ABarCode

Bar codes for Microsoft® Access™ Reports

versions 2.0 and 7.0

INTRODUCTION

ABarCode is a bar code generator utility for Microsoft Access reports.

The program has been entirely made and designed as a library database, therefore it does not install any additional file to the Windows system, nor need any DDE or OLE function to run.

Versions 2.0 and 7.0 differ as the first one runs under Access 2.0 and Windows 3.1x or 95, while the second one runs under Access 7.0 and Windows 95. The installation process is slightly different.

Any comments, ideas or suggestions are welcome and can be addressed to the author at:

tomasb@abaforum.es (Tomas Boixet)

The following instructions presuppose that the user has some experience in using Microsoft Access, particularly with reports design.

INSTALLING

Version 2.0 (for Access 2.0)

1. Copy the application files to the Microsoft Access directory:
   ```
   ABARCODE.MDA  (attention to the MDA extension)
   ABCINFES.WRI
   ABCREGES.WRI
   ABCINFEN.WRI  (this document)
   ABCREGEN.WRI
   ```

2. Open Microsoft Access and then any database.

3. From the **File** menu, choose **Add-Ins** and then **Add-In Manager**.

4. From the list of **Available Libraries** select **ABarCode** and click **Install**. It will appear an **x** which indicates that the library is installed. Press the **Close** button. If it does not appear **ABarCode** in the list is because the program file, abarcode.mda, can not be found in the Access directory. Revise step 1.

A message is shown prompting you to quit Microsoft Access and re-enter again in order to allow the installed library to be available. Once that it is done, all the functions of ABarCode will be available in any database.

Version 7.0 (for Access 7.0)

1. Copy the application files to the Microsoft Access directory:
   ```
   ABARCODE.MDB  (attention to the MDB extension)
   ABCINFES.WRI
   ABCREGES.WRI
   ABCINFEN.WRI  (this document)
   ABCREGEN.WRI
   ```

2. Open Microsoft Access and the data base where you wish to print bar codes.

3. Click **Design** for any module in the **Modules** tab, or click **New** if there is no one. Select **References** from the **Tools** menu.

4. If **ABARCODE.MDB** does not appear in the **Available References** list, click **Browse** to open the **Add Reference** window, and look for it selecting **Files of type: Databases (*.mdb; *.mda)**. Once it appears in the list, activate it with the corresponding check box and click **OK**.

5. Unlike to the version 2.0, where ABarCode is installed only once, you will need to do the steps 2 to 4 for each database where you want to get bar codes.

WORKING

Let's imagine that you want to draw a bar code symbol for every product in the Northwind catalog:

1. Open the report **Catalog** in **Design** mode.

2. In the **Detail** section, of interest to us, add up a new **TextBox** from the left end to the **ProductName**. This will give us an approximate width of 0.75 " (1.9 cm.), which is quite limited, but useful as an example. The height should be no less than 0.5" (1.3 cm.).

3. In the **Control Source** property of this new **Text Box**, select **ProductId**, which is where the information wanted to be presented in a bar code form is located.

4. Type **abarcode,ean8** in the **Tag** property.

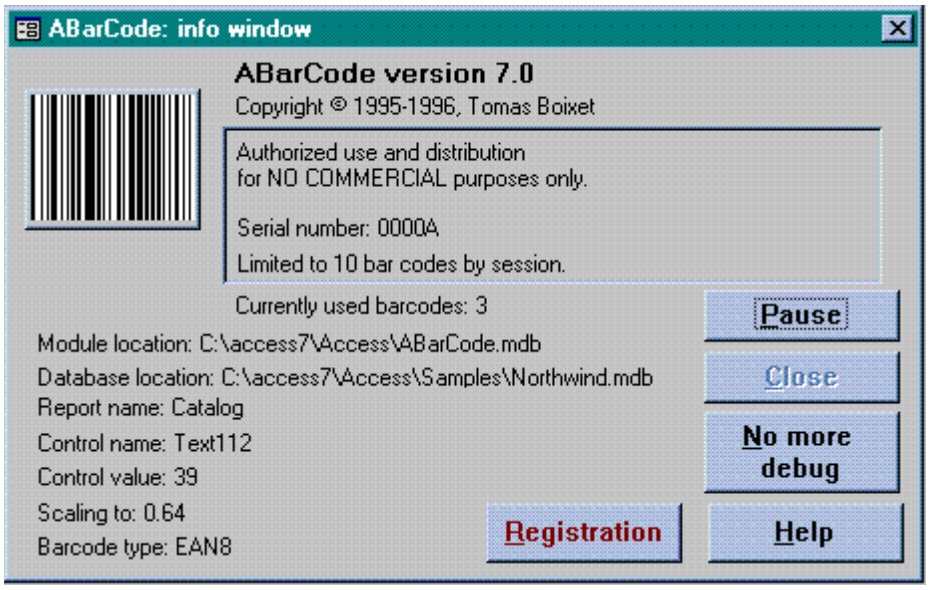
5. Finally, type **=abarcode()** in the **On Format** property of the **Detail** section. If you need to specify an **event procedure** for this property, you can put the **abarcode()** call in any point of this **procedure**. It can also be called from a macro associated to the same event.

