DataBar User Function Library for Crystal Reports

DataBar UFL Installation and Distribution

The DataBar UFL requires that the Microsoft .NET Framework 2.0 is installed. Since the DataBar UFL is designed to work with both the Crystal Reports Designer and Crystal Reports Redistributable versions, a manual instruction process is provided (the user performing the install must have Administrator rights):

1. Install the DataBar Fonts from their downloaded location on the local machine. After the DataBar font installation, it is recommended to reboot the system as the Font Cache does not always immediately reflect the installed fonts.
2. Download the DataBar UFL package and unzip the contents to a convenient location.
3. In the unzipped location, there are folders labeled 32-bit and 64-bit to distinguish the DLL runtime of each file. Install/distribute as indicated:

DataBar UFL Installation and Distribution: 32-bit Operating System

Type	File to Distribute	Install From	Install Command (adjust if different*)
32-bit DataBar UFL	\32bit\CRUFLIDAutomationDataBar.dll	%windir%\System32*	C:\Windows\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe /codebase C:\Windows\System32\CRUFLIDAutomationDataBar.dll

DataBar UFL Installation and Distribution: 64-bit Operating System

(Both UFL types are required)

Type	File to Distribute	Install From *	Install Command (adjust if different*)
32-bit DataBar UFL	\32bit\CRUFLIDAutomationDataBar.dll	%windir%\SysWOW64	C:\Windows\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe /codebase C:\Windows\SysWOW64\CRUFLIDAutomationDataBar.dll
64-bit DataBar UFL	\64bit\CRUFLIDAutomationDataBar.dll	%windir%\System32	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\RegAsm.exe /codebase C:\Windows\System32\CRUFLIDAutomationDataBar.dll

** %windir% is the Windows directory, which can be found by pressing Windows + R > type in %windir% > Enter*

DataBar UFL Tutorial

1. After installation is complete, the barcode functions will show in the formula editor under *Additional Functions > Visual basic UFLs*. The Functions available are listed below and are supported in 32 and 64 bit.

User Function Library Guide	
UFL Type	Barcode Type (Symbology)
DataBar UFL	**DataBar:** DataBar, DataBar Expanded, DataBar Expanded Stacked, DataBar Limited, DataBar Stacked, DataBar Stacked Omnidirectional, and DataBar Truncated

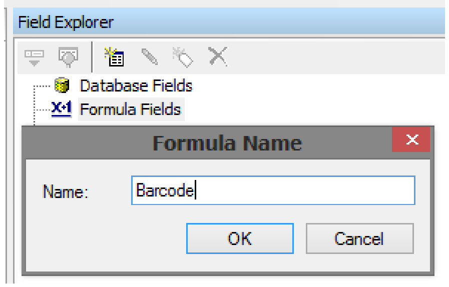
When using a Crystal Reports Redistributable version, IDAutomation does not recommend 'AnyCPU' compilations. Per SAP, Crystal Reports Designer runs only in 32 bit mode.

2. Choose the *Design* tab to enter the design mode of the report.

3. Open the *Field Explorer*. In Crystal 9, select *View - Field Explorer*. In versions prior to 9, select *Insert - Formula Field*.

4. Right-click on *Formula Fields* and select *New*.

5. In the *Formula Name* dialog box, enter a name to identify the formula field. In this example, the formula is simply named "Barcode."



Name: Barcode

6. Click *OK* and select *Use Editor* if asked. The Formula Editor should now be open.

7. In the *Functions* column, expand *Additional Functions > Visual basic UFLs* and expand the IDAutomation section to select the appropriate function.
   - If the IDAutomation functions do not show up in the *Additional Functions > Visual basic UFLs* section after installing the UFL, make sure that the Microsoft .NET Framework 2.0 is installed, reboot the machine, and repeat the installation process carefully.

8. Make sure the utilized function matches the font listed in the *Font to Use* column, using the IDAutomationDataBar functions with an IDAutomation DataBar font.

Functions for the Databar UFL

Barcode Type	Functions	Font to use
DataBar	IDAutomationDataBar(DataToEncode)	IDAutomation DataBar 34
DataBar Expanded	IDAutomationDataBarExpanded(DataToEncode , 22)	IDAutomation DataBar 34
DataBar Expanded Stacked	DataBar Expanded Stacked requires a custom formula field to be copied from the sample report provided.	IDAutomation DataBar 34
DataBar Limited	IDAutomationDataBarLimited(DataToEncode)	IDAutomation DataBar 13
DataBar Stacked	IDAutomationDataBarStacked(DataToEncode)	IDAutomation DataBar 13
DataBar Stacked Omnidirectional	IDAutomationDataBarStackedOmniDirectional(DataToEncode)	IDAutomation DataBar 34
DataBar Truncated	IDAutomationDataBar(DataToEncode)	IDAutomation DataBar 13

