INTERNATIONAL STANDARD

ISO/IEC 24723

First edition
2006-04-15

Information technology — Automatic identification and data capture techniques — EAN.UCC Composite bar code symbology specification

Technologies de l'information — Techniques d'identification automatique et de capture des données — Spécifications de la symbologie des codes à barres du Composant EAN.UCC

Reference number
ISO/IEC 24723:2006(E)

Contents

Page

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 24723 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

Introduction

Composite symbologies are a class of bar code symbology, the principal distinguishing feature of which is that they comprise two, or more, components, each of which is a distinct symbol, but which contain a set of related data. Typically one component is a linear symbol containing primary data, which may be read on its own in some areas of the application, and the other component(s) is a two-dimensional symbol containing supplementary data which qualifies the primary message, and requiring all components to be read to extract the complete message. The EAN.UCC Composite symbology is one such symbology. The use of the symbology is intended to comply with the GS1 General Specifications.

An EAN.UCC Composite symbol consists of a linear component (encoding the item's primary identification) associated with an adjacent 2D component (encoding supplementary data, such as a batch number or expiration date). The EAN.UCC Composite symbol always includes a linear component so that the primary identification is readable by all scanning technologies, and so that 2D imagers can use the linear component as a finder pattern for the adjacent 2D component. The EAN.UCC Composite symbol always includes a multi-row 2D component, for compatibility with linear and 2D imagers, and with linear and rastering laser scanners.

EAN.UCC Composite symbols are intended for encoding identification numbers and data supplementary to the identification in accordance with GS1 General Specifications. The administration of the numbering system by EAN and UCC ensures that identification codes assigned to particular items are unique world-wide and that they and the associated supplementary data are defined in a consistent way.

Information technology — Automatic identification and data capture techniques — EAN.UCC Composite bar code symbology specification

1 Scope

This International Standard defines the requirements for the EAN.UCC Composite symbology. It specifies the EAN.UCC Composite symbology characteristics, data character encodation, symbol formats, dimensions and print quality requirements, error correction rules, and reference decoding algorithms. For those linear and 2D components of EAN.UCC Composite symbols with published symbology specifications, those published specifications apply, except as specifically noted in this International Standard.

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, *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 15416, *Information technology — Automatic identification and data capture techniques — Bar code print quality test specification — Linear symbols*

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

ISO/IEC 15420, *Information technology — Automatic identification and data capture techniques — Bar code symbology specification — EAN/UPC*

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

ISO/IEC 15438, *Information technology — Automatic identification and data capture techniques — PDF417 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)*

ISO/IEC 24728, *Information technology — Automatic identification and data capture techniques — MicroPDF417 bar code symbology specification*

GS1 General Specifications (GS1, Brussels, Belgium)

3 Terms, definitions, abbreviated terms and mathematical operators

3.1 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.

3.1.1
2D component
CC
two-dimensional portion of an EAN.UCC Composite symbol, which encodes supplemental information about an item, such as its lot number or expiration date

3.1.2
AI element string
character string containing an application identifier followed by its associated data field

3.1.3
CC-A
2D component that is a structural variant of MicroPDF417

NOTE 1 CC-A components can be autodiscriminated from MicroPDF417 symbols, because CC-A components have a unique Rotation of 32 between any two adjacent Row Address Patterns.

NOTE 2 CC-A is one of the three choices for the 2D component in a symbol encoded in the EAN.UCC Composite symbology.

3.1.4
CC-B
2D component that is a MicroPDF417 symbol which begins with PDF417 codeword 920, indicating conformance with this International Standard

NOTE CC-B is one of the three choices for the 2D component in a symbol encoded in the EAN.UCC Composite symbology.

3.1.5
CC-C
2D component that is a PDF417 symbol which begins with PDF417 codeword 920, indicating conformance with this International Standard

NOTE CC-C is one of the three choices for the 2D component in a symbol encoded in the EAN.UCC Composite symbology.

