

---

---

**Information technology — Automatic  
identification and data capture  
techniques — MicroPDF417 bar code  
symbology specification**

*Technologies de l'information — Techniques d'identification  
automatique et de capture des données — Spécifications pour la  
symbologie de code à barres MicroPDF417*



**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2006

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

|                                                                       |     |
|-----------------------------------------------------------------------|-----|
| Foreword.....                                                         | vi  |
| Introduction .....                                                    | vii |
| 1 Scope .....                                                         | 1   |
| 2 Normative references .....                                          | 1   |
| 3 Terms and definitions .....                                         | 2   |
| 4 Symbols, operations and abbreviated terms .....                     | 4   |
| 4.1 Symbols .....                                                     | 4   |
| 4.2 Mathematical operations.....                                      | 5   |
| 4.3 Abbreviated terms .....                                           | 5   |
| 5 Requirements .....                                                  | 5   |
| 5.1 Symbology characteristics .....                                   | 5   |
| 5.1.1 Basic characteristics .....                                     | 5   |
| 5.1.2 Summary of additional features .....                            | 6   |
| 5.2 Symbol structure .....                                            | 7   |
| 5.2.1 MicroPDF417 symbol parameters.....                              | 7   |
| 5.2.2 Row and column combinations.....                                | 9   |
| 5.2.3 Row parameters .....                                            | 10  |
| 5.2.4 Codeword sequence.....                                          | 10  |
| 5.2.5 MicroPDF417 Row Address Patterns.....                           | 12  |
| 5.3 Basic encodation .....                                            | 14  |
| 5.3.1 Symbol character structure .....                                | 14  |
| 5.3.2 Start and Stop Patterns .....                                   | 15  |
| 5.4 High level (data) encodation.....                                 | 15  |
| 5.4.1 Function codewords.....                                         | 15  |
| 5.4.2 Text Compaction mode .....                                      | 21  |
| 5.4.3 Byte Compaction mode.....                                       | 25  |
| 5.4.4 Numeric Compaction mode .....                                   | 27  |
| 5.4.5 Advice to select the appropriate compaction mode .....          | 28  |
| 5.4.6 Treatment of MicroPDF417 reserved codewords .....               | 29  |
| 5.5 Extended Channel Interpretation .....                             | 29  |
| 5.5.1 Encoding the ECI assignment number.....                         | 30  |
| 5.5.2 Pre-assigned and default Extended Channel Interpretations ..... | 31  |
| 5.5.3 Encoding ECI sequences within compaction modes .....            | 31  |
| 5.5.4 Post-decode protocol .....                                      | 33  |
| 5.6 Determining the codeword sequence.....                            | 33  |
| 5.7 Error detection and correction .....                              | 34  |
| 5.7.1 Number of error correction codewords.....                       | 34  |
| 5.7.2 Error correction capacity .....                                 | 34  |
| 5.7.3 Defining the error correction codewords .....                   | 35  |
| 5.8 Dimensions.....                                                   | 35  |
| 5.8.1 Minimum width of a module (X).....                              | 35  |
| 5.8.2 Row height (Y).....                                             | 35  |
| 5.8.3 Quiet zones.....                                                | 35  |
| 5.9 Defining the symbol format .....                                  | 35  |
| 5.9.1 Defining the aspect ratio of the module .....                   | 36  |
| 5.9.2 Defining the symbol matrix of rows and columns .....            | 36  |
| 5.10 Generating the error correction codewords .....                  | 37  |
| 5.11 Low level encodation .....                                       | 39  |
| 5.11.1 Clusters.....                                                  | 40  |

|                     |                                                                                        |     |
|---------------------|----------------------------------------------------------------------------------------|-----|
| 5.11.2              | Determining the symbol matrix .....                                                    | 40  |
| 5.11.3              | Determining the values of the Row Address Patterns .....                               | 40  |
| 5.11.4              | Row encoding.....                                                                      | 46  |
| 5.12                | Printing Row Address Patterns .....                                                    | 46  |
| 5.13                | Structured Append.....                                                                 | 47  |
| 5.13.1              | Compaction modes and Structured Append.....                                            | 47  |
| 5.13.2              | ECIs and Structured Append .....                                                       | 47  |
| 5.14                | User guidelines .....                                                                  | 47  |
| 5.14.1              | Human readable interpretation.....                                                     | 47  |
| 5.14.2              | Autodiscrimination capability.....                                                     | 47  |
| 5.14.3              | User-defined application parameters.....                                               | 48  |
| 5.14.4              | MicroPDF417 symbol quality .....                                                       | 48  |
| 5.14.5              | Separation of multiple symbols.....                                                    | 49  |
| 5.15                | Reference decode algorithm.....                                                        | 49  |
| 5.16                | Error detection and error correction procedure .....                                   | 49  |
| 5.17                | Transmitted data .....                                                                 | 49  |
| 5.17.1              | Transmitted data in the basic (default) interpretation.....                            | 49  |
| 5.17.2              | Transmission protocol for Extended Channel Interpretation (ECI) .....                  | 49  |
| 5.17.3              | Transmitted data for Structured Append.....                                            | 51  |
| 5.17.4              | Transmission of reserved codewords using the ECI protocol.....                         | 51  |
| 5.17.5              | Symbology identifier.....                                                              | 51  |
| 5.17.6              | Transmission using older protocols .....                                               | 51  |
| Annex A (normative) | Encoding/decoding table of PDF417 symbol character bar-space sequences.....            | 52  |
| Annex B (normative) | The default character set for Byte Compaction mode .....                               | 67  |
| Annex C (normative) | Byte Compaction mode encoding algorithm .....                                          | 68  |
| Annex D (normative) | Numeric Compaction mode encoding algorithm.....                                        | 70  |
| Annex E (normative) | Error correction .....                                                                 | 72  |
| Annex F (normative) | Tables of coefficients for calculating MicroPDF417 error correction<br>codewords ..... | 73  |
| Annex G (normative) | Text Compaction mode encoding algorithm.....                                           | 77  |
| Annex H (normative) | Structured Append MicroPDF417 symbols .....                                            | 78  |
| H.1                 | Structured Append overview .....                                                       | 78  |
| H.2                 | Structured Append syntax .....                                                         | 78  |
| H.3                 | High level encoding considerations .....                                               | 81  |
| H.4                 | Encodation example .....                                                               | 81  |
| H.5                 | Structured Append and the Extended Channel Interpretation protocol .....               | 82  |
| H.6                 | Structured Append data transmission.....                                               | 83  |
| Annex I (normative) | Testing MicroPDF417 symbol quality.....                                                | 86  |
| I.1                 | Overview of methodology .....                                                          | 86  |
| I.2                 | Test scans for scan reflectance profile .....                                          | 86  |
| Annex J (normative) | Reference decode algorithm for MicroPDF417.....                                        | 88  |
| J.1                 | Phase 1: Initialization.....                                                           | 88  |
| J.2                 | Phase 2: Filling the matrix.....                                                       | 89  |
| J.3                 | Phase 3: Interpretation .....                                                          | 90  |
| J.4                 | Reference line-decode algorithm .....                                                  | 91  |
| Annex K (normative) | Error correction procedures .....                                                      | 95  |
| Annex L (normative) | Symbology identifier .....                                                             | 97  |
| Annex M (normative) | Transmission protocol for decoders conforming with original<br>PDF417 standards.....   | 98  |
| M.1                 | Basic Channel Mode .....                                                               | 98  |
| M.2                 | GLI encoded symbols.....                                                               | 98  |
| M.3                 | Structured Append symbols.....                                                         | 100 |
| M.4                 | Transmission of reserved codewords using the original PDF417 protocol .....            | 101 |

|                              |                                                                                                              |            |
|------------------------------|--------------------------------------------------------------------------------------------------------------|------------|
| <b>M.5</b>                   | <b>Achieving compatibility between old and new PDF417 equipment .....</b>                                    | <b>101</b> |
| <b>Annex N</b> (informative) | <b>Algorithm to minimise the number of codewords .....</b>                                                   | <b>104</b> |
| <b>Annex O</b> (informative) | <b>Guidelines to determine the symbol matrix .....</b>                                                       | <b>106</b> |
| <b>Annex P</b> (informative) | <b>Calculating the coefficients for generating the error correction<br/>codewords — worked example .....</b> | <b>107</b> |
| <b>Annex Q</b> (informative) | <b>Generating the error correction codewords — worked example .....</b>                                      | <b>109</b> |
| <b>Annex R</b> (informative) | <b>Division circuit procedure for generating error correction codewords .....</b>                            | <b>112</b> |
| <b>Annex S</b> (informative) | <b>Additional guidelines for the use of MicroPDF417 .....</b>                                                | <b>113</b> |
| <b>S.1</b>                   | <b>Autodiscrimination compatibility .....</b>                                                                | <b>113</b> |
| <b>S.2</b>                   | <b>Pixel-based printing .....</b>                                                                            | <b>113</b> |
| <b>Bibliography</b> .....    |                                                                                                              | <b>115</b> |

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24728 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic Identification and data capture techniques*.

This International Standard contains many provisions which are identical with those of ISO/IEC 15438.

## Introduction

MicroPDF417 is a multi-row symbology, derived from and closely based on PDF417. MicroPDF417 is designed for applications with a need for improved area efficiency but without the requirement for PDF417's maximum data capacity. A limited set of symbol sizes is available, together with a fixed level of error correction for each symbol size. Module dimensions are user-specified to enable symbol production and reading by a wide variety of techniques.

Since MicroPDF417's data character encodation, its error correction method, and many of its other symbol characteristics are, and are intended to remain, identical to those of PDF417, descriptions of these characteristics are quoted verbatim from the PDF417 symbology specification (ISO/IEC 15438) wherever appropriate, or with the appropriate modifications. For ease of cross-reference, this International Standard follows a similar document structure, with minor differences in clause/subclause numbering, to ISO/IEC 15438.



# Information technology — Automatic identification and data capture techniques — MicroPDF417 bar code symbology specification

## 1 Scope

This International Standard specifies the requirements for the bar code symbology known as MicroPDF417. It specifies the MicroPDF417 symbology characteristics, data character encodation, symbol formats, dimensions, error correction rules, decoding algorithm, and a number of application parameters.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 15415, *Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Two-dimensional symbols*

ISO/IEC 15417, *Information technology — Automatic identification and data capture techniques — Bar code symbology specification — Code 128*

ISO/IEC 15418, *Information technology — EAN/UCC Application Identifiers and Fact Data Identifiers and Maintenance*

ISO/IEC 15424, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

ISO/IEC 24723, *Information technology — Automatic identification and data capture techniques — EAN.UCC Composite bar code symbology specification*

ISO/IEC 19762-1, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms relating to AIDC*

ISO/IEC 19762-2, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 2: Optically readable media (ORM)*

AIM Inc. International Technical Standard: ITS/04-001, *Extended Channel Interpretations — Part 1: Identification Schemes and Protocols*<sup>1</sup>

GS1 General Specification<sup>2</sup>

<sup>1</sup> Published by AIM Global, 125 Warrendale-Bayne Road, Suite 100, Warrendale, PA 15086, USA.

<sup>2</sup> Published by GS1, Blue Tower, Avenue Louise 326, bte 10, B-1050 Brussels, Belgium.