9. With the cursor positioned between the parentheses of the function, select the data field that is to be encoded. Data fields used in this formula must be formatted as text. If the data fields are not text, use ToText or cStr for conversion.
   - The syntax of the formula should look similar to the formula below which formats *Field1* of *Table1* to a DataBar barcode font.
     `IDAutomationDataBar({Table1.Field1})` or
     `IDAutomationDataBar(cStr( {Table1.Field1},0 ) )`

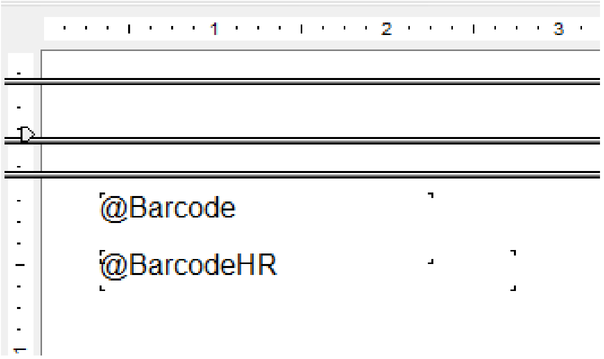
10. Choose Save and then Close. If an "A string is required" or similar error appears, convert the data to a string with the *cStr* function.

- For example:

  ```
  IDAutomationDataBar(Str( {Customers.CustNum},0 ) )
  ```

11. Return to the *Field Explorer* dialog box.

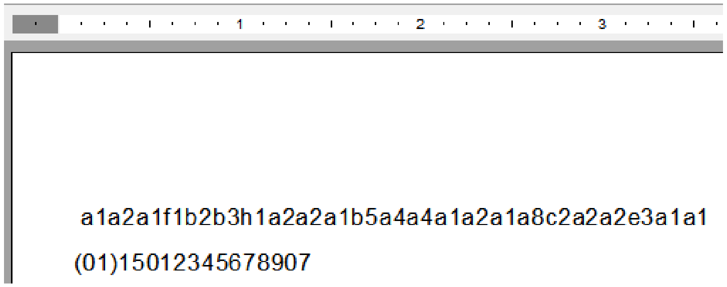
12. Click on the formula field and drag it onto the report.



@Barcode

@BarcodeHR

13. Once the fields are created, choose the *Preview* tab to verify the data is pulled and properly formatted to the barcode font. Note: Unrecognizable characters may appear as the data from the fields - this is the formatted data.



```
a1a2a1f1b2b3h1a2a2a1b5a4a4a1a2a1a8c2a2a2e3a1a1
(01)15012345678907
```

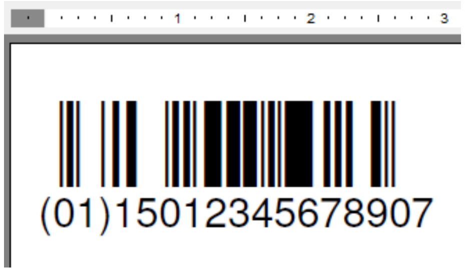
In the graphic above, the *@Barcode* formula produces formatted data for DataBar and the *@BarcodeHR* formula produces the human readable.

14. Select the *Design* tab again and size the barcode formula field to display the appropriate barcode font in the report.

15. Change the formula field to the appropriate barcode font and set the size to 12 points or another appropriate size. Ensure the formula used matches the font listed in the *Font to Use* column. *This is a very important step and the barcodes will not be properly created if this step is performed incorrectly.*

    Note: When selecting the font from the pull down menu, please do not use the fonts that begin with the "@" Symbol. Instead, scroll down in the list and select fonts that begin with "IDAutomation".

Select the *Preview* tab (or *File - Print Preview*) to view the barcodes in the report. Print one page of the report and test with a scanner.



(01)15012345678907

DataBar UFL Expanded Stacked Barcode Generation

This procedure inserts a formula field for DataBar Expanded Stacked symbols and is compatible with Crystal Reports version 9 and up. This is a necessary step to ensure that DataBar Expanded Stacked symbols are created properly in Crystal Reports.

1. Install the package and open the Crystal Report example provided.

2. Choose *View - Design* to switch to design mode.

3. Highlight the `@IDAutomationDataBarExpandedStacked` field and choose *Edit - Copy*.

4. Open the report choose *Edit - Paste* to paste the field where it is desired.

5. Highlight the field and choose *Edit - Edit Formula*.

    1. Modify the "`DataToEncode:=`" line of the formula to equal the data that is to be encoded in the barcode.
    2. Change the "`Segments:= 4;`" line of the formula to equal the number of segments needed.
       The complete formula should look something similar to:

       ```
       stringVar DataToEncode:= {Table1.DataField1};
       numberVar Segments:= 6;
       stringVar CompleteBarcodeString:="";
       numberVar i:=0;
       numberVar DataSegments:= ToNumber
       (IDAutomationDataBarExpandedStackedSet(DataToEncode,Segments));
       For i:=1 to DataSegments Do
       (
       CompleteBarcodeString := CompleteBarcodeString +
       IDAutomationDataBarExpandedStackedGet(DataToEncode,Segments,i);
       );
       CompleteBarcodeString
       ```

6. Continue implementation and testing with Step 11 of the tutorial.