3.1.6
linear component
linear portion of an EAN.UCC Composite symbol, which encodes the primary identification of an item

3.1.7
linkage flag
indicator encoded in an RSS or UCC/EAN-128 component to signal if a 2D component accompanies the linear component

3.1.8
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 MicroPDF417 symbol, or a CC-A or CC-B component

3.1.9
Row Address Pattern
RAP
one of a set of patterns made up of three bars and three spaces occupying ten modules, that serve both as start or stop patterns and as row indicators in a MicroPDF417 symbol, or a CC-A or CC-B component

3.1.10
separator pattern
pattern between the linear and 2D components of an EAN.UCC Composite symbol

3.1.11
symbol separator character
non-data character that is used to break the transmitted data string into separate transmissions, each beginning with the appropriate symbology identifier prefix

3.2 Abbreviated terms

AI Application Identifier (see ISO/IEC 15418)

RSS Reduced Space Symbology

3.3 Mathematical operators and notational conventions

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

div integer division operator which discards the remainder

mod integer remainder after integer division

The following ISO notational conventions are used.

0,2 a comma between digits separates the integer from the decimal fraction (e.g. 0,2 equals 2/10) except when used as an (n,k) designation

12 345 a space between digits indicates factors of a thousand

4 Symbol description

4.1 Basic characteristics

An EAN.UCC Composite symbol consists of a linear component associated with an adjacent 2D component. The characteristics of the EAN.UCC Composite symbology are:

a) Encodable character set:

   1) Both linear and 2D components encode a subset of ISO/IEC 646, consisting of the upper and lowercase letters, digits, and 21 selected punctuation characters.

   2) The function character FNC1 and a Symbol Separator character.

b) Symbol character structure: various edge-to-similar-edge decodable symbol characters are used, in accordance with the selected Linear and 2D components of the symbol.

c) Code type:

   1) Linear component: continuous, linear bar code symbology.

   2) 2D component: continuous, multi-row bar code symbology.

d) Maximum numeric data capacity (including implied application identifiers and calculated check digits where appropriate, but not including any encoded FNC1 characters):

   1) Linear component:

      i) UCC/EAN-128: up to 48 digits

      ii) EAN/UPC: UPC-A, EAN-8, or EAN-13 (12, 8, or 13 digits respectively)

      iii) RSS Expanded: up to 74 digits, see note

      NOTE The RSS Expanded data capacity depends on the encodation method. The maximum is 74 digits for (01) + other AI's, the maximum is 70 digits for any AI's, and the maximum is 77 digits for (01) + (392x) + any AI's.

      iv) Other RSS: 16 digits

   2) 2D component:

      i) CC-A: up to 56 digits

      ii) CC-B: up to 338 digits

      iii) CC-C: up to 2 361 digits

e) Error detection and correction:

   1) Linear component: one check character for error detection.

   2) 2D component: a fixed or variable number of Reed-Solomon error correction codewords, depending upon the specific 2D component.

f) Character self-checking: yes.

g) Bi-directionally decodable: yes.

4.2 Summary of additional features

The following is a summary of additional EAN.UCC Composite symbology features:

a) **Data compaction**: the 2D components utilize a bit-oriented compaction mode, designed to encode efficiently data using application identifiers.

b) **Component linkage**: the 2D component of each EAN.UCC Composite symbol contains a linkage flag, which indicates to the reader that no data shall be transmitted unless the associated linear component is also scanned and decoded. All linear components except EAN/UPC also contain an explicit linkage flag.

c) **UCC/EAN-128 emulation**: readers set to the UCC/EAN-128 emulation mode transmit the data encoded within the EAN.UCC Composite symbol as if the data were encoded in one or more UCC/EAN-128 symbols.

d) **2D component escape mechanism**: a mechanism to support future applications which require data content beyond the ISO/IEC 646 subset encodable in the standard form of the EAN.UCC Composite symbology.

4.3 Symbol structure

Each EAN.UCC Composite symbol consists of a linear component and a multi-row 2D component. The 2D component should nominally be printed with the same X dimension as the linear component. The 2D component is printed above the linear component (as defined in 12.2 and 12.3).

To facilitate printing the two components independently, 12.2 and 12.3 permit relative positional variation between the two components. A bar code reader should not use the relative locations of the components shown in the illustrations of EAN.UCC Composite symbols in this specification to predict the exact location of the 2D component from the location of the linear component.

