DataMatrix

Table of contents

also known as: ISO/IEC 16022:2000(E)

1 Example

2 Structure

The configuration for the default implementation is:

```
<barcode>
  <datamatrix>
      <module-width>{length:0.352777mm}</module-width> <!-- 1 pixel at 72dpi -->
      <quiet-zone enabled="{boolean:true}">{length:1mw}</quiet-zone>
      <shape>{shape:force-none}</shape>
      <min-symbol-size>{dimension}</min-symbol-size>
      <max-symbol-size>{dimension}</max-symbol-size>
  </datamatrix>
</barcode>
```

3 Notes

- This symbology has no human-readable part!
- The algorithm always chooses the smallest possible symbol arrangement.
- Structured append functionality is not implemented, yet.
- "min-symbol-size" and "max-symbol-size" are both optional and can be used to restrict the size of the generated DataMatrix symbols. If you omit either one, there's no constraint for either bigger or smaller symbols. This feature is particularly useful if you want to generate DataMatrix symbols for the German Post. They require the symbols to be either 22x22 or 26x26, depending on the message size. Since Barcode4J usually tries to always generate the smallest possible symbol, you might actually end up with a 24x24 symbol if the message can be encoded very efficiently. Specifying "26x26" for both "min-symbol-size" and "max-symbol-size" ensures that the symbol is always the right size. If the message doesn't fit inside a symbol that matches these constraints, an error will be generated. If the symbol is much larger than necessary to encode the message, padding codewords are automatically added. These padding codewords don't affect the decoded message.

4 Message format

- All ISO-8859-1 characters are valid message characters.
- Using only numeric characters allows for smaller symbol sizes.
- Currently, no ECI functionality is available. Only characters in the "ISO-8859-1" encoding can be used.
- Currently, the FNC1 and reader programming signal cannot be encoded.
- Some applications use special ASCII characters like <GS> (group separator) or <RS> (record separator). Just send them as is to Barcode4J. In Java a preamble of such an application

("[)>RS05GS") can be expressed as "[)>\u001E05\u001D". The same encoded as part of an URL (when using the barcode servlet) will be "%5B%29%3E%1E05%1D".

- Binary data can be supplied through URLs if they are enclosed in "url()". RFC 2397 data URLs can be used to encode inline data. An example to encode the text "~Test~": `url(data:;base64,flRlc3R+)` or `url(data:text/plain;charset=iso-8859-1,%7ETest%7E)` (the "charset="iso-8859-1" is important to get characters above the 7bit US-ASCII set correctly!).