

barValid gs1 decoder library

manual



manual version:
19.1
barValid library version
19.1
Implements GS1 version
Release 19.1, Ratified, Jul 2019

Table of content

Document history	2
Introduction	3
Contacts	3
Supported barcodes types.....	3
Functionality of the barValid decoder library	3
Preconditions for the barcode string.....	6
Decoder details	6
Decoder result	7
Decoder validation Problems.....	8
Usage examples.....	8
JavaScript examples	8
Using node REPL	8
Legal notices	10
Legal Statement.....	10
Third party licenses	10

Document history

Version	Date	Description
19.1	07.11.2019	Initial version of the barValid decoder library manual

Introduction

This technical document describes the usage of the barValid gs1 decoder engine.

The barValid gs1 decoder parses a GS1 formatted barcode string with the aim to detect nested application identifiers and their corresponding content values.

Depending on the package you are using, the barValid gs1 decoder engine comes as a source code file(s) or in binary format.

The most recent version of the library implements the following official GS1 standard:

GS1 General Specifications

Release 19.1, Ratified, Jul 2019

Contacts

In case of questions, suggestions or any other concerns you can contact barValid by mail:

info.barValid@gmail.com

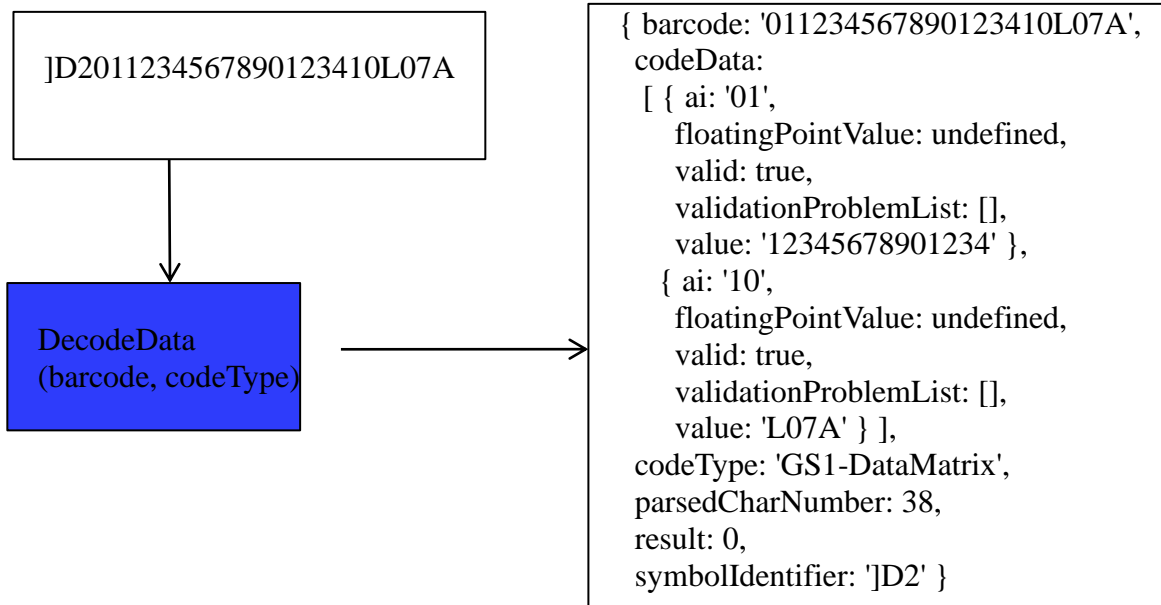
Supported barcodes types

The library supports natively the following barcode types:

- GS1-128
- GS1 DataBar
- GS1-DataMatrix
- GS1-QR-Code

Functionality of the barValid decoder library

The library exposes a function, which expects as input parameter the barcode as a string and returns the decoding result as a JSON dictionary:



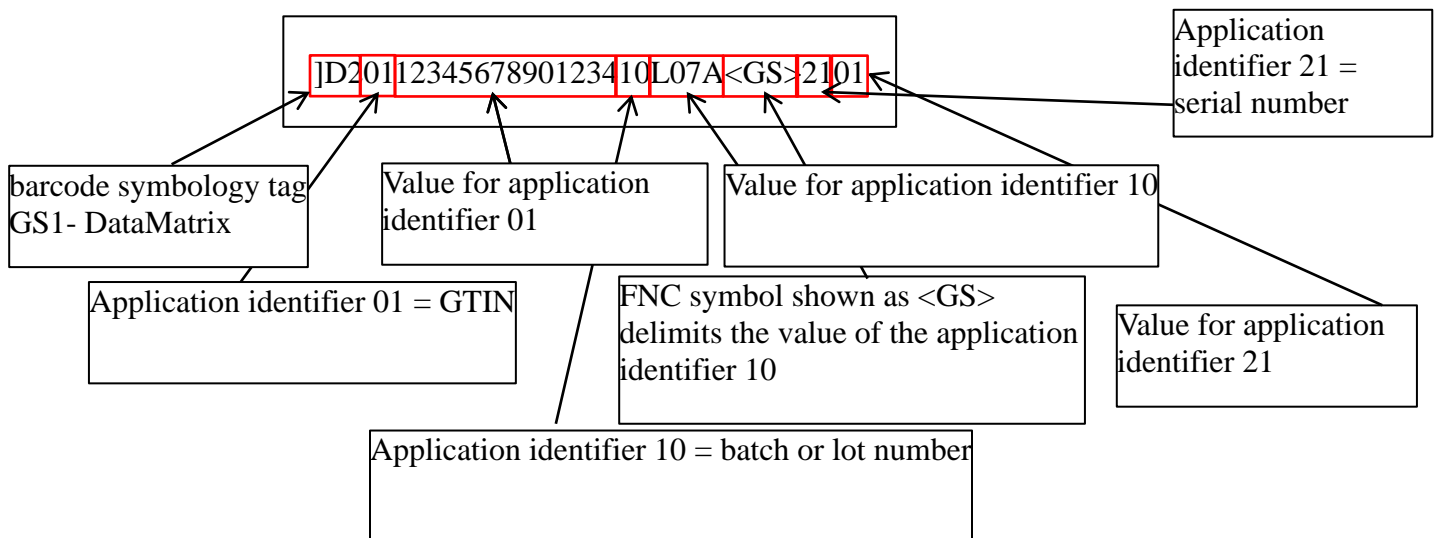
Preconditions for the barcode string

The barcode string must reflect the value of the barcode, which has been readen by a Barcode Scanner component (the Barcode Scanner component is not part of this library).

GS1 barcodes contain, next to the readable characters, also control characters. The most important character is the so-called FNC1 symbol. Normally barcode scanners decode the FNC1 symbol as the ASCII-29 **GS** character. The FNC symbol is used inside GS1 barcodes as a stop symbol for variable-length parts.

Some (but not all) Barcode Scanner components deliver also the barcode symbology tag. The

barcode symbology reflects the type of a given barcode (see section **Supported barcodes types**).



Decoder details

The barValid library implements an optimistic decoding of GS1 barcodes: It tries to interpret the barcode even if the structure is not fully complaint to the GS1 standard:

- application identifiers might be included into brackets:
 - `(10)LA05` is interpreted as `10LA05`
- For application identifiers with variable length, which are not terminated by the FNC symbol, it is assumed they occupy the maximum length
- barcode are decoded, even if not all characters can be assigned to an application identifier:
 - `151905213` is interpreted as `15190521`

Decoder result

The decoding function will return a JSON document containing the result. The JSON document is described with the following example for the input barcode string “[D2011234567890123410L07A”:

```
{
  // barcode without the barcode symbology tag
  barcode: '011234567890123410L07A',
  // a JSON array, which contains an item for each recognized application identifier
  codeData:
  [
    {
      // label of the recognized application identifier
      ai: '01',
      // floating point value of the content (depends from application identifier)
      floatingPointValue: undefined,
      // true, if content does not contain errors (see validationProblemList)
      valid: true,
      // JSON array, containing the problems identified while decoding the content of the given application identifier
      // see section Decoder validation Problems for a list of possible validationProblems.
      validationProblemList: [],
      // unformatted raw value of the given application identifier
      value: '12345678901234'
    },
    {
      ai: '10',
      floatingPointValue: undefined,
      valid: true,
      validationProblemList: [],
      value: 'L07A'
    }
  ],
  // recognized barcode type
  codeType: 'GS1-DataMatrix',
  // number of successfully processed characters
  parsedCharNumber: 16,
  // overall decoding result
  // 0 = successful; 1 = problem(s) detected; 2 = possible problem(s) detected
  result: 1,
  // symbol identifier
```

```
symbolIdentifier: 'JD2'  
}
```

Decoder validation Problems

The following list reflects the possible problems detected by the decoder engine:

Problem const String	description
LBL_VALID_PROBLEM_FPV_VIOLATION	Unable to convert the content into a floating-point value
LBL_VALID_PROBLEM_PARSE_INT	Unable to convert the content into an integer
LBL_VALID_PROBLEM_FIXED_LENGTH_VIOLATION	The content's length does not follow the specification of the given application identifier

Usage examples

JavaScript examples

Using node REPL

You can import the barValid JavaScript library easily into a Node REPL session and test out the library:

```
> var bV = require('./ barValidBarcodeParser.min');
undefined
> bV.decodeData(']D2011234567890123410L07A', 'DATA_MATRIX');
{ barcode: '011234567890123410L07A',
  codeData:
    [ { ai: '01',
      floatingPointValue: undefined,
      valid: true,
      validationProblemList: [],
      value: '12345678901234' },
      { ai: '10',
      floatingPointValue: undefined,
      valid: true,
      validationProblemList: [],
      value: 'L07A' } ],
  codeType: 'GS1-DataMatrix',
  parsedCharNumber: 38,
  result: 0,
  symbolIdentifier: ']D2' }
>
```


Legal notices

Legal Statement

The information contained in this document is subject to change without notice. barValid makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. barValid shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damage in connection with the furnishing, performance, or use of this material and the barValid applications.

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

The barValid gs1 decoder library, the barValid app and other related items are protected by copyright. All rights are reserved.

A license holder is allowed to use the barValid gs1 decoder library, but it is prohibited to sell, distribute or reproduce the applications or parts of the applications to third parties.

Copyright © 2019 barValid, Harald Kofler

Third party licenses

The GS1 mark and all related items are protected by copyright of the GS1 AISBL association. Visit <https://www.gs1.org/terms-use> for details.