The linear component is one of:

a) a member of the EAN/UPC symbology (EAN-13, EAN-8, UPC-A, or UPC-E, which may include an add-on symbol, in accordance with ISO/IEC 15420),

b) a member of the Reduced Space symbology family,

c) UCC/EAN-128.

The choice of linear component determines the name of the EAN.UCC Composite symbol, such as an EAN-13 Composite symbol, or a UCC/EAN-128 Composite symbol.

The 2D component (abbreviated as CC) is chosen based on the selected linear component, and on the amount of supplementary data to be encoded. The three 2D components, listed in order of increasing maximum data capacity, are:

a) *CC-A*, a variant of MicroPDF417, designed for efficient encoding of supplemental application identifier data,

b) *CC-B*, a MicroPDF417 symbol with a codeword of 920 in the first data codeword position as a linkage flag, and denoting EAN.UCC data compaction, and

c) *CC-C*, a PDF417 symbol with a codeword of 920 in the first data codeword position as a linkage flag, and denoting EAN.UCC data compaction.

Figure 1 — An RSS Limited Composite symbol

Figure 1 illustrates an RSS Limited Composite symbol which utilizes a 4-row CC-A component as its 2D component. The RSS Limited component in Figure 1 identifies the product as “0113112345678906”, and the CC-A component encodes the expiration date and lot number (as “1701061510A123456”). The human-readable interpretation of the data in the symbols would be shown, if present, as “(01)13112345678906” and “(17)010615(10)A123456” respectively.

Figure 2 — A UCC/EAN-128 Composite symbol

Figure 2 illustrates a UCC/EAN-128 Composite symbol which utilizes a 5-row CC-C component as its 2D component. The UCC/EAN-128 component in Figure 2 identifies the product as “0193812345678901”, and the CC-C component encodes the lot number and deliver to location number (as “10ABCD123456<FNC1>4103898765432108”). The human-readable interpretation of the data in the symbols would be shown, if present, as "(01)93812345678901" and "(10)ABCD123456(410)3898765432108" respectively.

4.4 Supported component combinations

Based upon the width of the linear component, a choice of "best-fit" 2D component is specified. Table 1 lists all of the permissible combinations.

Table 1 — Permissible combinations of linear and 2D components

Linear component	CC-A/CC-B	CC-C
UPC-A and EAN-13	Yes (4-columns)	No
EAN-8	Yes (3-columns)	No
UPC-E	Yes (2-columns)	No
UCC/EAN-128	Yes (4-columns)	Yes (variable width)
RSS-14	Yes (4-columns)	No
RSS-14 Stacked	Yes (2-columns)	No
RSS-14 Stacked Omni-directional	Yes (2-columns)	No
RSS Limited	Yes (3-columns)	No
RSS Expanded	Yes (4-columns)	No
RSS Expanded Stacked	Yes (4-columns)	No

In all cases where a CC-A component is shown in the table, the printing software shall automatically use a CC-B component (of the same number of columns) when the data to be encoded exceeds the maximum capacity of the CC-A or other means shall be provided to enable the correct selection of CC-A or CC-B to suit the length of the data. The presence of an add-on symbol shall not affect the choice of 2D component. When the linear component is UCC/EAN-128, either CC-A/CC-B or CC-C may be used; see 7.3.

5 Source data encodation into a binary string

The user data to be encoded into a 2D component shall always consist of application identifiers and data fields that comply with the data standard of the GS1 General Specifications. The UCC/EAN-128 rules for concatenation of AI element strings, such as the termination of a variable-length string by a FNC1 character, shall be followed when encoding a 2D component.

Before encoding a 2D component, the given data string is encoded into a binary string, where data characters are represented with a variable number of bits. The resulting binary string consists of two or three binary fields. The fields are:

a) Encodation Method (see 5.1),

b) compressed data (see 5.2), and

c) general-purpose data compaction (see 5.3).

The Encodation Method is always encoded as the first field (see 5.1). One or both of the other fields are also encoded in a 2D component. When both are present they are encoded in the order shown above.

The compressed data field efficiently encodes data using specific AIs, such as date and lot number or data using AI 90 and serial numbers (see 5.2).

The general-purpose compaction field can encode data using any combination of AIs (see 5.3).

In the text of this clause, bit fields will be indicated by their binary values enclosed in double quotation marks.

