the bars

ImageHeight required target height of barcode image (in units determined by the setting of Units)

ImageWidth required target width of barcode image (in units determined by the setting of Units)

Orientation specifies the orientation of the barcode image

PictureHeight the height of barcode picture generated (in Pixels) (Windows only)

PictureWidth the width of barcode picture generated (in Pixels) (Windows only)

Units specifies the units of ImageHeight, ImageWidth and BorderSize parameters; default is mm (millimeter), also allowed are cm (centimeter) and in (inch)

Xunit specifies the thickness of each barcode element in Mils (1/1000 inch)

Ysize specifies the height of bar elements as multiple of the Xunit value.

Many of these properties have default values (see the reference section), so do not require changing if you can use the default values. The properties that must be set to obtain a barcode are

CodeType specifies the barcode type required

Data specifies the characters that make up the code

BackColor

Type: LONG

Default: &H00FFFFFF& on Windows

Allowed values: 0 (black)

to &H00FFFFFF& (white)

Sets the colour of the image background. This value may be over-ridden by the Transparent property for WMF images.

This property should be set in code and not as a parameter in a graphic location URL.

(Ignored for PNG images in Windows).

BarReduce

Type: SHORT

Default: 0 the thickness of each line drawn on the barcode image is reduced by this percentage amount. This property may be used to compensate for ink spreading during wet-ink printing.

Allowed values: 0 - 50 (%)

BorderSize

Type: FLOAT

Default: 0.0

Allowed values: 0 (no border)

to half size of image.

Units are those specified by the Units property.

CodeType

Type: SHORT

Default: 0

Allowed values: The range of values defined in the barcode types table shown below.

Barcode Types Tables for the OLE server

The barcode types supported by the Windows image creator in this release of 2D Barcode Tools for Crystal Reports are listed in the table below, arranged alphabetically using their most common barcode names

Barcode Type	**CodeType value**
Aztec	0
Datamatrix	1
PDF417	2
Micro PDF417	3
Truncated PDF417	4
QR Code	5
Micro QR Code	6
GS1-DataBar*	7
Maxicode*	8

*Notes:

GS1-Databar was formerly known as RSS symbology.

Columns

Type: SHORT

Default: 0

When non-zero this parameter specifies the number of element columns or codeword columns in the symbol, and so determines the overall shape of the symbol.

For Aztec and Datamatrix symbols the value specifies the number of elements across the symbol. This value can range from 15 – 150 for Aztec or 10 – 144 for Datamatrix symbols.

For PDF417 the value specifies the number of column groups (excluding the start and stop groups) across the symbol. Each group of columns in a PDF417 symbol is a codeword and a row may contain from 1 – 30 codewords.

For Databar Expanded-stacked symbols the value specifies the number of codeword pairs in the first row of the symbol – subsequent rows may be shorter than the first row. The value can range from 1 – 10.

For all other barcode types the value of Columns is not used.

Data

Type: BSTR

Default: "12345"

Allowed values: Any text string.

Note: only text strings recognised as valid barcodes will result in a barcode picture. Illegal text string will cause an ErrorCode value to be set and not generate an image.

The Data should be the last property set - as setting this property causes the barcode image to be recreated.

Errorcode

The read-only Property M_barcode.ErrorCode AFTER the Pic(type) method has been called or a Picture image retrieved may contain an error code if the server was unable to generate a barcode image. The property value will be one of the following:

0	No error
1	Wrong code length
2	Unrecognised code type
4	Illegal character in code
5	Error in embedded code
6	Generated line width less than 1 unit
8	Invalid device context
9	Invalid Data property

ForeColor

Type: LONG

Default: 0 on Window

Allowed values: 0 (black)

to &H00FFFFFF& (white)

Set the colour of the image foreground, i.e. the bars and text colour. This property should be set in code and not as a parameter in a graphic location URL.

(Ignored for PNG images in Windows).