Now you can see the report with the bar codes (from page 3). Possibly some of them are split between two pages. In order to prevent that happening, set up the **Keep Together** property of the section **Detail** section as **Yes**.

To save the report we were working on, choose **Save as** from the **File** menu and save the modified report under a different name, like **Catalog BarCoded**.

The information window

Whenever a reports' Preview using bar codes is being prepared, ABarCode shows this window:



The information window can be very useful sometimes:

- To know at what scale the codes are printed: each type of bar code has standard measures defined as factor 1. If the code is too narrow, it could be difficult for the reader or scanner to identify it. In the above example, the scale is 0.65, when the minimum recommended is 0.8. However, with good quality printing and a good reader, it may be readable.

- To know which values are codified: EAN8 codification allows only numeric digits, 7 to be precise. Therefore, the program fills in zeros from the left until seven digits are completed, any extra digits are eliminated from the right. The program also changes into zero any non-numeric characters, and calculates and adds the check digit. The information window shows the original value without changes.

We know now how to create an EAN8. It is important to introduce the key words correctly. If you type **ean-8** (with a dash) instead of **ean8**, the program will not identify it. However, it will recognize them if they are written in upper cases or lower cases, or mixed.

All the available options can be found in the **Keywords** section.

Codification types

This version provides, in addition to **ean8**, **ean13**, **upca**, **itf**, and **code39** available in the previous version, the **code39x** (extended), **code128**, and **postnet** types.

To obtain any of these types, you should introduce the corresponding keyword on the **Tag** property of the text box.

Text

ABarCode uses the values of the **FontName** and **FontSize** properties of the **TextBox** to print the human readable code below the bar code symbol. This function is automatically activated when printing horizontal codes, but it is not possible to activate it when vertical. If you wish an horizontal code without text, insert the **notext** keyword into the **Tag** property.

Another keyword, **textonly**, allows you to print only the human readable code without the bar code. It is normally used to print the text on top or below a vertical bar code: this text will be the one being symbolized, i.e. it can have added or changed characters, the check digit, etc.

Color

The bars are printed in the color specified in the **ForeColor** property, and the background according to the property **BackColor**.

Orientation

The bar codes can be printed in horizontal or vertical (+90º) format. If it is not indicated, ABarCode places it horizontally. To print vertically, you must insert the **vert** keyword in the **Tag** property.

Limitations

It is not possible to print bar code symbols in more than **one section** of the same report. They can be put in any section, several at a time, but do not try to put a bar code in the Detail section and another in the Header section, for example. You could get the most incredible results.

KEYWORDS

This is a list of the keywords which ABarCode recognizes in the **Tag** property of a Text Box. It does not matter in which order are placed, nor if they appear in their upper or lower case, nor if they are separated one from the others by spaces, comas, dashes or if they appear together. If one incorrectly writes "abarcode;upc;ean8", one will not obtain an ean8 as it is the last one which has been written. It will appear a upc, as it is the last one that ABarCode has looked for and found. The importance fall on that ABarCode finds the right chain of characters.

Look at **Specifications** section to obtain more information about the different types of bar codes.

abarcode
Required. When this keyword is found, ABarCode knows that the text box must be processed.

addcd
For **itf** and **code39** types. The program calculates and adds the **check digit**.

bbars
To print the support bars around a **ITF** symbol. Used for expedition units (packaging).

notext
To avoid that ABarCode automatically prints the human readable code under the symbol. It is not necessary when the **vert** option is used.

textonly
Only the human readable code is printed.

vert
Puts the symbol vertically (+90º). This option deactivates the **textonly** option, if it was activated, and automatically activates the **notext** option.

code128, code39, code39x, ean128, ean13, ean8, itf, postnet, upc
These are the available types of bar code symbols.

Note that if you write, for example, **abarcode39**, the program will identify the keywords **abarcode** and **code39**.

SPECIFICATIONS

ABarCode uses the space delimited by the TextBox to place the bar code symbol, including security margins and the readable text. The only exception is on ITF type, where the readable text is printed outside, below the TextBox space.

ABarCode automatically calculates and adds the check digit only for those type of codification which have it as a standard established, so that the symbol would be unreadable for any reader without such digit.