5.1 Encodation Method field

5.1.1 Description of the data string

The Encodation Method field consists of one or two bits. It occupies the first bits in the encoded binary data. It defines whether the symbol is a general-purpose symbol or begins with an application-oriented compressed data field (such as for efficient representation of expiration date and lot number AI element strings). The Encodation Method field is defined in Table 2.

Table 2 — Encodation Method field

Encodation Method field bits	Description
0	AI element string data using general-purpose data compaction
10	Date, lot number, and other AI element strings
11	AI 90 (containing alphanumeric data) optionally followed by other AI element strings

An Encodation Method field of “0” is directly followed by AI data encoded using general-purpose data compaction (see 5.3). The encoded data starts with the first application identifier and its element string data, optionally followed by one or more AI element strings.

An Encodation Method field of “10” or “11” is directly followed by a compressed data field, optionally followed by a general-purpose data compaction field (see 5.3).

5.2 Compressed data field

If the Encodation Method field is other than a single “0” bit, this indicates the presence of a field of compressed data. The compression is specified to encode the data of specific AI element strings efficiently. An Encodation Method field of “10” (see 5.2.1) is directly followed by a compressed data field suited for marking pharmaceuticals and many other small items requiring an Expiration or Production Date and/or Lot number. An Encodation Method field of “11” (see 5.2.2) is directly followed by a compressed data field suited for many applications for marking small products with alphanumeric data. It may be necessary to sort the AI element strings into the required sequence in order to benefit from the applicable Encodation Method.

5.2.1 Encodation Method field “10” — date and lot number

This Encodation Method may be used if the element strings of AI 11 followed by AI 10, AI 17 followed by AI 10, or the single AIs 10, 11 or 17 occur at the start of the data message to be encoded. An Encodation Method field of “10” is followed by a compressed date field (which may be empty), an optional lot number, possibly followed by other AI element strings (which if present, are encoded using general-purpose data compaction).

The compressed date field following the “10” Encodation Method field consists of either:

a) A 17-bit field, consisting of:

   1) a 16-bit value encoding 0 to 38 399 for YYMMDD (year, month, and day) according to the equation

   $$(YY \times 384) + ((MM-1) \times 32) + (DD)$$

   2) followed by a single bit indicating the type of date: “0” for Production Date (AI 11) or “1” for Expiration Date (AI 17)

or:

b) A 2-bit date field of “11” which is a flag indicating that no date was encoded.

NOTE 1 The 16-bit date values 0-38 399 will start with “0” or “10” so that “11” is a unique two-bit flag.

NOTE 2 The YYMMDD format for data is required to comply with EAN.UCC application specifications. Separate algorithms are provided to convert the date into its format of CCYYMMDD.

The date is encoded by stripping the two-digit AI 11 or AI 17 from the date element string and encoding the remaining six digits into 16 bits using the equation above. For AI 11, the next bit is “0”; for AI 17 the next bit is “1”. If there was no date, a 2-bit field of “11” is encoded instead of the 17-bit field.

If a lot number directly follows the date element string, the two-digit AI 10 is stripped from the lot number element string, and the remainder of the lot number element string is encoded using general-purpose data compaction, directly following the date field. If more AI element strings follow the lot number, a FNC1 separates the lot number data from the next AI element string to be encoded.

If a lot number does not directly follow the date element string, a FNC1 is encoded following the date element string, even if no more data follows the date element string (this FNC1 shall not be transmitted by the decoder). If more data follows, it is encoded using general-purpose data compaction beginning with the digits of the next AI.

The decoder shall reconstruct the AI element strings from a compressed data field using an Encodation Method field of “10” according to the following procedure:

a) If the bits “11” follow the method “10”, no date is encoded, and the decoder shall insert the two-digit AI 10 before the remaining general-purpose data compaction field is decoded.

b) Otherwise, the 6-digit date shall be extracted according to the above equation. If the seventeenth bit is “0”, an AI 11 for production date is added as a prefix by the decoder. If the seventeenth bit is “1”, an AI 17 for expiration date is added. If the first encoded data character following the date field is FNC1, there is no lot number (this FNC1 shall not be transmitted by the decoder). Otherwise the decoder shall insert the two-digit AI 10 before the remaining general-purpose data compaction field is decoded.