### 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762-1, ISO/IEC 19762-2 and the following apply.

**3.1  
application identifier**  
sequence of 2 to 4 digits, used to define the nature or use of the subsequent data characters, in accordance with ISO/IEC 15418 and the GS1 General Specification

**3.2  
bar-space sequence**  
sequence which represents the module widths of the elements of a symbol character

**3.3  
basic channel mode**  
standard system for encoding and transmitting bar code data where data message bytes are output from the decoder but no control information about the message is transmitted

**3.4  
cluster**  
any of three mutually exclusive subsets of PDF417 symbol characters, also used in MicroPDF417; the symbol characters in a given cluster conform with particular structural rules which are used in decoding the symbology

**3.5  
compaction mode**  
any of three data compaction algorithms in PDF417 (Text, Numeric and Byte Compaction modes), also used in MicroPDF417, which are used to map 8-bit data bytes efficiently to PDF417 codewords

**3.6  
e-distance**  
measurement from the leading edge of one element to the leading edge of the next element, or from the trailing edge of an element to the trailing edge of the next element, in a symbol character or Row Address Pattern

**3.7  
error correction codeword**  
codeword which encodes a value derived from the error correction codeword algorithm to enable decode errors to be detected and, depending on the number of error correction codewords, to be corrected

**3.8  
Extended Channel Interpretation  
ECI**  
procedure within some symbologies, including MicroPDF417, to replace the default interpretation with another interpretation in a reliable manner

NOTE The interpretation intended prior to producing the symbol can be retrieved after decoding the scanned symbol to recreate the data message in its original format.

**3.9  
extended channel model**  
system for encoding and transmitting both data message bytes and control information about the message, the control information being communicated using Extended Channel Interpretation (ECI) escape sequences

**3.10  
family**  
set of MicroPDF417 symbol versions sharing the same number of columns and the same RAP rotation

**3.11****function codeword**

codeword which initiates a particular operation within a symbology, for example to switch between data encoding sets, to invoke a compaction scheme, to program the reader, or to invoke Extended Channel Interpretations

**3.12****Global Label Identifier****GLI**

procedure in the PDF417 symbology which behaves in a similar manner to Extended Channel Interpretation

NOTE The GLI system was the symbology-dependent precursor to the symbology-independent ECI system.

**3.13****macro character**

character representing a predefined sequence of data characters (e.g. an industry-specific message header) used to reduce the number of symbol characters needed to encode data in a symbol using certain structured formats

**3.14****Mode Latch codeword**

codeword which is used to switch from one mode to another mode, which stays in effect until another latch or shift codeword is implicitly or explicitly brought into use, or until the end of the symbol is reached

**3.15****Mode Shift codeword**

codeword which is used to switch from one mode to another for one codeword, after which encoding returns to the original mode

**3.16****RAP rotation**

difference between the number designating a Center or Right Row Address Pattern and the number designating the nearest Row Address Pattern to the left, in the same row of a symbol

**3.17****Row Address Patterns**

special patterns made up of three bars and three spaces occupying ten modules that serve both as start or (with the stop bar) stop patterns and as row indicators in MicroPDF417 symbols

**3.18****stop bar**

single-module bar, adjoining the rightmost Row Address Pattern, which forms the right boundary of the symbol

**3.19****Structured Append**

procedure within the MicroPDF417 symbology to distribute data logically from a computer file across a number of related symbols

NOTE 1 This procedure is identical to the Macro PDF417 feature of PDF417.

NOTE 2 The procedure considerably extends the data capacity beyond that of a single symbol.

**3.20****UCC/EAN-128**

subset of Code 128 symbols as defined in ISO/IEC 15417, reserved for use in accordance with GS1 General Specification

## 4 Symbols, operations and abbreviated terms

### 4.1 Symbols

For the purposes of this document, the following mathematical symbols apply. There are some cases where the symbols below have been used in a different manner in an equation. This has been done for consistency with a more general use of the notation and is always clearly defined in the text.

|       |                                                                                                                                        |
|-------|----------------------------------------------------------------------------------------------------------------------------------------|
| $b$   | the element width in a symbol character                                                                                                |
| $c$   | number of columns in the symbol in the data region (excluding Row Address Patterns)                                                    |
| $d$   | data codeword including ECI Descriptor and all function codewords                                                                      |
| $E$   | error correction codeword                                                                                                              |
| $e$   | an edge to similar edge dimension in a symbol character                                                                                |
| $f$   | number of substitution errors                                                                                                          |
| $F$   | row number                                                                                                                             |
| $H$   | height of symbol including quiet zone                                                                                                  |
| $K$   | cluster number                                                                                                                         |
| $k$   | number of error correction codewords                                                                                                   |
| $L$   | number of erasures                                                                                                                     |
| $m$   | number of source data codewords (including the codeword in the ECI Descriptor position but prior to the addition of any pad codewords) |
| $n$   | total number of data codewords (including ECI Descriptor and pad codewords)                                                            |
| $p$   | pitch or width of a symbol character                                                                                                   |
| $Q_H$ | horizontal quiet zone                                                                                                                  |
| $Q_V$ | vertical quiet zone                                                                                                                    |
| $r$   | number of rows in the symbol                                                                                                           |
| $W$   | width of symbol including quiet zone                                                                                                   |
| $X$   | X-dimension or module width                                                                                                            |
| $Y$   | module height (also called row height)                                                                                                 |

## 4.2 Mathematical operations

For the purposes of this document, the following mathematical operations apply.

|     |                                                                                                                                                                                                                                                                      |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| div | is the integer division operator, rounding down                                                                                                                                                                                                                      |
| INT | is the integer value i.e. where a number is rounded down to its whole number component, ignoring its decimal fractions                                                                                                                                               |
| mod | is the positive integer remainder after division. If the remainder is negative, then add the value of the divisor in order to make the result positive. For example, the remainder of $-29\ 160$ divided by $929$ is $-361$ , which when added to $929$ yields $568$ |

## 4.3 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

|     |                                 |
|-----|---------------------------------|
| AI  | Application Identifier          |
| EC  | Error Correction                |
| ECI | Extended Channel Interpretation |
| GLI | Global Label Identifier         |
| RAP | Row Address Pattern             |

# 5 Requirements

## 5.1 Symbology characteristics

### 5.1.1 Basic characteristics

MicroPDF417 is a multi-row symbology which may be utilized by applications needing to encode a moderate amount of data in a two-dimensional symbol (up to 150 bytes, 250 alphanumeric characters, or 366 numeric digits), and when minimizing symbol size is a primary concern. MicroPDF417 is identical to PDF417 in terms of its encodation modes, error correction method, and symbol character sets. However, MicroPDF417 replaces PDF417's 17-module-wide start/stop patterns and left/right row indicators with a unique set of 10-module-wide Row Address Patterns, which were designed both to reduce overall symbol width and to facilitate linear scanning at row heights as low as 2X. MicroPDF417, unlike PDF417, may only be printed in certain defined combinations of  $r$  (number of rows),  $c$  (number of columns), and  $k$  (number of error correction codewords), up to a maximum of four data columns by 44 rows (see 5.2.2).

MicroPDF417 has the following basic characteristics:

a) Encodable character set:

- 1) Text Compaction mode (see 5.4.2) permits all printable ASCII characters to be encoded, i.e. values 32 to 126 inclusive in accordance with ISO/IEC 646, as well as selected control characters.
- 2) Byte Compaction mode (see 5.4.3) permits all 256 possible 8-bit byte values to be encoded. This includes all ASCII characters value 0 to 127 inclusive and provides for international character set support.
- 3) Numeric Compaction mode (see 5.4.4) permits efficient encoding of numeric data strings.
- 4) Up to 811 800 different character sets or data interpretations.
- 5) Various function codewords for control purposes.

- b) Symbol character structure:  $(n, k, m)$  characters of 17 modules ( $n$ ), 4 bar and 4 space elements ( $k$ ), with the largest element 6 modules wide ( $m$ ).
- c) Maximum possible number of data characters per symbol (for a maximum size MicroPDF417 symbol): 125 data codewords which can encode
  - 1) Text Compaction Mode: 250 characters (2 data characters per codeword).
  - 2) Byte Compaction mode: 150 characters (1,2 data characters per codeword).
  - 3) Numeric Compaction mode: 366 characters (2,93 data characters per codeword).
- d) Symbol Size:
  - 1) Number of rows: 4 to 44 (available in defined combinations with number of columns).
  - 2) Number of data columns: either one, two, three, or four
  - 3) Width in modules: 40X, 57X, 84X, or 101X including quiet zones
  - 4) Maximum codeword capacity: 176 codewords.
  - 5) Maximum data codeword capacity: 125 codewords.
- e) Number of error correction codewords: fixed for each available row/column combination, ranging from 7 to 50 codewords per symbol and reserving from 28% to 67% of codewords for error detection and correction, depending on symbol size.
- f) Non-data overhead: per row: 23 modules for the one- and two-column versions; 33 modules for the three- or four-column version, including quiet zones.
- g) Code type: continuous, multi-row bar code symbology
- h) Character self-checking: Yes.
- i) Bi-directionally decodable: Yes.

### 5.1.2 Summary of additional features

The following are additional features in MicroPDF417:

- a) **Data compaction:** Three schemes are defined to compact a number of data characters into codewords. Generally data is not directly represented on a one character for one codeword basis (see 5.4.2 to 5.4.4).
- b) **Extended Channel Interpretations:** These mechanisms allow up to 811 800 different data character sets or interpretations to be encoded (see 5.5).
- c) **Structured Append:** MicroPDF417 uses the Macro PDF417 mechanism for Structured Append. This mechanism allows files of data to be represented logically and consecutively in a number of MicroPDF417 symbols. Up to 99 999 different MicroPDF417 symbols can be so linked or concatenated and be scanned in any sequence to enable the original data file to be correctly reconstructed (see 5.13).
- d) **Edge to edge decodable:** MicroPDF417 can be decoded by measuring elements from edge to similar edge (see 5.3.1).
- e) **Cross-row scanning:** The combination of three characteristics in MicroPDF417 facilitates cross-row scanning:
  - being synchronised horizontally, or self clocking
  - row identification
  - being vertically synchronised, by using the cluster values to achieve local row discrimination.

This combination allows a single linear scan to cross a number of rows and achieve a partial decode of the data so long as at least one complete symbol character per row is decoded into its codeword. The decoding algorithm can then place the individual codewords into a meaningful matrix.

- f) **Code 128 emulation codewords:** special codewords may be used at the start of a MicroPDF417 symbol as a signal to the decoder to transmit the symbol's data as if it were encoded in a Code 128 symbol. The Symbology Identifier of the resulting transmission can thus signal the presence of an implied FNC1 flag, for special applications (see 5.4.1.5).

## 5.2 Symbol structure

### 5.2.1 MicroPDF417 symbol parameters

Each MicroPDF417 symbol consists of a stack of vertically-aligned rows (with a minimum of 4 and maximum of 44 rows); the allowable numbers of rows are specified separately for each of the one-, two-, three- or four-column versions. Each row shall include a minimum of 1 symbol character and a maximum of 4 symbol characters, excluding the Row Address Pattern columns. The symbol shall include a quiet zone on all four sides.

Figure 1 shows sample one-, two-, three-, and four-column MicroPDF417 symbols, each with twenty rows.

