Barcode Generator User Guide

Table of Contents

Version

Date	Author	Version	Change
February 8, 2019	Insight Works	1.0	Initial Version
February 5, 2020	Insight Works	1.1	Include examples and 3of9 usage guide

1   Version Information

To check the current version of the product navigate to Extension Management. There you can use the ellipsis to view detailed information, including the version.

2   Basic Barcode Generation

While Barcode Generator is intended as a developer tool there are a number of reports included that may be used immediately. Use cases are setting them in report selections, searching for them manually, or calling them in an extension.

Name	ID	Parent DataItem	Description
Barcode Generation	23044800	Integer	This report allows you to enter any value manually. Since it includes the barcode formats and settings it is a good way to test configuration before creating your own reports.
Generic Item Barcode Label	23044801	Item	This report uses the data matrix format. Items may be filtered by Item fields or Item Ledger Entry fields. A barcode for each Item in the filter with the Item No. will be generated along with a barcode for the Routing No. when applicable.
Generic Bin Label	23044802	Bin	This report uses the data matrix format. Bins may be filtered by Bin fields. For each Bin a barcode will be generated with the Bin Code.
Generic Location Label	23044803	Location	This report uses the data matrix format. Locations may be filtered by Location fields. For each Location a barcode will be generated with the Location Code.

3   Advanced Barcode Generation

The main purpose of this extension is to provide a tool to developers that have a requirement to add barcodes to their own or their client's reports.

3.1   Retrieve Barcode Generation Symbols

Using the product in this way does require that the tenant have Barcode Generator installed. It will then be possible to download the symbols and use the generation functions. Here are steps on how to acquire the symbols.

1. Create an AL project in Visual Studio Code
2. Set the launch.json file in the project to the Microsoft Dynamics 365 Business Central sandbox. Which sandbox it connects to will depend on your login information.

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "type": "al",
            "request": "launch",
            "name": "Microsoft cloud sandbox",
        }
    ]
}
```

3. Include the Barcode Generation extension as a dependency in the dependencies portion of the app.json file.

```
"dependencies": [
    {
        "appId": "62080fe0-d57f-4d4c-aed1-ac539db3a244",
        "publisher": "Insight Works",
        "name": "Barcode Generator",
        "version": "1.1.2.0"
    }
]
```

4. While in that same project use the search window to select "AL: Download Symbols." This should download the symbol file from the sandbox in to the .alpackages folder in the project.
5. You now have access to the Barcode Generation procedures. These may be used in your own reports that can then be deployed as a per tenant extension to your or your client's Microsoft Dynamics 365 Business Central environment.

3.2 Barcode Generation Functions

The procedures are contained in Codeunit 23044800 - IWX Library - Barcode Gen.

3.2.1 GenerateDataMatrixBarcode

This function generates a 2D barcode. The following parameters are in order of the function signature.

Parameter	Type	Input	Comment
precBarcode	var Record "IWX Barcode" temporary	temporary "IWX Barcode" variable	When the function is successful "IWX Barcode".Image will be populated with the barcode image.
ptxtBarcode	Text[1024]	barcode data	The barcode image will be created with the information contained in this string. The format of that string and how it is interpreted when the barcode is scanned will be unique to your needs.
piBarcodeDotSize	Integer	desired dot size	These 3 fields are used by DataMatrix.Net to determine the height and width (in pixels) of the resulting image. The options for the margin side are based on the DataMatrix.Net library.
piBarcodeMarginSize	Integer	desired margin size	
piBarcodeImageSize	Integer	Integer selection from below	width = 2 * this._marginSize + (this._region.SymbolCols * this._moduleSize); height = 2 * this._marginSize + (this._region.SymbolRows * this._moduleSize);
		0	Auto
		1	10x10
		2	12x12
		3	14x14
		4	16x16
		5	18x18
		6	20x20
		7	22x22
		8	24x24

		9	26x26
		10	32x32
		11	36x36
		12	40x40
		13	44x44
		14	48x48
		15	52x52
		16	64x64
		17	72x72
		18	80x80
		19	88x88
		20	96x96
		21	104x104
		22	120x120
		23	132x132
		24	144x144
		25	8x18
		26	8x32
		27	12x26
		28	12x36
		29	16x36
		30	16x48

3.2.2 Generate39Barcode

This function generates a 1D barcode. The following parameters are in order of the function signature.

Parameter	Type	Input	Comment
precBarcode	var Record "IWX Barcode" temporary	temporary "IWX Barcode" variable	When the function is successful "IWX Barcode".Image will be populated with the barcode image.
ptxtBarcode	Text[1024]	barcode data	The barcode image will be created with the information contained in this string. The format of that string and how it is interpreted when the barcode is scanned will be unique to your needs.
piWidth	Integer	desired width in pixels	
piHeight	Integer	desired height in pixels	

3.3 Use Barcode Generation in a Report

These functions return a TempBlob object that has the barcode image set in the Blob field. That blob field may then be used in a DataItem column which will show up where referenced in the report layout.

3.3.1 Report Object

In the report object you would set the value under the relevant DataItem. In the example below '100' would be replaced with whatever string represents the data you would like to have included in the barcode.

```
dataitem(Loop; "Integer")
{
    DataItemTableView = SORTING (Number);
    column(barcodeImage; Barcode.Image)
    {
    }

    trigger OnPreDataItem();
    begin
        SETRANGE(Number, 1, 1);
    end;

    trigger OnAfterGetRecord();
        BarcodeGeneration: Codeunit "IWX Library - Barcode Gen";
    begin
        BarcodeGeneration.Generate39Barcode(Barcode, '100',
         550, 220);
    end;
}

var
        Barcode: Record "IWX Barcode" temporary;
```

3.3.2 Report Layout

Next you will want to include this column in the report layout. To do so you would add an image element. Based on the above example the properties would look like the following.

```
field: =First(Fields! barcodeImage.Value, "DataSet_Result")
MIME: image/bmp
Display: Fit proportional
```