

PrecisionID PDF417 and DataMatrix ActiveX Control User Manual

Notice: When you use this product, you agree to the End User License Agreement (EULA). The EULA is provided as a file in the package for this product. To view the license on-line, please visit www.PrecisionID.com/licenses/

Limitations of the demo version: The demo version of this product may be used for evaluation purposes only. In the demo version, the demo watermark is displayed and a pop-up message may appear upon initialization. To obtain a successful scan, the margins may be increased to move the watermark outside of the barcode symbol. All other functions are exactly the same as the purchased version. If you are using the demo version and you would like to order, please visit: <http://www.precisionid.com/>

| | |
|---|-------------------|
| Product Overview..... | 1 |
| Compatibility..... | 1 |
| Installation..... | 1 |
| Microsoft Windows..... | 2 |
| Tutorials for Specific Applications..... | 2 |
| Microsoft Access..... | 2 |
| Microsoft Excel..... | 3 |
| Microsoft Visual Basic 6..... | 4 |
| Properties and Defaults..... | 6 |
| Data Matrix Properties (All preceded with DataMatrix):..... | 6 |
| PDF417 Properties (All preceded with PDF):..... | 6 |
| MACRO PDF417 Properties (All preceded with PDF):..... | 7 |

Product Overview

PrecisionID ActiveX Controls may be used in applications compatible with ActiveX graphic objects such as Microsoft Access, Excel and Visual Basic. Once installed, the properties of the object may be modified as necessary to create high quality, vector-based bar code images. Supported barcode types in the 2D package include PDF417 with Text and Binary Compaction, Macro PDF417 and Data Matrix ECC200 with C40, Text, ASCII and Byte Compaction.

The PrecisionID ActiveX Control is a high performance component created in C++ using ATL. After installation, it should reside in the Windows/System32 directory and be registered as a COM object. It has no dependencies and may be distributed in your application with a Developer's License.

Compatibility

PrecisionID ActiveX Controls may be installed on Windows 95 or greater operating systems including Windows 98, ME, 2000, 2003 Server, Windows XP and Windows Vista with 128 MB or more of RAM. They have been tested and are compatible with Office 97, Office 2000, Office XP, Office 2003 and Office 2007.

Installation

Microsoft Windows

Decompress the package and run the supplied **Setup.exe** file to install the ActiveX Control. After installation, it will be registered as an available ActiveX Control in Windows and may be used by any application that uses ActiveX Controls.

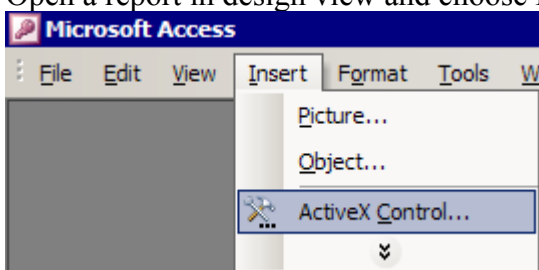
Tutorials for Specific Applications

The results of the following tutorials are saved in the *examples* folder of the product zip file. We encourage you to refer to the examples for additional assistance. These tutorials use Data Matrix. However, PDF417 could also be used if desired by following the same procedures.

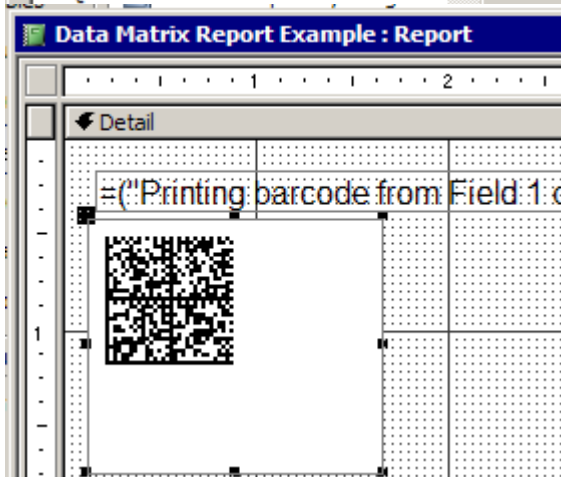
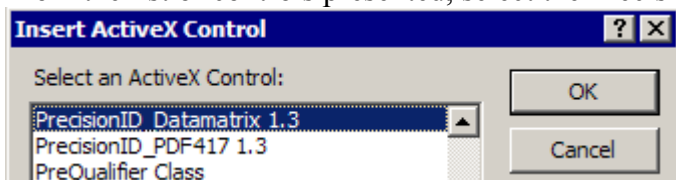
Microsoft Access