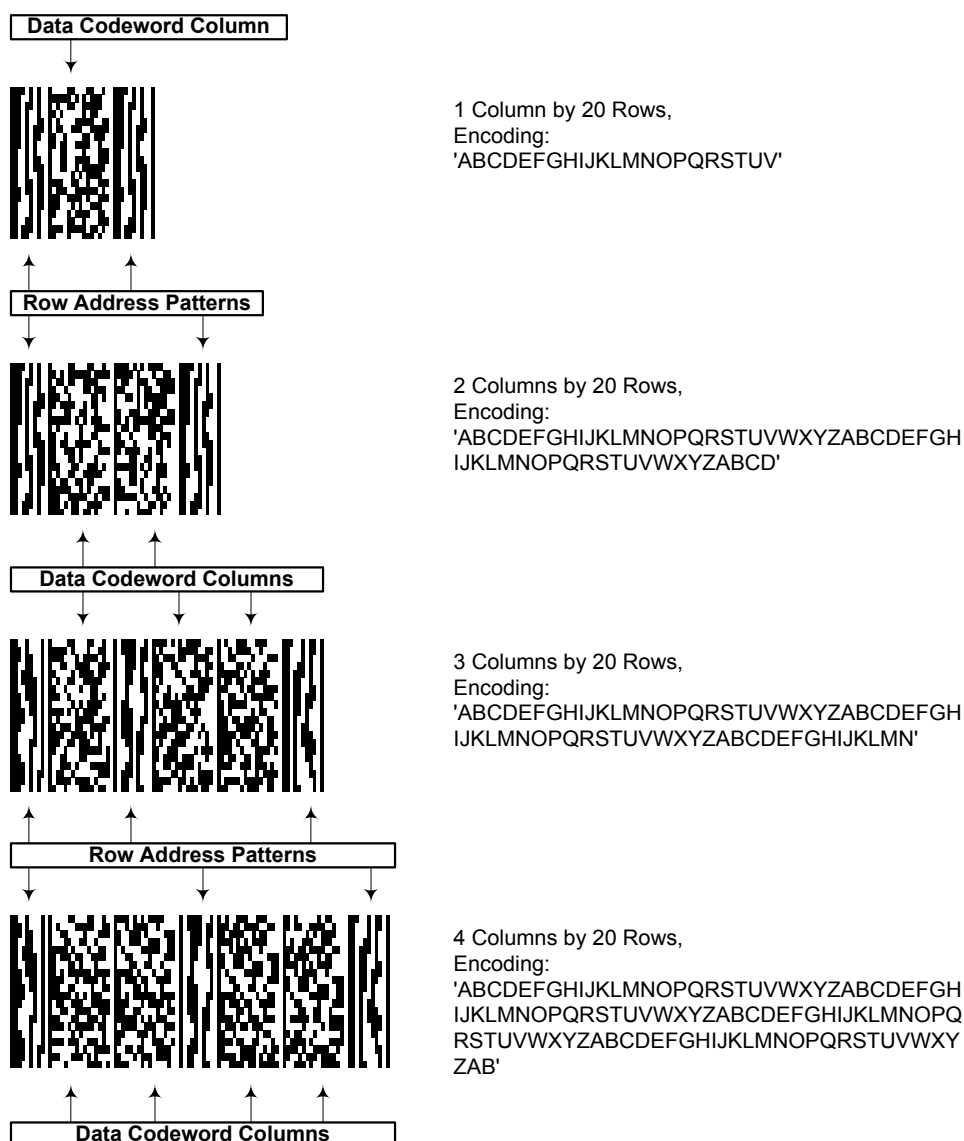


Figure 1 — Four MicroPDF417 symbols

### 5.2.2 Row and column combinations

MicroPDF417 symbols shall conform with certain predefined combinations of numbers of rows, columns, and number of error-correction codewords. These versions are defined in Table 1.

**Table 1 — MicroPDF417 version characteristics**

| Number of Data Columns (c) | Number of Rows (r) | Total CWs in Data Region | Number of EC CWs (k) | % of CWs for EC | Number of non-EC CWs | Number of CWs for Data (Note 1) | Max Data Bytes (Note 2) | Max Alpha chars (Note 3) | Max Digits (Note 3) | Symbol Width, in X (Note 4) | Symbol Height, in X (Note 5) |
|----------------------------|--------------------|--------------------------|----------------------|-----------------|----------------------|---------------------------------|-------------------------|--------------------------|---------------------|-----------------------------|------------------------------|
| 1                          | 11                 | 11                       | 7                    | 64              | 4                    | 3                               | 3                       | 6                        | 8                   | 40                          | 24                           |
| 1                          | 14                 | 14                       | 7                    | 50              | 7                    | 6                               | 7                       | 12                       | 17                  | 40                          | 30                           |
| 1                          | 17                 | 17                       | 7                    | 41              | 10                   | 9                               | 10                      | 18                       | 26                  | 40                          | 36                           |
| 1                          | 20                 | 20                       | 8                    | 40              | 12                   | 11                              | 13                      | 22                       | 32                  | 40                          | 42                           |
| 1                          | 24                 | 24                       | 8                    | 33              | 16                   | 15                              | 18                      | 30                       | 44                  | 40                          | 50                           |
| 1                          | 28                 | 28                       | 8                    | 29              | 20                   | 19                              | 22                      | 38                       | 55                  | 40                          | 58                           |
| 2                          | 8                  | 16                       | 8                    | 50              | 8                    | 7                               | 8                       | 14                       | 20                  | 57                          | 18                           |
| 2                          | 11                 | 22                       | 9                    | 41              | 13                   | 12                              | 14                      | 24                       | 35                  | 57                          | 24                           |
| 2                          | 14                 | 28                       | 9                    | 32              | 19                   | 18                              | 21                      | 36                       | 52                  | 57                          | 30                           |
| 2                          | 17                 | 34                       | 10                   | 29              | 24                   | 23                              | 27                      | 46                       | 67                  | 57                          | 36                           |
| 2                          | 20                 | 40                       | 11                   | 28              | 29                   | 28                              | 33                      | 56                       | 82                  | 57                          | 42                           |
| 2                          | 23                 | 46                       | 13                   | 28              | 33                   | 32                              | 38                      | 64                       | 93                  | 57                          | 48                           |
| 2                          | 26                 | 52                       | 15                   | 29              | 37                   | 36                              | 43                      | 72                       | 105                 | 57                          | 54                           |
| 3                          | 6                  | 18                       | 12                   | 67              | 6                    | 5                               | 6                       | 10                       | 14                  | 84                          | 14                           |
| 3                          | 8                  | 24                       | 14                   | 58              | 10                   | 9                               | 10                      | 18                       | 26                  | 84                          | 18                           |
| 3                          | 10                 | 30                       | 16                   | 53              | 14                   | 13                              | 15                      | 26                       | 38                  | 84                          | 22                           |
| 3                          | 12                 | 36                       | 18                   | 50              | 18                   | 17                              | 20                      | 34                       | 49                  | 84                          | 26                           |
| 3                          | 15                 | 45                       | 21                   | 47              | 24                   | 23                              | 27                      | 46                       | 67                  | 84                          | 32                           |
| 3                          | 20                 | 60                       | 26                   | 43              | 34                   | 33                              | 39                      | 66                       | 96                  | 84                          | 42                           |
| 3                          | 26                 | 78                       | 32                   | 41              | 46                   | 45                              | 54                      | 90                       | 132                 | 84                          | 54                           |
| 3                          | 32                 | 96                       | 38                   | 40              | 58                   | 57                              | 68                      | 114                      | 167                 | 84                          | 66                           |
| 3                          | 38                 | 114                      | 44                   | 39              | 70                   | 69                              | 82                      | 138                      | 202                 | 84                          | 78                           |
| 3                          | 44                 | 132                      | 50                   | 38              | 82                   | 81                              | 97                      | 162                      | 237                 | 84                          | 90                           |
| 4                          | 4                  | 16                       | 8                    | 50              | 8                    | 7                               | 8                       | 14                       | 20                  | 101                         | 10                           |
| 4                          | 6                  | 24                       | 12                   | 50              | 12                   | 11                              | 13                      | 22                       | 32                  | 101                         | 14                           |
| 4                          | 8                  | 32                       | 14                   | 44              | 18                   | 17                              | 20                      | 34                       | 49                  | 101                         | 18                           |
| 4                          | 10                 | 40                       | 16                   | 40              | 24                   | 23                              | 27                      | 46                       | 67                  | 101                         | 22                           |
| 4                          | 12                 | 48                       | 18                   | 38              | 30                   | 29                              | 34                      | 58                       | 85                  | 101                         | 26                           |
| 4                          | 15                 | 60                       | 21                   | 35              | 39                   | 38                              | 45                      | 76                       | 111                 | 101                         | 32                           |
| 4                          | 20                 | 80                       | 26                   | 33              | 54                   | 53                              | 63                      | 106                      | 155                 | 101                         | 42                           |
| 4                          | 26                 | 104                      | 32                   | 31              | 72                   | 71                              | 85                      | 142                      | 208                 | 101                         | 54                           |
| 4                          | 32                 | 128                      | 38                   | 30              | 90                   | 89                              | 106                     | 178                      | 261                 | 101                         | 66                           |
| 4                          | 38                 | 152                      | 44                   | 29              | 108                  | 107                             | 128                     | 214                      | 313                 | 101                         | 78                           |
| 4                          | 44                 | 176                      | 50                   | 28              | 126                  | 125                             | 150                     | 250                      | 366                 | 101                         | 90                           |

CW = Codeword; EC = Error correction

NOTE 1 Excludes EC codewords and codeword in the ECI Descriptor position

NOTE 2 Assumes remaining in default Byte Compaction Mode (see 5.4)

NOTE 3 Assumes that a Latch to Text or Numeric Compaction mode replaces the ECI Descriptor

NOTE 4 Including 1X quiet zones on either side

NOTE 5 Assumes  $Y = 2X$ ; includes 1X quiet zones at top and bottom

### 5.2.3 Row parameters

Each row within a MicroPDF417 symbol has the following structure (examples are shown in Figure 1):

1. a leading Quiet Zone (see 5.8.3);
2. a Left MicroPDF417 Row Address Pattern (see 5.2.5);
3. one of the following:
  - a) for the one-column version, one PDF417 codeword, or
  - b) for the two-column version, two PDF417 codewords, or
  - c) for the three-column version, one PDF417 codeword followed by a Center Row Address Pattern (see 5.2.5) and two more PDF417 codewords, or
  - d) for the four-column version, two PDF417 codewords followed by a Center Row Address Pattern and two more PDF417 codewords;
4. a Right MicroPDF417 Row Address Pattern;
5. a one-module stop bar; (see 5.2.5);
6. a trailing Quiet Zone (see 5.8.3).

All MicroPDF417 symbols contain at least two Row Address Pattern columns (separated from each other by one or two data columns), and every codeword column is adjacent to at least one Row Address Pattern column.

### 5.2.4 Codeword sequence

#### 5.2.4.1 Overall sequence

A MicroPDF417 symbol may contain up to 176 symbol characters or codewords. Symbol character is the more appropriate term to refer to the printed bar/space pattern; codeword is more appropriate for the numeric value of the symbol character. The codewords shall follow this sequence:

- a. The first codeword in a MicroPDF417 symbol may have a special meaning as an ECI Descriptor (see 5.2.4.2).
- b. The data codewords shall follow, from the most significant encodable character. Codewords with values above 899 may be inserted to change the compaction mode, to perform other special functions, or to indicate ECIs.
- c. Pad codewords as needed to enable the codeword sequence to be represented in the number of rows and columns specified for the selected MicroPDF417 version (usually the smallest of sufficient capacity).
- d. An optional Structured Append Control Block.
- e. Error correction codewords for error detection and correction.