These are some of the specific characteristics of the available bar code types:

Code 128

ABarCode uses the **B** and **C** subsets of this codification system, alternating them within the same code when necessary. That is, subset **B** is used only when there is no other option, even in this case, it goes back to subset **C** if it finds a sequence of four or more numeric digits, obtaining thereafter the maximum possible compression level, as the subset **C** includes two numeric digits in the same space as an alphanumeric one, or any of **B**.

The length is variable, and it is alphanumeric: **ASCII** characters with hexadecimal value between **20** and **7F** can be symbolized. If an invalid character is found, ABarCode converts it to a blank space. Check digit is automatically computed and added.

Because of the high density of characters per inch, it is recommended to care about the good quality of the printed codes.

Code 39 and Code 39x

This system allows to symbolize numbers, uppercase letters, and the characters - . $ / + % and space, without limitation in length. ABarCode put in uppercase any lowercase letter, and converts to a space any invalid character.

The Extended subset, Code 39X, allows any ASCII character between 0 and 127 decimal value to be represented, using two Code 39 positions to symbolize the characters not supported by the standard Code 39.

Code 39 is strongly self checked and most situations do not require a check digit. If a specific application requires exceptional data security, a check digit can be added setting up the **addcd** option.

EAN / UPC

EAN/UPC codes are a fixed length numeric codes using a check digit, and are used mainly for retail applications. You should address to the official organization in your country if you wish to codify your products and they are to be sold to the whole market.

ABarCode replaces any non-numeric character by zero, and make the necessary arrangements to obtain a 11, 12, or 7 numeric digits for UPC, EAN13, or EAN8 respectively: filling with zeroes on the left, or deleting characters on the right.

The check digit is always computed and added by ABarCode.

ITF - Interleaved 2 of 5

This system offers a very high density of numeric characters per inch and does not need a good quality printing, as it only uses two bar width. But because of this facility, you are advised to use it in two ways: as fixed length codes, or as variable length but using a check digit. ABarCode will calculate and add the check digit if indicated with the **addcd** option.

Any non-numeric character is replaced by zero. In any case, and because ITF is symbolizing each pair of digits with 5 bars, the resulting code must be a number with an even length. Therefore, ABarCode may add a leading zero digit if necessary.

Postnet

The U.S. postal service uses this bar code symbols to automatically sort mail using zip codes. The length may be 5, 9 or 11 digits, plus a check digit.

ABarCode takes out all the numeric characters from the source field, removing any non-numeric, then adjusting their length and computing the check digit. This allows you to use, for example, a source field containing the city, state, and zip code:

> “OREM, UT, 84059 - 9908”

ABarCode will codify the zip code “840599908”, compute and add the check digit (“8”), and then put the result as a postnet bar code symbol (“8705999088”).

SHAREWARE

ABarCode is distributed as SHAREWARE: you can try it, and if you like it, you can purchase it. This version is limited to 20 bar codes in one database session, however the symbols are fully readable as in the registered version. The only difference is that the registered version has not got the limitation mentioned above.

If after evaluating the program you are interested, please print out the Registration/Order form from file ABCREGEN.WRI, fill it in and send it attaching the required payment. You will receive a registered copy of ABarCode, by e-mail or by regular mail, as you wish.

You can also register ABarCode and pay for it by credit card through the RegNet service:

- Internet: http://www.xmission.com/~wintrnx/regnet/regnet.htm
- USA toll-free number: 1-800-999-2734
- Fax number: 801-531-0621
- International (24 h.): 801-355-5110

You can make as many copies of ABarCode as you like, and freely distribute it, following this conditions:

- The program and its documentation can not be altered in any form, nor any of its components eliminated.

- The distribution of this program is free of charge, except for the costs of postage and packing, or the normal costs of connection to the BBS or network.

THE REGISTERED COPY OF ABARCODE INCLUDES A LICENSE ONLY FOR THE PURCHASER'S USE. UNDER NO CIRCUMSTANCES IT CAN BE COPIED, DISTRIBUTED OR USED IN MORE THAN ONE COMPUTER.

Additional stuff

ABarCode now includes a function that can be useful for many users, and is not subject to the shareware limitations: that is, it can be used when and wherever you want, as well as copied, distributed, etc. Everything apart for sell it.

This function, “ABCedNUM”, accepts an integer number of up to 9 digits and returns a string with it textual representation.

Syntax:

abcednum(source, language)

where ‘language’ may be:

“en”:	convert to English text (i.e. thirty one)
“es”:	convert to Spanish text, feminine. (i.e. treinta y una)
“esm”:	convert to Spanish text, masculine, ending (i.e. treinta y uno)
“esn”:	convert to Spanish text, masculine, without ending (i.e. treinta y un)