To create a barcode in a Microsoft Access report:

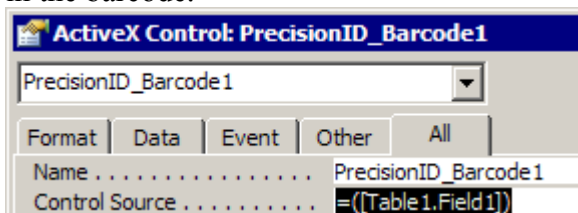
1. Run the Setup program to install the ActiveX Control, if it has not been installed.
2. Open a report in design view and choose Insert – ActiveX Control.



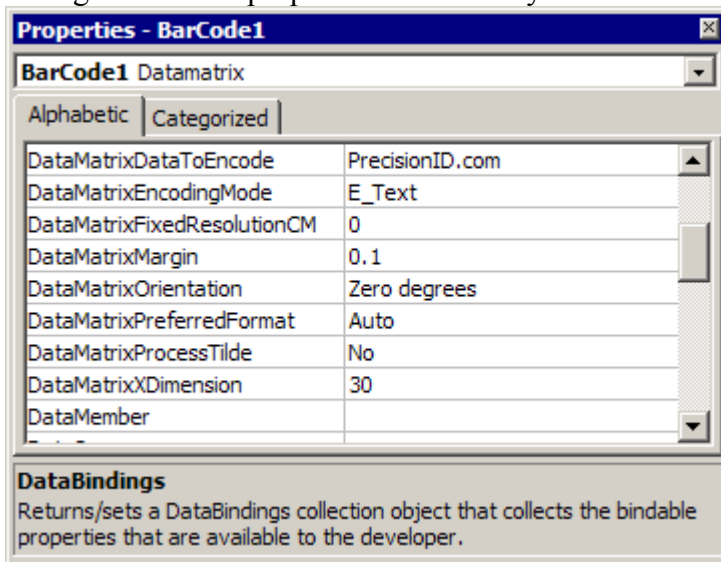
3. From the list of controls presented, select the PrecisionID ActiveX Control.



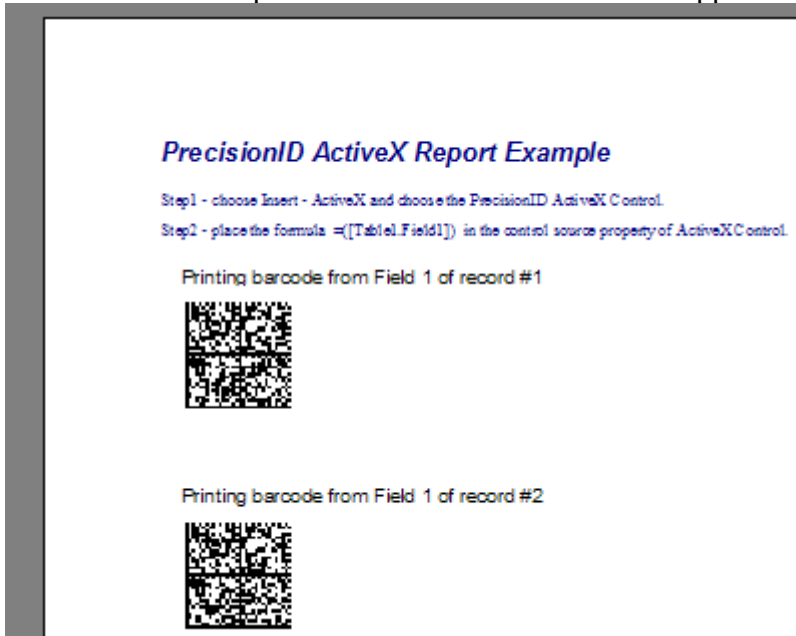
4. After the control is placed on the report, right-click on it and choose Properties.
5. Modify the Control Source property to point to the table and field of the data you wish to encode in the barcode.



- Change additional properties as necessary such as the X Dimension and the Encoding Mode.

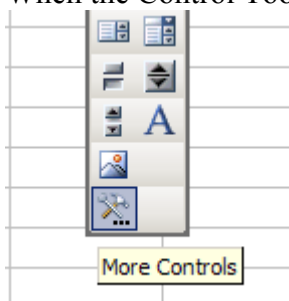


- Save and run the report. You should see the barcode appear in the report.