The codewords are arranged with the most significant codeword at the upper left corner of the matrix, and are encoded from left to right, top row to bottom. In the three- and four-column versions, the center Row Address Pattern does not change the codeword sequence (that is, the first three or four codewords of the sequence are placed left to right in the first row, the Center Row Address Pattern being inserted before the last two codewords in the row; the next three or four codewords are placed in the same manner in the second row, and so on). Figure 2 illustrates in layout format the sequence for a 14-row, two-column MicroPDF417 symbol.

|                        |                 |                 |                         |
|------------------------|-----------------|-----------------|-------------------------|
| Left RAP <sub>1</sub>  | d <sub>18</sub> | d <sub>17</sub> | Right RAP <sub>1</sub>  |
| Left RAP <sub>2</sub>  | d <sub>16</sub> | d <sub>15</sub> | Right RAP <sub>2</sub>  |
| Left RAP <sub>3</sub>  | d <sub>14</sub> | d <sub>13</sub> | Right RAP <sub>3</sub>  |
| Left RAP <sub>4</sub>  | d <sub>12</sub> | d <sub>11</sub> | Right RAP <sub>4</sub>  |
| Left RAP <sub>5</sub>  | d <sub>10</sub> | d <sub>9</sub>  | Right RAP <sub>5</sub>  |
| Left RAP <sub>6</sub>  | d <sub>8</sub>  | d <sub>7</sub>  | Right RAP <sub>6</sub>  |
| Left RAP <sub>7</sub>  | d <sub>6</sub>  | d <sub>5</sub>  | Right RAP <sub>7</sub>  |
| Left RAP <sub>8</sub>  | d <sub>4</sub>  | d <sub>3</sub>  | Right RAP <sub>8</sub>  |
| Left RAP <sub>9</sub>  | d <sub>2</sub>  | d <sub>1</sub>  | Right RAP <sub>9</sub>  |
| Left RAP <sub>10</sub> | d <sub>0</sub>  | e <sub>8</sub>  | Right RAP <sub>10</sub> |
| Left RAP <sub>11</sub> | e <sub>7</sub>  | e <sub>6</sub>  | Right RAP <sub>11</sub> |
| Left RAP <sub>12</sub> | e <sub>5</sub>  | e <sub>4</sub>  | Right RAP <sub>12</sub> |
| Left RAP <sub>13</sub> | e <sub>3</sub>  | e <sub>2</sub>  | Right RAP <sub>13</sub> |
| Left RAP <sub>14</sub> | e <sub>1</sub>  | e <sub>0</sub>  | Right RAP <sub>14</sub> |

where:

RAP<sub>j</sub> is the appropriate Left or Right Row Address Pattern for row j of this symbol version

d<sub>18</sub> = first codeword (ECI Descriptor or function codeword)

d<sub>17</sub> - d<sub>0</sub> = encoded data, together with pad codewords if required

e<sub>8</sub> - e<sub>0</sub> = Error Correction codewords

**Figure 2 — MicroPDF417 example of symbol layout schematic**

The rules and advice for structuring the matrix are included in 5.9.

#### 5.2.4.2 ECI Descriptor codeword

Because MicroPDF417 is intended for applications where symbol size is critical, it is expected that MicroPDF417 will frequently be used in conjunction with a Transformation ECI appropriate for the data requirements of the application. MicroPDF417 supports efficient encoding of a single Transformation ECI sequence at the start of each data stream by defining an optional ECI Descriptor codeword that can be encoded at the start of the first symbol of each MicroPDF417-encoded data stream. This Transformation ECI co-exists with the Interpretative ECI (whether default or explicitly encoded) currently in effect, since Interpretative ECIs and Transformation ECIs have independent scopes.

If the input to the MicroPDF417 encoder calls for a Transformation ECI in the range 000900 to 001799 to be encoded at the start of the symbol (or at the start of the first symbol of a Structured Append series), then the MicroPDF417 encoder may encode that ECI in shortened form as a single data codeword (range 0 to 899), as the first codeword in the data region. This codeword is calculated as (ECI\_number MOD 900). Otherwise the

MicroPDF417 encoder shall encode a codeword of 900 or greater as the first codeword, operating as a standard MicroPDF417 function codeword (see 5.4.1).

**EXAMPLE** A Transformation ECI 001002 needs to precede the data to be encoded. It can be efficiently encoded at the start of a MicroPDF417 symbol using the single codeword 102 (because 102 is the third codeword of the standard PDF417 representation [926] [0] [102] of that ECI. (See 5.5.1)).

Where no ECI needs to be encoded at the start of the symbol and the data is most efficiently represented in Text Compaction mode, the encoder should place a codeword 900 (Latch to Text Compaction) at the start of the symbol. Since the codeword is above 899, it will not be interpreted as an ECI Descriptor.

If the first codeword has a value less than 900, then once the MicroPDF417 decoder has successfully performed error correction, it shall, for the purposes of high-level decoding and transmission, interpret this codeword as if it were the third codeword of a Transformation ECI sequence (whose first two codewords are 926, 0). However, if the symbol is the second or later symbol within a Structured Append series, then this special interpretation shall not apply (see 5.13).

### 5.2.5 MicroPDF417 Row Address Patterns

MicroPDF417 Row Address Patterns replace the PDF417 Start and Stop patterns and Left and Right Row Indicators. These patterns are placed at the left and right ends of each row. A Center Row Address Pattern appears after the first codeword (reading left to right) of each row of a three-column MicroPDF417 symbol or the second codeword of a four-column MicroPDF417 symbol. The Row Address Patterns indicate each row number. Unlike the PDF417 Row Indicators, the Row Address Patterns do not explicitly encode the number of rows ( $r$ ), the number of columns ( $c$ ), nor the number of error correction codewords ( $k$ ). For this reason, MicroPDF417 only supports fixed combinations of  $r$ ,  $c$ , and  $k$  as specified in 5.2.2.

The Row Address Patterns in a MicroPDF417 symbol are (10,3) symbol characters, where 10 shows each character consists of ten modules, and 3 shows that it comprises three bars and three spaces. These (10,3) characters create two distinct pattern sets (one set is used for the left and right patterns, and the other set is used for the center patterns); each pattern encodes a pattern number from 1 to 52. The Row Address Patterns always begin with a bar and end with a space. There is a one-module stop bar after the rightmost space of the Right Row Address Pattern.

The sets of MicroPDF417 Row Address Patterns are arranged in an order such that vertically adjacent patterns only differ by a single one-module edge shift. Each of the 52 Pattern Numbers, as shown in Table 2, has two distinct patterns associated with it: one pattern when used as a Left or Right Row Address Pattern, and another pattern when used as a Center Row Address Pattern.

Table 2 — MicroPDF417 Row Address Patterns

| Pattern<br>Number | L or R |   |   |   |   |   |  | C |   |   |   |   |   | Pattern<br>Number | L or R |   |   |   |   |   |  | C |   |   |   |   |   |
|-------------------|--------|---|---|---|---|---|--|---|---|---|---|---|---|-------------------|--------|---|---|---|---|---|--|---|---|---|---|---|---|
|                   | B      | S | B | S | B | S |  | B | S | B | S | B | S |                   | B      | S | B | S | B | S |  | B | S | B | S | B | S |
| 1                 | 2      | 2 | 1 | 3 | 1 | 1 |  | 1 | 1 | 2 | 2 | 3 | 1 | 27                | 2      | 2 | 1 | 1 | 1 | 3 |  | 1 | 1 | 3 | 2 | 2 | 1 |
| 2                 | 3      | 1 | 1 | 3 | 1 | 1 |  | 1 | 2 | 1 | 2 | 3 | 1 | 28                | 2      | 2 | 1 | 1 | 2 | 2 |  | 1 | 1 | 3 | 2 | 1 | 2 |
| 3                 | 3      | 1 | 2 | 2 | 1 | 1 |  | 1 | 2 | 2 | 1 | 3 | 1 | 29                | 2      | 2 | 1 | 1 | 3 | 1 |  | 1 | 1 | 3 | 1 | 2 | 2 |
| 4                 | 2      | 2 | 2 | 2 | 1 | 1 |  | 1 | 3 | 1 | 1 | 3 | 1 | 30                | 2      | 2 | 1 | 2 | 2 | 1 |  | 1 | 2 | 2 | 1 | 2 | 2 |
| 5                 | 2      | 1 | 3 | 2 | 1 | 1 |  | 1 | 3 | 1 | 2 | 2 | 1 | 31                | 2      | 2 | 2 | 1 | 2 | 1 |  | 1 | 3 | 1 | 1 | 2 | 2 |
| 6                 | 2      | 1 | 4 | 1 | 1 | 1 |  | 1 | 3 | 2 | 1 | 2 | 1 | 32                | 3      | 1 | 2 | 1 | 2 | 1 |  | 1 | 3 | 1 | 1 | 1 | 3 |
| 7                 | 2      | 2 | 3 | 1 | 1 | 1 |  | 1 | 4 | 1 | 1 | 2 | 1 | 33                | 3      | 2 | 1 | 1 | 2 | 1 |  | 1 | 2 | 2 | 1 | 1 | 3 |
| 8                 | 3      | 1 | 3 | 1 | 1 | 1 |  | 1 | 4 | 1 | 2 | 1 | 1 | 34                | 2      | 3 | 1 | 1 | 2 | 1 |  | 1 | 1 | 3 | 1 | 1 | 3 |
| 9                 | 3      | 2 | 2 | 1 | 1 | 1 |  | 1 | 4 | 2 | 1 | 1 | 1 | 35                | 2      | 3 | 1 | 1 | 1 | 2 |  | 1 | 1 | 2 | 2 | 1 | 3 |
| 10                | 4      | 1 | 2 | 1 | 1 | 1 |  | 1 | 3 | 3 | 1 | 1 | 1 | 36                | 2      | 2 | 2 | 1 | 1 | 2 |  | 1 | 1 | 2 | 2 | 2 | 2 |
| 11                | 4      | 2 | 1 | 1 | 1 | 1 |  | 1 | 3 | 2 | 2 | 1 | 1 | 37                | 2      | 1 | 3 | 1 | 1 | 2 |  | 1 | 1 | 2 | 3 | 1 | 2 |
| 12                | 3      | 3 | 1 | 1 | 1 | 1 |  | 1 | 3 | 1 | 3 | 1 | 1 | 38                | 2      | 1 | 2 | 2 | 1 | 2 |  | 1 | 1 | 2 | 3 | 2 | 1 |
| 13                | 2      | 4 | 1 | 1 | 1 | 1 |  | 1 | 2 | 2 | 3 | 1 | 1 | 39                | 2      | 1 | 2 | 2 | 2 | 1 |  | 1 | 1 | 1 | 4 | 2 | 1 |
| 14                | 2      | 3 | 2 | 1 | 1 | 1 |  | 1 | 2 | 3 | 2 | 1 | 1 | 40                | 2      | 1 | 2 | 1 | 3 | 1 |  | 1 | 1 | 1 | 3 | 3 | 1 |
| 15                | 2      | 3 | 1 | 2 | 1 | 1 |  | 1 | 2 | 4 | 1 | 1 | 1 | 41                | 2      | 1 | 2 | 1 | 2 | 2 |  | 1 | 1 | 1 | 3 | 2 | 2 |
| 16                | 3      | 2 | 1 | 2 | 1 | 1 |  | 1 | 1 | 5 | 1 | 1 | 1 | 42                | 2      | 1 | 2 | 1 | 1 | 3 |  | 1 | 1 | 1 | 2 | 3 | 2 |
| 17                | 4      | 1 | 1 | 2 | 1 | 1 |  | 1 | 1 | 4 | 2 | 1 | 1 | 43                | 2      | 1 | 1 | 2 | 1 | 3 |  | 1 | 1 | 1 | 2 | 2 | 3 |
| 18                | 4      | 1 | 1 | 1 | 2 | 1 |  | 1 | 1 | 4 | 1 | 2 | 1 | 44                | 2      | 1 | 1 | 1 | 2 | 3 |  | 1 | 1 | 1 | 1 | 3 | 3 |
| 19                | 4      | 1 | 1 | 1 | 1 | 2 |  | 1 | 2 | 3 | 1 | 2 | 1 | 45                | 2      | 1 | 1 | 1 | 3 | 2 |  | 1 | 1 | 1 | 1 | 2 | 4 |
| 20                | 3      | 2 | 1 | 1 | 1 | 2 |  | 1 | 2 | 3 | 1 | 1 | 2 | 46                | 2      | 1 | 1 | 1 | 4 | 1 |  | 1 | 1 | 1 | 2 | 1 | 4 |
| 21                | 3      | 1 | 2 | 1 | 1 | 2 |  | 1 | 2 | 2 | 2 | 1 | 2 | 47                | 2      | 1 | 1 | 2 | 3 | 1 |  | 1 | 1 | 2 | 1 | 1 | 4 |
| 22                | 3      | 1 | 1 | 2 | 1 | 2 |  | 1 | 2 | 2 | 2 | 2 | 1 | 48                | 2      | 1 | 1 | 2 | 2 | 2 |  | 1 | 2 | 1 | 1 | 1 | 4 |
| 23                | 3      | 1 | 1 | 2 | 2 | 1 |  | 1 | 2 | 1 | 3 | 2 | 1 | 49                | 2      | 1 | 1 | 3 | 1 | 2 |  | 1 | 2 | 1 | 1 | 2 | 3 |
| 24                | 3      | 1 | 1 | 1 | 3 | 1 |  | 1 | 2 | 1 | 4 | 1 | 1 | 50                | 2      | 1 | 1 | 3 | 2 | 1 |  | 1 | 2 | 1 | 1 | 3 | 2 |
| 25                | 3      | 1 | 1 | 1 | 2 | 2 |  | 1 | 1 | 2 | 4 | 1 | 1 | 51                | 2      | 1 | 1 | 4 | 1 | 1 |  | 1 | 1 | 2 | 1 | 3 | 2 |
| 26                | 3      | 1 | 1 | 1 | 1 | 3 |  | 1 | 1 | 3 | 3 | 1 | 1 | 52                | 2      | 1 | 2 | 3 | 1 | 1 |  | 1 | 1 | 2 | 1 | 4 | 1 |

The specific arrangement of the Row Address Patterns minimizes the signal interference from adjacent rows, and thus maximizes the readability of the Row Address Patterns. The Reference Decode algorithm (see Annex J of this document) describes how any residual signal interference is detected and corrected, using the cluster number of the PDF417 codeword adjacent to the decoded Row Address Pattern.