5.2.2 Encodation Method field of “11”: AI 90

This Encodation Method may be used if an element string with an AI 90 occurs at the start of the data message, and if the data field following the two-digit AI 90 starts with an alphanumeric substring which complies with a specific format. The format of the alphanumeric data that can be used in this compaction method is 0, 1, 2, or 3 digits (strings with leading zeros do not comply with the required format) followed by an uppercase alphabetic character.

An Encodation Method field of “11” is followed by a compressed data field which includes the encoded special-format alphanumeric string, followed by the remainder of its data field, optionally additional compressed data fields, and optionally additional AI element strings.

The compressed data field following the “11” Encodation Method consists of the following:

a) One or two bits indicating the starting data encodation scheme used for the remainder of the AI 90 data field. A “0” indicates Alphanumeric encodation, a “10” indicates Numeric encodation and an “11” indicates Alpha encodation (see 5.2.3).

b) One or two bits indicating the absence or presence, respectively, of specific AIs after the first FNC1 (which terminates the AI 90 element string, unless no further data is encoded). “0” indicates that either no more data is encoded, or that the remaining data is encoded according to general-purpose data compaction rules. Otherwise, the AI of the next element string will not be explicitly encoded. “10” indicates that an AI 21 follows, and “11” indicates that an AI 8004 follows.

c) Nine or 20 bits encoding the 1 to 4 characters of the alphanumeric string that followed the AI 90 in the source message, encoded as follows:

   1) Convert the numeric portion of the alphanumeric string to a value. If the string contains no numeric digits, the value is 0.

2) If the numeric value is less than 31, and if the uppercase alphabetic letter is one of those listed in Table 3, encode the numeric value as a five-bit binary string. Then convert the uppercase alphabetic character to four bits of binary data using Table 3 and encode these as a four-bit binary string.

3) Otherwise encode a five-bit string “11111”, followed by a 10-bit string representing the value of the numeric digits, followed by a five-bit string representing the ASCII value of the letter minus 65 (where “A” is encoded as “00000”, and “Z” is encoded as “11001”).

Table 3 — Supported uppercase alphabetic letters

Letter	Binary Data	Letter	Binary Data
B	0000	P	1000
D	0001	Q	1001
H	0010	R	1010
I	0011	S	1011
J	0100	T	1100
K	0101	V	1101
L	0110	W	1110
N	0111	Z	1111

During encoding, the AI 90 and the alphanumeric string are stripped from the first AI element string. The next two to four bits specify the starting encodation scheme for the remainder of the element string, and whether an AI 21 or AI 8004 element string follows the AI 90 element string. Then the compacted alphanumeric string is encoded in the next nine or twenty bits. The remainder of the AI 90 element string is encoded using either general-purpose data compaction starting in Alphanumeric or Numeric schemes, or the Alpha encodation scheme, as specified by the encodation scheme bit field. The AI element string data is terminated by an encoded FNC1, unless it is the last AI element string of the input data.

If an AI 21 or AI 8004 element string follows the AI 90 element string, the 21 or 8004 is stripped from the element string, and the remainder of the AI element string is encoded, starting in the default Numeric encodation scheme of general-purpose data compaction. If the AI 90 element string is encoded using the numeric encodation scheme when the FNC1 is encountered, the first digit of the following AI element string may be encoded with the FNC1 in the same seven bits of the numeric encodation scheme. Any additional data is then encoded.

The decoder shall reconstruct the AI element strings from a “11” compressed data field according to the following procedure.

a) The 2-digit AI 90 is inserted to start the decoded data string.

b) The first two to four bits are decoded to determine the encodation scheme and whether a specific element string of AI 21 or AI 8004 follows the AI 90 element string.

c) The next five bits are decoded to determine the number of digits in the alphanumeric string and their value.

1) If the value of the five bits is less than 31, these five bits are decoded to determine the number of digits and their value (discarding any leading zeros), and the next four bits are decoded using Table 3 to determine the uppercase alphabetic character in the alphanumeric string.

2) Otherwise, the ten bits following the “11111” are decoded to determine the number of digits and their value (discarding any leading zeros); and the next five bits are decoded to determine the uppercase letter in the alphanumeric string.

d) Append the alphanumeric string to the decoded data string.