ImageHeight

Type: FLOAT

Default: 15 mm

Allowed values: 5 – 1000 mm

to page height in units determined by the setting of Units

The target height of the barcode image – which is ignored if a non-zero value of Xunit is provided..

Note that this is the target size of the image. The actual size may vary to accommodate the requirement that each barcode element is an integer number of pixels at the specified resolution. Also this value will be placed in the metafile header of WMF images.

ImageWidth

Type: FLOAT

Default: 30 mm

Allowed values: 0.1 – 1000 mm

to form width in units determined by the setting of Units

The target width of the barcode image – which is ignored if a non-zero value of Xunit is provided..

Note that this is the target size of the image. The actual size may vary to accommodate the requirement that each barcode element is an integer number of pixels at the specified resolution. Also this value will be placed in the metafile header of WMF images.

If the value of Xunit is NOT 0 then the value of ImageWidth will be recalculated based on the number of elements in the barcode and the value of Xunit.

Mode

Type: Integer

Default: 0

Allowed values;

Aztec Mode values

Type	Mode
Normal	0
Compact	1
Full Range	2
Rune	3

Datamatrix Mode values

Type	Mode
Automatic – Square	0
Automatic – Rectangular	1
ASCII encoding – Square	2
ASCII encoding – Rectangular	3
C40 encoding – Square	4
C40 encoding – Rectangular	5
Text encoding – Square	6
Text encoding – Rectangular	7
X12 encoding – Square	8
X12 encoding – Rectangular	9
EDIFACT encoding – Square	10
EDIFACT encoding – Rectangular	11
Base 256 encoding – Square	12
Base 256 encoding – Rectangular	13

Type	Mode

Square	0
Rectangular	1

GS1-Databar

Type	Mode
Omnidirectional	0
Truncated	1
Limited	2
Stacked	3
Stacked-Omnidirectional	4
Expanded	5
Expanded-Stacked	6

PDF417 Mode values

Type	**Mode**
EXC Alpha	0
EXC Lower case	1
EXC Mixed	2
EXC Punctuation	3
Binary/ASCII Plus	4
Numeric	5

Note: Mode 0 also provides fully automatic encoding.

Micro PDF417 Mode values

Type	**Mode**
Text	0
Byte	1
Numeric	2

QR Code Mode values

Type	**Mode**
Numeric	0
Alphanumeric	1
Byte	2

Orientation

Type: SHORT

Default: 0 The value of this parameter determines the orientation of the barcode image created.

Allowed values: 0 = normal orientation

1 = image rotated left by 90 degrees (clockwise)

2 = image rotated right by 90 degrees (anti-clockwise)

3 = image inverted.

Note that the rotation of text is only supported by TrueType and other rotatable fonts. Note also that some applications do not correctly handle metafiles that contain rotated text.

Transparent

Type: BOOL

Default: FALSE

Allowed values TRUE or FALSE. When set to TRUE this property causes a barcode WMF image to be created on a transparent background (ie. and background colour is ignored). The property should be set to 1 only by users who are confident that the final barcode image will be produced on a background that does not interfere with barcode scanning.

Units

Type BSTR

Default: “mm”

The units used for ImageWidth, ImageHeight, MarginSize and BearerSize values.

Allowed values:

“mm” - millimetres
“cm” – centimeters
“in” - Inches

Xunit

Type: SHORT

Default: 0

Allowed range: 10 - 255

The xunit property may be used to specify the width (in Mils) of the smallest element in the barcode.

Note that setting this property to a value other than 0 causes the control to resize itself to a width calculated from the number of X units in the barcodes. Using values smaller than 8 will produce a barcode image, but that image will not meet standard specifications and may not scan.

Ysize

Type: SHORT

Default: 0

Allowed range: 0 - 255

Specifies the height of bars as multiples of the bar width. Applies only to the PDF417 family – for which the allowed range is 2 – 5 inclusive – and the GS1-Databar family. Ignored for other barcode types.

Index