Table 2 shows that the sequence of the 52 vertically adjacent row patterns is cyclical in nature and the minimum-difference property therefore also applies when comparing pattern 52 with pattern 1. Thus, the minimum-difference property also holds true if the patterns are “rotated” any arbitrary amount. After RAP rotation, it remains true that any two vertically adjacent patterns differ by only one module.

In order to facilitate discrimination between the versions of MicroPDF417, groups of versions utilize different predefined “RAP rotations” of the set of 52 patterns for the center (if present) and right Row Address Pattern columns relative to the nearest neighboring RAP column to the left. These “RAP rotations,” and the mapping of these pattern combinations to row numbers, are described in 5.11, Low Level Encodation. In addition to providing row location information, MicroPDF417 Row Address Patterns also provide for the detection of scan direction across the symbol (i.e., whether a scan is a forward scan, in which case the Left Row Address Pattern will have been scanned first, or a reverse scan, in which case the stop bar and Right Row Address Pattern will have been scanned first). Table 2 shows that all of the Left and Right Row Address Patterns begin with a two-, three- or four-module bar. Since there is always a one-module stop bar following the MicroPDF417 Right Row Address Pattern, it is possible to detect a reverse scan of these patterns.

5.3 Basic encodation

5.3.1 Symbol character structure

MicroPDF417 utilizes the PDF417 symbol characters, with the exception of the PDF417 Start and Stop Characters.

Each symbol character shall consist of four bar elements and four space elements, each of which can be one to six modules wide. The four bar and four space elements shall measure 17 modules in total. MicroPDF417 symbol characters can be decoded by measuring the e-distances within the character.

Each symbol character is defined by an 8-digit sequence which represents the module widths of the eight elements of that symbol character. Figure 3 illustrates a symbol character with the bar-space sequence 51111125.

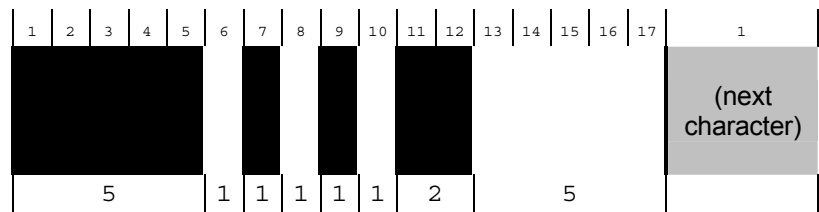


Figure 3 — A PDF417 symbol character

There are 929 defined symbol character values (codewords), numbered from 0 to 928.

The set of codewords is represented in three mutually exclusive symbol character sets, or clusters. Each cluster encodes the 929 available codewords into different bar-space patterns so that one cluster is distinct from another. The cluster numbers are 0, 3, 6. The cluster definition applies to all symbol characters, but not to the MicroPDF417 Row Address Patterns.

The cluster number *K* is defined by the following formula:

$$K = (b_1 - b_2 + b_3 - b_4 + 9) \bmod 9$$

where *b*<sub>1</sub>, *b*<sub>2</sub>, *b*<sub>3</sub> and *b*<sub>4</sub> represent the width in modules of the four bar elements respectively

The cluster number *K* for the symbol character in Figure 3 is:

$$K = (5 - 1 + 1 - 2 + 9) \bmod 9 = 3$$

The codewords and the bar-space sequences for each cluster of symbol characters are given in Annex A.

### 5.3.2 Start and Stop Patterns

The start character function is performed by the Left Row Address Pattern, and the stop character function by the Right Row Address Pattern together with the stop bar.

## 5.4 High level (data) encodation

High level encoding converts the data characters into their corresponding codewords.

Data compaction schemes shall be used to achieve high level encoding. Three modes are defined below, each of which defines a particular efficient mapping between user defined data and codeword sequences. MicroPDF417 utilises the three data compaction modes of PDF417:

- Text Compaction mode (see 5.4.2).
- Byte Compaction mode (see 5.4.3).
- Numeric Compaction mode (see 5.4.4).

A given string of data bytes may be represented by different codeword sequences, depending on how the encoder switches between compaction modes and sub-modes. There is no single specified way to encode data in a MicroPDF417 symbol.

900 codewords are available in each mode for data encodation and other functions within the mode. The remaining 29 codewords are assigned to specific functions (see 5.4.1) independent of the current compaction mode. If data codewords appear before any ECI is invoked in the symbol then the first codeword must be a function codeword. This may require the use of a codeword 901 (latch to Byte Compaction mode) even though this is the default compaction mode (see 5.4.3).

MicroPDF417 also supports the Extended Channel Interpretation system, which allows different interpretations of data to be accurately encoded in the symbol (see 5.5).

### 5.4.1 Function codewords

Codewords 900 to 928 are assigned as function codewords as follows:

- for switching between modes (see 5.4.1.1)
- for enhanced applications using Extended Channel Interpretations (ECIs) (see 5.4.1.2)
- for other enhanced applications (see 5.4.1.3 through 5.4.1.7)

At present codeword 919 is reserved. Table 3 defines the complete list of assigned and reserved function codewords. Their functions are defined in 5.4.1.1 to 5.4.1.7. See 5.4.6 for the treatment of reserved codewords.

**Table 3 — Assignments of function codewords**

| <b>Codeword</b> | <b>Function</b>                                                                                                                           | <b>See clause</b> |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| 900             | Mode Latch to Text Compaction mode                                                                                                        | 5.4.1.1           |
| 901             | Mode Latch to Byte Compaction mode                                                                                                        | 5.4.1.1           |
| 902             | Mode Latch to Numeric Compaction mode                                                                                                     | 5.4.1.1           |
| 903             | UCC/EAN-128 emulation: transmit ]C1 or ]L3; implied Mode Latch 900                                                                        | 5.4.1.5           |
| 904             | UCC/EAN-128 emulation: transmit ]C1 or ]L3; implied Mode Latch 902                                                                        | 5.4.1.5           |
| 905             | UCC/EAN-128 emulation: transmit ]C1 or ]L3; implied Mode Latch 902, followed by implied "01" AI and 14-digit expansion of first 13 digits | 5.4.1.5           |
| 906             | "Linked" UCC/EAN-128: transmit ]C1 or ]L3; implied Mode Latch 900                                                                         | 5.4.1.5           |
| 907             | "Linked" UCC/EAN-128: transmit ]C1 or ]L3; implied Mode Latch 902                                                                         | 5.4.1.5           |
| 908             | Code 128 emulation: transmit ]C2 or ]L4; implied Mode Latch 900                                                                           | 5.4.1.5           |
| 909             | Code 128 emulation: transmit ]C2 or ]L4; implied Mode Latch 902                                                                           | 5.4.1.5           |
| 910             | Code 128 emulation: transmit ]C0 or ]L5; implied Mode Latch 900                                                                           | 5.4.1.5           |
| 911             | Code 128 emulation: transmit ]C0 or ]L5; implied Mode Latch 902                                                                           | 5.4.1.5           |
| 912             | "Linked" UCC/EAN-128, with leading date field: transmit ]C1 or ]L3; implied Mode Latch 902; specially encoded date field.                 | 5.4.1.5           |
| 913             | mode shift to Byte Compaction mode                                                                                                        | 5.4.1.1           |
| 914             | "Linked" UCC/EAN-128, with lot number: transmit ]C1 or ]L3; implied Mode Latch 902, followed by implied "10" AI.                          | 5.4.1.5           |
| 915             | "Linked" UCC/EAN-128, with serial number: transmit ]C1 or ]L3; implied Mode Latch 902; followed by implied "21" AI.                       | 5.4.1.5           |
| 916             | 05 Macro strings added to transmitted data; implied mode latch 902                                                                        | 5.4.1.7           |
| 917             | 06 Macro strings added to transmitted data; implied mode latch 900                                                                        | 5.4.1.7           |
| 918             | linkage flag to associated linear component, in a composite symbol (other than an EAN.UCC Composite symbol)                               | 5.4.1.6           |
| 919             | reserved                                                                                                                                  |                   |
| 920             | linkage flag to associated linear component, in an EAN.UCC Composite symbol                                                               | 5.4.1.6           |
| 921             | reader initialisation                                                                                                                     | 5.4.1.4           |
| 922             | terminator codeword for Structured Append control block                                                                                   | 5.13              |
| 923             | sequence tag to identify the beginning of optional fields in the Structured Append control block                                          | 5.13              |
| 924             | mode latch to Byte Compaction mode (used differently from 901)                                                                            | 5.4.1.1           |
| 925 to 927      | identifier for an Extended Channel Interpretation (ECI)                                                                                   | 5.5               |
| 928             | Structured Append marker codeword to indicate the beginning of a Structured Append Control Block                                          | 5.13              |

#### 5.4.1.1 Function codewords for mode switching

In one MicroPDF417 symbol it is possible to switch back and forth between modes as often as required. Advice about selecting the appropriate modes is given in 5.4.5.

A mode latch codeword may be used to switch from the current mode to the indicated destination mode which stays in effect until another mode switch is explicitly brought into use. Codewords 900 to 902 and 924 are assigned to this function. Table 3 defines their function.

The mode shift codeword 913 shall cause a temporary switch from Text Compaction mode to Byte Compaction mode. This switch shall be in effect for only the next codeword, after which the mode shall revert to the prevailing sub-mode of the Text Compaction mode. Codeword 913 is only available in Text Compaction mode; its use is described in 5.4.2.4.

**Table 4 — Mode definition and mode switching codewords**

| Destination mode   | Mode latch | Mode shift |
|--------------------|------------|------------|
| Text Compaction    | 900        |            |
| Byte Compaction    | 901/924    | 913        |
| Numeric Compaction | 902        |            |

NOTE The table identifies the codeword which shall be used to switch to the defined mode.

The switching rules between the three modes are defined in Table 5 and shown schematically in Figure 4.