Microsoft Excel

- Run the Setup program to install the ActiveX Control, if it has not been installed.
- In Excel, choose View – Toolbars – Control Toolbox.
- When the Control Toolbox appears, choose the More Controls button.



- From the list of controls presented, select the PrecisionID ActiveX Control.

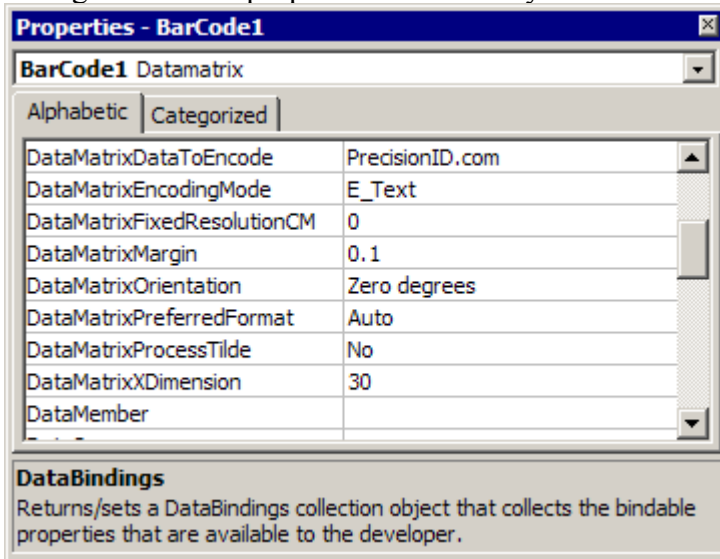
5. Select the area to place the control in the spreadsheet.



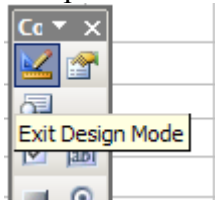
6. Right-click on the control, choose Properties and change the Linked Cell Property to the cell of text data you wish to encode in the barcode.

| | |
|------------|------|
| Height | 45 |
| Left | 126 |
| LinkedCell | A6 |
| Locked | True |

7. Change additional properties as necessary such as the X Dimension and the Encoding Mode.

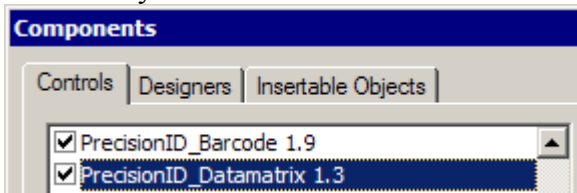


8. After editing the properties, choose the Exit Design Mode button and the barcode will appear in the spreadsheet.



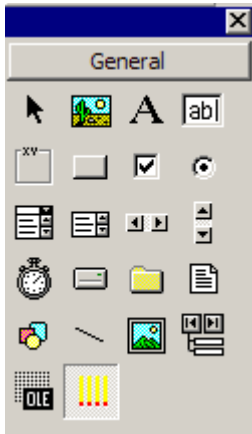
Microsoft Visual Basic 6

1. Run the Setup program to install the ActiveX Control, if it has not been installed.
2. In Visual Basic 6, choose Project – Components. From the list of available controls, select the control of your choice and choose OK.



If the Toolbox is not visible, choose View – Toolbox. The control should now appear at the

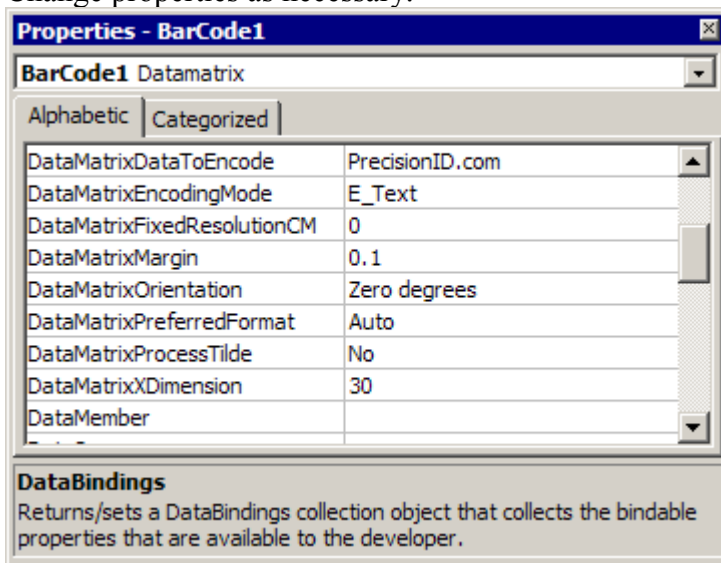
bottom of the Toolbox as a yellow and red bar code object.