**Table 5 — Mode transition table, showing codewords and their function**

| Original mode  | Destination mode |                                                    |                |
|----------------|------------------|----------------------------------------------------|----------------|
|                | Text             | Byte                                               | Numeric        |
| <b>Text</b>    | 900 mode latch   | 913 mode shift<br>901 mode latch<br>924 mode latch | 902 mode latch |
| <b>Byte</b>    | 900 mode latch   | 901 mode latch<br>924 mode latch                   | 902 mode latch |
| <b>Numeric</b> | 900 mode latch   | 901 mode latch<br>924 mode latch                   | 902 mode latch |

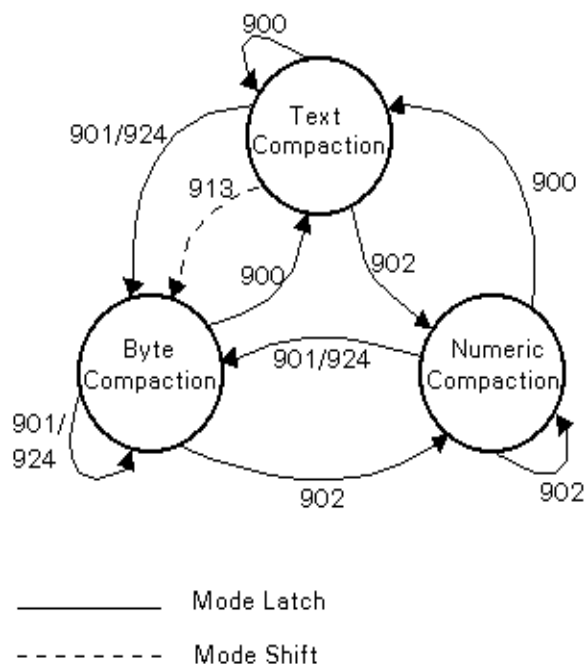


Figure 4 — Available mode switching

The switching rules into Byte Compaction mode are more fully defined in 5.4.3.1.

#### 5.4.1.2 Function codewords for switching to Extended Channel Interpretations

An ECI codeword can be used to switch to a particular interpretation, which stays in effect until another ECI codeword is explicitly brought into use or until the end of the data. Codewords 925 to 927 are assigned to this function (see 5.5).

#### 5.4.1.3 Function codewords for Structured Append

Structured Append symbols (see 5.13) shall use codeword 928 at the start of the Structured Append Control Block. Codewords 922 and 923 are used for special functions in Structured Append.

#### 5.4.1.4 Function codeword for reader initialization

Codeword 921 shall be used to instruct the reader to interpret the data contained within the symbol as programming for reader initialisation. If used, codeword 921 shall appear as the first codeword. In the case of a Structured Append initialisation sequence, Codeword 921 shall appear in every symbol.

The data contained in an initialisation symbol, or sequence of symbols, shall not be transmitted by the reader.

#### 5.4.1.5 Function codewords for Code 128 emulation

Codewords 903 through 915 (excluding 913), if they appear in the first codeword position (that is, in place of the ECI Descriptor codeword), indicate that the MicroPDF417 symbol's data output shall conform with the Code 128 specification as defined in ISO/IEC 15417. Each of these codewords implies a mode latch to either Text Compaction Mode (with an implied **ml** latch to Mixed sub-mode character) (see 5.4.2) or Numeric

Compaction Mode (see 5.4.4); see Table 3 for the specific codeword assignments. If Symbology Identifiers are enabled, then each such symbol's transmission shall be prefixed by either "Jm" or "JLm", where the modifier "m" is as shown in Table 3. MicroPDF417 readers shall be configurable to transmit either symbology identifier prefix, as required by the application.

Whether or not a codeword in the range 903 to 905 appears in the first codeword position, codewords in that range may appear in subsequent codeword positions as a field delimiter, similar to the use of FNC1 in Code 128 symbols. The reader shall transmit any such delimiter codeword as ASCII character 29 (<sup>G</sup><sub>s</sub>). When appearing in other than the first codeword position, codeword 903 implies a latch to the Alpha sub-mode of Text Compaction mode, codeword 904 implies a latch to the Mixed sub-mode of Text Compaction mode, and codeword 905 implies a latch to Numeric Compaction mode. A 905 delimiter codeword, following a series of codewords in Numeric Compaction mode, serves to terminate the current grouping and then restarts Numeric Compaction mode. The appearance of an adjacent pair of repeated 903, 904 or 905 codewords will cause the decoder to divide the transmission at that point (but without transmitting the two implied ASCII 29 (<sup>G</sup><sub>s</sub>) characters themselves); both substrings shall begin with the appropriate Symbology Identifier prefix.

In addition to the features just described, these Code 128 emulation codewords provide additional functionality as follows:

- Codewords 903 through 907, 912, 914 and 915 emulate UCC/EAN-128 symbols. In order to conform fully with that application standard:
  - It is the encoder's responsibility to ensure that strings longer than 48 characters are divided into two or more shorter strings by placing a single or double delimiter codeword (903, 904 or 905) before an AI; and
  - It is the reader's responsibility to break up long transmissions at doubled delimiter (two 903, 904 or 905 codewords) boundaries, or at single delimiter boundaries if necessary, to create multiple transmissions no more than 48 characters in length.
- Codeword 905 provides additional compaction when the message to be encoded begins with an Application Identifier "01" representing an SSC-14 number. In this case, then the leading "01" is not encoded (but will be transmitted by the reader), nor is the fourteenth digit of the SSC-14 number encoded (a mod 10 check digit is calculated and transmitted by the reader).
- Codewords 906 and 907 indicate UCC/EAN-128 emulation, but also indicate that a linear symbol printed below the MicroPDF symbol is "linked" to the data of the MicroPDF symbol. When the reader decodes a MicroPDF417 symbol beginning with 906 or 907, the reader shall not transmit the MicroPDF417 data unless a linear symbol is also found and decoded within the scanner's field of view. If successful, the decoder shall transmit the data from the linear symbol first (using the appropriate symbology identifier for the linear symbol) followed by the MicroPDF417 data in UCC/EAN-128 format as a second transmission. If the linear symbol is a UCC/EAN-128 symbol, then the MicroPDF417 data may be concatenated to the linear symbol's data, and sent as a single transmission (if the 48-character limit is not exceeded).
- Codewords 908 and 909 cause the decoder's symbology identifier prefix to be set to Jc2 or JL4, indicating that the first two digits or single alpha character of the data represents an Application Indicator, assigned by AIM International, followed by an implied FNC1 character. No special action is required on the decoder's part to conform with this rule (beyond setting the prefix to Jc2 or JL4), as the Application Indicator is transmitted exactly as encoded.
- Codewords 910 and 911 cause the decoder's symbology identifier to be set to Jc0 or JL5. No other special action is required on the decoder's part to conform with this rule as the data is transmitted exactly as encoded.
- Codeword 912 is similar to Codeword 907, indicating a "linked" UCC/EAN 128 emulation with an implied start in Numeric Compaction Mode. However, Codeword 912 also indicates that the data begins with a 6-digit date field (AI's 11, 13, 15, or 17), and this date field may be followed by an implied AI 10 or 21 as well. The first 6 digits of encoded data are defined as follows:

- The first 2 digits YY represent a year (range 00..99)
- The next two digits xx represent the month, and the date type, as follows:
  - $(xx \bmod 13)$  represents the month MM (range 01..12).

NOTE xx values of 00, 13, 26, and 39 are unused.

- $(xx \div 13)$  represents D, the type of date, as follows:
  - if D is 0: Implied AI is "17" (Expiration date)
  - if D is 1: Implied AI is "15" (Sell-by date)
  - if D is 2: Implied AI is "13" (Packaging date)
  - if D is 3: Implied AI is "11" (Production date)
- The last two digits zz represent the day, and the following AI, as:
  - $(zz \bmod 32)$  represents the day DD (range 00..31, where 00 means "day not specified")
  - $(zz \div 32)$  represents A, the presence or absence of a next implied AI:
    - if A is 0: no Implied AI follows the date (either the data string ends after the date, or an explicit AI follows the date)
    - if A is 1: an Implied AI of "10" (Lot number) follows the date
    - if A is 2: an Implied AI of "21" (Serial number) follows the date
- Codewords 914 and 915 are similar to Codeword 907, indicating a "linked" UCC/EAN 128 emulation with an implied start in Numeric Compaction Mode. However, Codeword 914 also indicates that the data begins with an implied AI 10 (Lot number); similarly, Codeword 915 indicates that the data begins with an implied AI 21 (Serial number).

Refer to the GS1 specifications for further information about the data content of UCC/EAN 128 symbols.

#### 5.4.1.6 Function codewords for linkage flags in composite symbols

Codeword 920 shall be used as a linkage flag to signal the presence of an associated linear component in accordance with ISO/IEC 24723.

Codeword 918 shall be used as a linkage flag to signal the presence of an associated linear component in any other composite symbology.

When used, the 918 or 920 codeword may appear in any position in the symbol. The applicable composite symbology specification may define a specific position of the linkage flag.

Readers supporting the indicated composite application should decode and transmit the data from all components as specified in the relevant composite symbology specification. Readers not supporting the indicated composite application may treat the 918 or 920 codeword as a reserved codeword (see 5.4.6). In addition, readers not supporting the indicated 918 composite application may have an option to ignore the 2D component and transmit only the data from the associated linear component.

#### 5.4.1.7 Macro Characters

MicroPDF417 provides a means of abbreviating an industry-specific header and trailer in one symbol character. This feature exists to reduce the number of symbol characters needed to encode data in a symbol using certain structured formats.

The two Macro characters apply only when in the first symbol character position. Their functions are defined as follows:

- if the first symbol character is 916 (i.e. encoding Macro character 05), then the preamble  $[>^R_s 05^G_s]$  shall precede the encoded data that follows it. The data begins with an implied latch to Numeric Compaction mode.
- if the first symbol character is 917 (i.e. encoding Macro character 06), then the preamble  $[>^R_s 06^G_s]$  shall precede the encoded data that follows it. The data begins with an implied latch to the mixed sub-mode of Text Compaction mode.
- the postamble  $^R_s E_{OT}$  shall be transmitted after the data in both cases.

The header shall be transmitted as a prefix to the data stream and the trailer shall be transmitted as a suffix to the data stream. Within the encoded data, codewords 903 through 905 may be used as field separators transmitted as ASCII  $^G_s$ . The symbology identifier, if used, shall precede the header.

The Macro characters shall not be used in conjunction with a series of Structured Append symbols.

#### 5.4.2 Text Compaction mode

The Text Compaction mode includes all the printable ASCII characters (i.e. values from 32 to 126) and three ASCII control characters: HT or tab (ASCII value 9), LF or line feed (ASCII value 10), and CR or carriage return (ASCII value 13). The Text Compaction mode also includes various latch and shift characters which are used exclusively within the mode.

The Text Compaction mode encodes up to 2 characters per codeword. The compaction rules for converting data into PDF417 codewords are defined in 5.4.2.2. The sub-mode switches are defined in 5.4.2.3.

##### 5.4.2.1 Text Compaction sub-modes

The Text Compaction mode has four sub-modes:

- Alpha (uppercase alphabetic)
- Lower (lowercase alphabetic)
- Mixed (numeric and some punctuation)
- Punctuation

Each sub-mode contains 30 characters, including sub-mode latch and shift characters.

A latch codeword from another mode to the Text Compaction mode shall always switch to the Text Compaction Alpha sub-mode.

All the characters and their values are defined in Table 6.

Table 6 — Text Compaction sub-mode definition

| Base 30 value | Text Compaction sub-modes |       |       |       |       |       |             |       |
|---------------|---------------------------|-------|-------|-------|-------|-------|-------------|-------|
|               | Alpha                     |       | Lower |       | Mixed |       | Punctuation |       |
|               | Char                      | ASCII | Char  | ASCII | Char  | ASCII | Char        | ASCII |
| 0             | A                         | 65    | a     | 97    | 0     | 48    | ;           | 59    |
| 1             | B                         | 66    | b     | 98    | 1     | 49    | <           | 60    |
| 2             | C                         | 67    | c     | 99    | 2     | 50    | >           | 62    |
| 3             | D                         | 68    | d     | 100   | 3     | 51    | @           | 64    |
| 4             | E                         | 69    | e     | 101   | 4     | 52    | [           | 91    |
| 5             | F                         | 70    | f     | 102   | 5     | 53    | \           | 92    |
| 6             | G                         | 71    | g     | 103   | 6     | 54    | ]           | 93    |
| 7             | H                         | 72    | h     | 104   | 7     | 55    | _           | 95    |
| 8             | I                         | 73    | i     | 105   | 8     | 56    | '           | 96    |
| 9             | J                         | 74    | j     | 106   | 9     | 57    | ~           | 126   |
| 10            | K                         | 75    | k     | 107   | &     | 38    | !           | 33    |
| 11            | L                         | 76    | l     | 108   | CR    | 13    | CR          | 13    |
| 12            | M                         | 77    | m     | 109   | HT    | 9     | HT          | 9     |
| 13            | N                         | 78    | n     | 110   | ,     | 44    | ,           | 44    |
| 14            | O                         | 79    | o     | 111   | :     | 58    | :           | 58    |
| 15            | P                         | 80    | p     | 112   | #     | 35    | LF          | 10    |
| 16            | Q                         | 81    | q     | 113   | -     | 45    | -           | 45    |
| 17            | R                         | 82    | r     | 114   | .     | 46    | .           | 46    |
| 18            | S                         | 83    | s     | 115   | \$    | 36    | \$          | 36    |
| 19            | T                         | 84    | t     | 116   | /     | 47    | /           | 47    |
| 20            | U                         | 85    | u     | 117   | +     | 43    | "           | 34    |
| 21            | V                         | 86    | v     | 118   | %     | 37    |             | 124   |
| 22            | W                         | 87    | w     | 119   | *     | 42    | *           | 42    |
| 23            | X                         | 88    | x     | 120   | =     | 61    | (           | 40    |
| 24            | Y                         | 89    | y     | 121   | ^     | 94    | )           | 41    |
| 25            | Z                         | 90    | z     | 122   | pl    |       | ?           | 63    |
| 26            | space                     | 32    | space | 32    | space | 32    | {           | 123   |
| 27            | ll                        |       | as    |       | ll    |       | }           | 125   |
| 28            | ml                        |       | ml    |       | al    |       | '           | 39    |
| 29            | ps                        |       | ps    |       | ps    |       | al          |       |

al = latch to alpha

ml = latch to mixed

as = shift to alpha

pl = latch to punctuation

ll = latch to lower

ps = shift to punctuation

NOTE The 'Char' columns above show the default interpretation in ECI 000003 of the byte values shown in the adjacent 'ASCII' columns. Each table entry represents half a codeword, i.e. the value range from 0 to 29 (see 5.4.2.2)

### 5.4.2.2 Compaction rules for encoding in Text Compaction mode

In Text Compaction mode, pairs of data characters are represented in a single codeword. The values assigned to the data characters are in the range 0 to 29 (i.e. base 30) and are defined in Table 6. Annex G defines the algorithm and illustrates a worked example.

### 5.4.2.3 Text Compaction sub-mode switching : latch and shift function

Switching from one sub-mode to another within Text Compaction mode shall be through the latch and shift values defined for the sub-mode in effect prior to the switch.

A **sub-mode shift** shall be used to switch from one Text Compaction sub-mode to another for only one data character. Subsequent codewords revert to the sub-mode being used immediately prior to the shift (except when **ps** is used as a pad, see 5.4.2.4). The shift functions are as follows:

ps = shift to punctuation sub-mode

as = shift to uppercase alphabetic sub-mode

A **sub-mode latch** shall be used to switch from one Text Compaction sub-mode to another, which stays in effect until another latch or shift is explicitly brought into use. The latch functions are as follows:

al = latch to uppercase alphabetic sub-mode

ll = latch to lowercase alphabetic sub-mode

ml = latch to mixed (numeric and other punctuation) sub-mode

pl = latch to punctuation sub-mode

A limited set of latch and shift functions is available within each Text Compaction sub-mode. Those which are available are listed in Table 6. Table 7 shows the transition table between Text Compaction sub-modes; Figure 5 shows this schematically.

NOTE: A sub-mode latch may be followed by another sub-mode latch or sub-mode shift; but a sub-mode shift may not be followed by either a sub-mode shift or sub-mode latch.

**Table 7 — Text Compaction sub-mode transition table**

| Original sub-mode | Destination sub-mode |       |       |             |
|-------------------|----------------------|-------|-------|-------------|
|                   | Alpha                | Lower | Mixed | Punctuation |
| Alpha             |                      | ll    | ml    | ps          |
| Lower             | as                   |       | ml    | ps          |
| Mixed             | al                   | ll    |       | ps<br>pl    |
| Punctuation       | al                   |       |       |             |

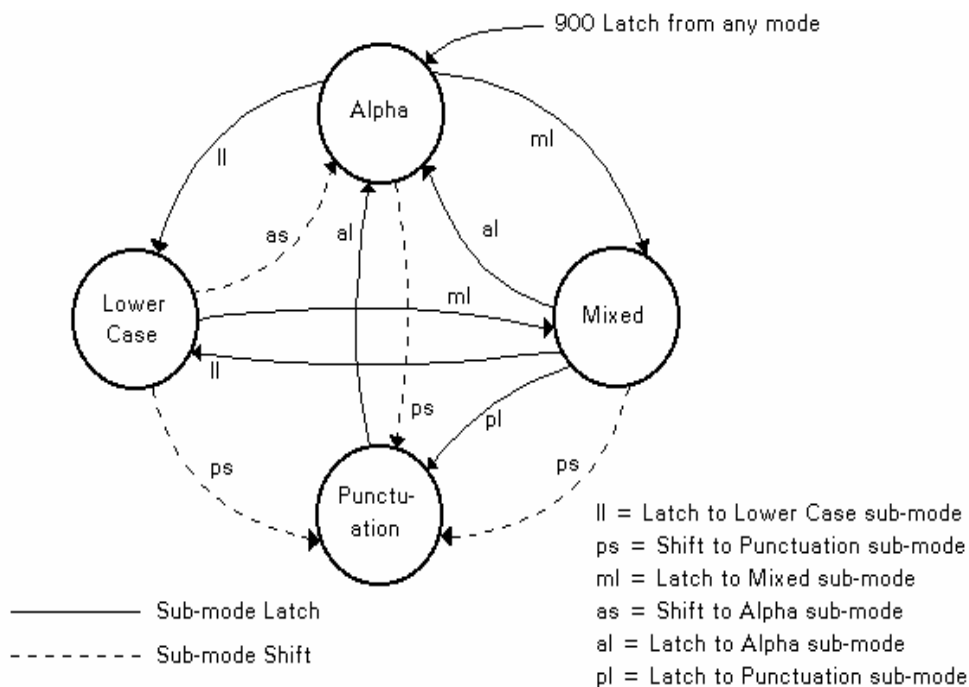


Figure 5 — Text Compaction sub-mode switching

#### 5.4.2.4 Mechanisms for using a pad in Text Compaction mode

If the Text Compaction character sequence does not result in an even number of base 30 values, a pad shall be added to the end of the character sequence. An example is illustrated in Table G.1. As there are no specific null functions in Text Compaction mode, the sub-mode shift and latch shall be used in accordance with the mechanisms defined for the following cases.

The cases are as follows:

- a) If the character sequence continues to the end of the data, or the Text Compaction mode character sequence is followed by latching to another compaction mode, then the pad can be any of the sub-mode shifts or sub-mode latches.
- b) If the Text Compaction mode character sequence is followed by a byte shift (codeword 913) to encode a single Byte Compaction mode character, two mechanisms can be used depending on the Text Compaction sub-mode being used prior to the Byte Compaction shift:
  - 1) If the Text Compaction sub-mode is other than punctuation, then base 30 value 29 (**ps**) should be used if encodation is intended to revert to the same Text Compaction sub-mode. The decoder shall ignore a **ps** immediately preceding codeword 913. If encodation is intended to change to a different sub-mode after the Byte Compaction mode character, the appropriate sub-mode latch may be used as the last base 30 value before the codeword 913.
  - 2) If the Text Compaction sub-mode is punctuation, then base 30 value 29 (**al**) shall be used. The decoder shall not ignore the **al** and shall therefore return to the alpha sub-mode.

#### 5.4.2.5 Switching from Text Compaction mode

Text Compaction mode may be terminated by the end of the symbol, or by any of the following codewords:

- 900 (Text Compaction mode latch)
- 901 (Byte Compaction mode latch)

- 902 (Numeric Compaction mode latch)
- 924 (Byte Compaction mode latch)
- 928 (Beginning of Structured Append Control Block)
- 923 (Beginning of Structured Append Optional Field)
- 922 (Structured Append Terminator)

The last three codewords only occur within the Structured Append Control Block of a Structured Append symbol (see 5.13.1). Text Compaction mode is also affected by the presence of a reserved codeword (see 5.4.6).

If the decoder is in the Text Compaction mode and encounters codeword 913 (Byte Compaction mode shift), it decodes the codeword following codeword 913 as a single binary byte and then returns to the Text Compaction mode. The sub-mode to which the decoder returns is the most-recently-latched sub-mode that was in effect prior to codeword 913; a **ps** sub-mode shift immediately prior to codeword 913 is ignored.

If the decoder is in the Text Compaction mode and encounters codeword 900 (Text Compaction mode latch), the decoder reinitialises to the Alpha sub-mode.

### 5.4.3 Byte Compaction mode

The Byte Compaction mode enables a sequence of 8-bit bytes to be encoded into a sequence of codewords. It is accomplished by a Base 256 to Base 900 conversion, which achieves a compaction ratio of six bytes to five codewords (1,2 : 1).

All the characters and their values (0 to 255) are defined in Annex B. This shall be treated as the default graphical and control character interpretation. When ECIs are invoked (see 5.5) this interpretation is defined as ECI 000003 (see 5.5.2).

**NOTE** In AIM ITS/98-001, superseded by this International Standard, the default character set corresponded to ECI 000002 (a code page of the MS-DOS operating system). The interpretation of byte character values below 128 is unchanged, and the operation of MicroPDF417 printing and scanning equipment is unaffected. New applications that use byte character values above 127 should assume the ECI 000003 default interpretation for broadest compatibility with current systems. Existing applications utilizing values above 127 may continue to encode and process data as before. Applications that rely upon the prior default interpretation of values above 127, may encode ECI 000002 explicitly if they wish to signal this interpretation.

The default compaction mode for MicroPDF417 in effect at the start of each symbol shall always be Byte Compaction mode, introduced by an implied mode latch codeword 901 (see 5.4.3.3) or an explicit codeword 901 or 924 if data codewords appear before any ECI is invoked in the symbol.

If Text or Numeric Compaction mode would result in a smaller symbol, then explicit latches to these modes may be used. In such situations the appropriate Mode Latch may be encoded in the first or subsequent codeword position (if the ECI Descriptor codeword is not used), or in the second or subsequent codeword position (if the ECI Descriptor codeword is used).

#### 5.4.3.1 Switching to Byte Compaction mode

When in either Text or Numeric Compaction mode, to switch to Byte Compaction mode, it is necessary to use one of the following codewords:

- **mode latch 924** shall be used when the total number of bytes to be encoded is an integer multiple of 6
- **mode latch 901** shall be used when the total number of bytes to be encoded is not a multiple of 6
- **mode shift 913** can be used instead of codeword 901 when in Text Compaction mode and a single Byte Compaction character has to be encoded

#### 5.4.3.2 Compaction rules for encoding a single Byte Compaction character (using mode shift 913)

To encode a single Byte Compaction character, the codeword shall be the decimal value (0 to 255) of the character as defined in Annex B.

#### 5.4.3.3 Compaction rules for encoding longer Byte Compaction character strings (using mode latch 924 or 901)

The following procedure shall be used to encode Byte Compaction character data:

- a) Establish the total number of bytes.
- b) If not a perfect multiple of 6, mode latch 901 shall be used (either implicitly at the beginning of the symbol if in the default mode and an ECI is invoked before any data codewords, or explicitly); if a perfect multiple of 6, mode latch 924 shall be used.
- c) Sub-divide the number of bytes into a sequence of 6 characters, from left to right (the most to least significant characters). If less than 6 characters go to Step 7.
- d) Assign the decimal values of the 6 data bytes to be encoded in Byte Compaction mode as  $b_5$  to  $b_0$  (where  $b_5$  is the first data byte).
- e) Carry out a base 256 to base 900 conversion to produce a sequence of 5 codewords. Annex C defines an algorithm and illustrates a worked example.
- f) Repeat from Step 3 as necessary.
- g) For the remaining bytes when mode latch 901 is used, (i.e. when the last group is less than 6 bytes) the codeword(s) shall be the decimal value(s) (0 to 255) of the character(s) as defined in Annex B, the most to the least significant.