3. Select the control and place it on a form as you would any other control.



4. After the control is placed on the form, right-click on it and choose Properties.
5. Change properties as necessary.



6. Using code in the form, you may set the data to be encoded and send the barcode to the printer, a file or to the clipboard.

The following code determines the data to be encoded:

```
Datamatrix1.DataMatrixDataToEncode = "Data Matrix ECC200 Barcode Symbol"
```

The following will print the barcode:

```
Printer.PaintPicture Datamatrix1.Picture, 2048, 1024
```

The following will copy the barcode to the clipboard:

```
Clipboard.Clear
```

```
Clipboard.SetData Datamatrix1.Picture, vbCFMetafile
```

The following will save the barcode to a file:

```
Datamatrix1.SaveBarCode "DataMatrix_ECC200_.wmf"
```

Properties and Defaults

All measurements are in inches unless otherwise specified.

Data Matrix Properties (All preceded with DataMatrix):

| Property | Default Value | Description |
|-------------------|---------------|---|
| DataToEncode | na | The text or binary data to be encoded. |
| EncodingMode | BASE256 | Determines the method used to encode data, which may reduce symbol size depending on the data that is encoded: C40 – encodes uppercase letters and numbers best Text – encodes text characters best ASCII – encodes ASCII characters best BASE256 – encodes bytes of data (Default) |
| FixedResolutionCM | 0 | Determines the resolution the symbol is drawn to. Leave this at the default setting of 0 which is automatic resolution mode. |
| Margin | 0.1 | The margin around the symbol. |
| Orientation | 0 | The orientation of the symbol. |
| PreferredFormat | Auto | Determines the smallest symbol size. However, if the data encoded creates a larger symbol, the size will automatically increase. |
| ProcessTilde | No | If enabled, ~d??? will be converted to an ASCII character where ??? is a 3 digit number. For example, ~d009 may be used for a tab function and ~d013 for a return. |
| XDimension | 30 | The X Dimension in MILS where 1 MIL equals 1/1000 of an inch. |

PDF417 Properties (All preceded with PDF):

| Property | Default Value | Description |
|------------------------|---------------|--|
| Columns | 3 | The number of columns in the symbol. Generally, 10 to 15 is a good number to use when increasing the data above 20 characters. |
| DataToEncode | na | The text or binary data to be encoded. |
| ErrorCorrectionLevel | 2 | The level of error correction in the symbol. A larger number will allow damaged symbols to be read easier and increase symbol size. |
| FixedResolutionCM | 0 | Determines the resolution the symbol is drawn to. Leave this at the default setting of 0, which is automatic resolution mode. |
| LeftMargin & TopMargin | 0.01 | The margins around the symbol. |
| Mode | Binary | Determines the method used to encode data which may reduce symbol size: Text – encodes text characters including ASCII 9 & ASCII 13 Binary – encodes bytes of data (Default) |
| Orientation | 0 | The orientation of the symbol. |
| ProcessTilde | 1 (True) | When enabled, ~??? will be converted to an ASCII character where ??? is a 3 digit number. For example, ~009 may be used for a tab function and ~013 for a return. |
| Rows | 0 | This should usually be left at 0. Determines the smallest symbol size according to the number of rows. |

| | | |
|------------|----|---|
| XDimension | 15 | The X Dimension in MILS where 1 MIL equals 1/1000 of an inch. |
|------------|----|---|

MACRO PDF417 Properties (All preceded with PDF):

MacroPDF is an implementation of PDF417 capable of encoding large amounts of data into multiple PDF417 barcodes that are combined by scanning equipment when read.

| Property | Default Value | Description |
|-----------------------|----------------------|--|
| MacroPDFEnable | 0 (False) | When true, indicates that this barcode is part of a MacroPDF sequence. |
| MacroPDFFileID | 0 | Assigns a file id to the barcode. Each barcode in the sequence must have the same file id assigned to it. Default is 0; valid options are 0-899. |
| MacroPDFSegment Index | 0 | The index number of this barcode in relation to the set. Default is 0; valid options are 0-99998; each barcode in the sequence must have a unique segment index. |
| MacroPDFLast Segment | 0 (False) | When true, indicates that this is the final barcode in the MacroPDF sequence. |