NOTE: Byte Compaction mode following mode latch 901 assumes that the total number of bytes to be encoded is not a multiple of six. If the number of bytes to be encoded in Byte Compaction mode happens to be an integer multiple of six, then either a 901 or a 924 Byte Compaction Latch shall be encoded, placed at any point in the symbol that would create a correct encodation according to these encodation rules. For example, a 924 codeword as either the first or second codeword would identify the following stream of Byte Compaction mode codewords as encoding a multiple-of-six number of bytes. Alternatively, a 901 could be placed at any position within the Byte Compaction mode codeword stream that would split that stream into two segments, neither of which encodes a multiple-of-six number of bytes.

If additional encodation is required in Text Compaction or Numeric Compaction modes, the appropriate latch characters shall be used (see 5.4.1.1).

#### 5.4.3.4 Switching from Byte Compaction mode

Byte Compaction mode may be terminated by the end of the symbol, or by any of the following codewords:

- 900 (Text Compaction mode latch)
- 901 (Byte Compaction mode latch)
- 902 (Numeric Compaction mode latch)
- 924 (Byte Compaction mode latch)
- 928 (Beginning of Structured Append Control Block)
- 923 (Beginning of Structured Append Optional Field)
- 922 (Structured Append Terminator)

The last three codewords only occur within the Structured Append Control Block of a Structured Append symbol (see 5.13.1). Byte Compaction mode is also affected by the presence of a reserved codeword (see 5.4.6).

Re-invoking Byte Compaction mode (by using codeword 901 or 924 while in Byte Compaction mode) serves to terminate the previous Byte Compaction mode grouping of 6 Byte Compaction characters as described in 5.4.3.3, and then to start a new grouping. This procedure may be necessary when an ECI assignment number needs to be encoded (see 5.5.3.2).

During the decode process for Byte Compaction mode, the treatment of the final group of codewords differs depending on whether Byte Compaction mode is invoked with codeword 901 or 924.

If Byte Compaction mode is invoked with codeword 924, the total number of codewords within the compaction mode shall be a multiple of five. If this is not the case, the symbol is invalid. All the 5-codeword groups are decoded into 6-byte groups.

If Byte Compaction mode is invoked with codeword 901, the final group of codewords is interpreted directly as one byte per codeword, without compaction. Therefore, if the last group consists of five codewords, the group is interpreted as 5 bytes, rather than 6.

#### 5.4.4 Numeric Compaction mode

The Numeric Compaction mode is a method for base 10 to base 900 data compaction and should be used to encode long strings of consecutive numeric digits. The Numeric Compaction mode encodes up to 2,93 numeric digits per codeword.

##### 5.4.4.1 Latch to Numeric Compaction mode

Numeric Compaction mode may be invoked when in Text Compaction or Byte Compaction modes using mode latch 902.

##### 5.4.4.2 Compaction rules for encoding long strings of consecutive numeric digits

The following procedure shall be used to compact numeric data:

1. Divide the string of digits into groups of 44 digits, except for the last group, which may contain fewer.
2. For each group add the digit 1 to the most significant position to prevent the loss of leading zeros.

EXAMPLE

|               |                  |
|---------------|------------------|
| original data | 00246812345678   |
| after step 2  | 1 00246812345678 |

NOTE The leading digit 1 is removed in the decode algorithm.

3. Perform a base 10 to base 900 conversion. Annex D defines an algorithm for this and illustrates a worked example.
4. Repeat from Step 2 as necessary.

The following rules can be used to determine the precise number of codewords in Numeric Compaction mode:

- Groups of 44 numeric digits compact to 15 codewords.
- For groups of shorter sequences of digits, the number of codewords can be calculated as follows:

$$\text{Codewords} = \text{INT}(\text{number of digits} / 3) + 1$$

EXAMPLE:

For a 28 digit sequence

$$\text{INT} (28 / 3) + 1$$

$$= 9 + 1$$

$$= 10 \text{ codewords}$$

#### 5.4.4.3 Switching from Numeric Compaction mode

Numeric Compaction mode may be terminated by the end of the symbol, or by any of the following codewords:

- 900 (Text Compaction mode latch)
- 901 (Byte Compaction mode latch)
- 902 (Numeric Compaction mode latch)
- 924 (Byte Compaction mode latch)
- 928 (Beginning of Structured Append Control Block)
- 923 (Beginning of Structured Append Optional Field)
- 922 (Structured Append Terminator)

The last three codewords only occur within the Structured Append Control Block of a Structured Append symbol (see 5.13.1). Numeric Compaction mode is also affected by the presence of a reserved codeword (see 5.4.6).

Re-invoking Numeric Compaction mode (by using codeword 902 while in Numeric Compaction mode) serves to terminate the current Numeric Compaction mode grouping as described in 5.4.4.2, and then to start a new grouping. This procedure may be necessary when an ECI assignment number needs to be encoded (see 5.5.3.4).

During the decode process for Numeric Compaction mode, the result of the base 900 to base 10 conversion shall result in a number whose most significant digit is a '1'. If the base 900 to base 10 conversion does not result in a number beginning with '1', the symbol shall be treated as invalid. The leading '1' is removed to produce the original number.

#### 5.4.5 Advice to select the appropriate compaction mode

All basic implementations for printing and scanning MicroPDF417 symbols shall support the three modes: Text Compaction, Byte Compaction and Numeric Compaction. The default character set for Text Compaction shall be as defined in Table 6; and that for Byte Compaction shall be as defined in Annex B. Text Compaction mode is usually more efficient than Byte Compaction mode for encoding standard ASCII text files because of its better compaction of ASCII character values 9, 10, 13 and 32 to 126.

The Numeric Compaction mode should be used for long numeric strings.

Advice about switching between modes to minimise the number of codewords is provided as an algorithm in Annex N.

## 5.4.6 Treatment of MicroPDF417 reserved codewords

### 5.4.6.1 Overview

MicroPDF417 symbols intended for use in open systems should not employ any of the codewords that are listed as reserved (see 5.4.1) in the current edition of this standard. However, decoding equipment should support the transmission of reserved codewords using escape sequences as defined in 5.17.4. Decoding equipment may also support an option of treating such symbols as invalid, as would be the case when operating in Basic Channel Mode.

Receiving systems should discard data containing any escape sequences using reserved codewords, unless the system is aware of a new definition for a previously reserved codeword.

### 5.4.6.2 Making future use of reserved codewords

Currently, function codeword 919 is reserved. Any new function codewords, to be defined in future revisions of this standard, shall have their encoding rules specified to provide backwards compatibility with pre-existing equipment. If a new function codeword introduces a new compaction mode, then the interpretation of the subsequent codewords has a new meaning unknown to pre-existing equipment, and thus the subsequent codewords all must be transmitted as escape sequences, until a known compaction mode codeword is reached. Alternatively, if a new function codeword defines a “signalling” function (such as to alter the transmitted Symbolology Identifier option value) with no effect on the interpretation of subsequent codewords, then only the new codeword itself would need be transmitted as an escape sequence. Specifically:

- When a new signalling codeword is encoded, it shall immediately be followed by an appropriate compaction mode latch so that the subsequent data codewords are interpreted and transmitted as a byte stream, rather than as a series of escaped uninterpreted codewords. This approach will achieve the desired results with decoding equipment conforming with any revision of this MicroPDF417 standard, regardless of whether that equipment employs the original or the new transmission protocol.
- At the receiving system, the ECI decoder will process the control ECIs (i.e. Structured Append Control Blocks and escaped uninterpreted codewords) before the encodable ECIs (such as encryption schemes and character sets). Thus, the encoder should take into account the order of operations as follows:
  1. The Structured Append Control Block ECIs, if present, will be used to assemble the complete byte stream in the proper order.
  2. The escaped data codewords will be translated by the ECI decoder according to the rules of the new compaction mode or signal codeword, and the resulting data bytes will be inserted into their proper place within the byte stream.
  3. Finally, the character set and other encodable ECIs will be applied to the resulting byte stream.

## 5.5 Extended Channel Interpretation

The Extended Channel Interpretation (ECI) protocol allows the output data stream to have interpretations different from that of the default character set. The ECI protocol is defined consistently across a number of symbologies, including MicroPDF417. ECIs are assigned by AIM Global, Inc.

**NOTE** Originally a symbology specific scheme called Global Label Identifiers (GLIs) was defined for PDF417. Encoding and decoding of ECIs is identical to early specifications for PDF417 GLIs. However, the transmission protocol for decoded messages according to early PDF417 specifications for GLIs is different from the transmission protocol for ECIs. There are also differences with respect to the use of interpretative ECIs with Structured Append. This standard permits the use of the earlier and current protocols in such a way that old and new equipment can continue to co-exist.

Five broad types of interpretations are supported in MicroPDF417:

- a) character sets (or code pages)

- b) general-purpose interpretations such as data encryption and data compression (as distinct from the compaction modes of the symbology)
- c) user defined interpretations for closed systems
- d) transmission of control information for Structured Append
- e) transmission of uninterpreted PDF417 codewords

Transmission of the Extended Channel Interpretation protocol is fully specified in AIM ITS/04-001, Part 1. The protocol provides a consistent method to specify particular interpretations of byte values before printing and after decoding.

The Extended Channel Interpretation (ECI) is identified by a 6-digit number which is encoded in the MicroPDF417 symbol by one of three specific codewords followed by one or two codewords (see 5.5.1). A specific ECI may be invoked anywhere in the encoded message subject to the rules of the compaction modes (see 5.5.3).

The ECI protocol can only be used with decoders enabled to transmit the symbology identifier (see 5.17.5). Decoders that are not enabled to transmit the symbology identifier cannot reliably convey the escape sequences from any symbol containing an ECI.

#### 5.5.1 Encoding the ECI assignment number

An ECI can be invoked anywhere in the data stream, subject to the conditions defined in 5.5.3. Once an ECI has been invoked, switching may take place between any of the compaction modes. The compaction mode used is determined strictly by the 8-bit data values being encoded and does not depend on the ECI in force. For example, a sequence of values in the range 48 to 57 (decimal) would be most efficiently encoded in Numeric Compaction mode even if the sequence is not to be interpreted as numbers.

The ECI assignment number is encoded in one of three ECI codeword sequences, which begin with the codewords 927, 926 or 925. One or two additional codewords are used to encode the ECI assignment number. The encodation rules are defined in Table 8. In addition, MicroPDF417 supports the application of the rule in 5.2.4.2 (ECI Descriptor codeword).

**Table 8 — Encoding ECI assignment numbers**

| ECI assignment number | Codeword sequence | Codewords          | Ranges                                                   |
|-----------------------|-------------------|--------------------|----------------------------------------------------------|
| 000000 to 000899      | C <sub>0</sub>    | 927                | C <sub>1</sub> =(0 to 899)                               |
|                       | C <sub>1</sub>    | ECI_no             |                                                          |
| 000900 to 810899      | C <sub>0</sub>    | 926                | C <sub>1</sub> =(0 to 899)<br>C <sub>2</sub> =(0 to 899) |
|                       | C <sub>1</sub>    | ECI_no div 900 - 1 |                                                          |
|                       | C <sub>2</sub>    | ECI_no mod 900     |                                                          |
| 810900 to 811799      | C <sub>0</sub>    | 925                | C <sub>1</sub> =(0 to 899)                               |
|                       | C <sub>1</sub>    | ECI_no - 810900    |                                                          |

There are 811 800 possible ECI assignment numbers available in MicroPDF417.

**NOTE** The encodation method is identical to the GLI scheme supported in the PDF417 sponsor's original specification and incorporated in the AIM USA (1994) and AIM Europe (1994) specifications.