BRITISH STANDARD

BS EN ISO/IEC 15438:2010

Information technology — Automatic identification and data capture techniques — PDF417 bar code symbology specification

ICS 35.040

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI Documents
Document Sharing Hub
British Standards

Full document details: BS EN ISO/IEC 15438:2010

National foreword

This British Standard is the UK implementation of EN ISO/IEC 15438:2010. It is identical to ISO/IEC 15438:2006. It supersedes BS ISO/IEC 15438:2006, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/34, Automatic identification data capture techniques.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 June 2006

© BSI 2010

ISBN 978 0 580 70516 8

Amendments/corrigenda issued since publication

Date	Comments
30 September 2010	This corrigendum renumbers BS ISO/IEC 15438:2006 as BS EN ISO/IEC 15438:2010

Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN ISO/IEC 15438

August 2010

ICS 35.040

Supersedes EN ISO/IEC 15438:2003

English Version

Information technology - Automatic identification and data capture techniques - PDF417 bar code symbology specification (ISO/IEC 15438:2006)

Technologies de l'information - Techniques automatiques d'identification et de capture des données - Spécifications pour la symbologie de code à barres PDF417 (ISO/IEC 15438:2006)

Informationstechnik - Verfahren der automatischen Identifikation und Datenerfassung - Spezifikation der Strichcodesymbologie PDF417 (ISO/IEC 15438:2006)

This European Standard was approved by CEN on 29 July 2010.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: Avenue Marnix 17, B-1000 Brussels

© 2010 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No. EN ISO/IEC 15438:2010: E

Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

Foreword

The text of ISO/IEC 15438:2006 has been prepared by Technical Committee ISO/IEC JTC 1 “Information technology” of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) and has been taken over as EN ISO/IEC 15438:2010 by Technical Committee CEN/TC 225 “AIDC technologies” the secretariat of which is held by NEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by February 2011, and conflicting national standards shall be withdrawn at the latest by February 2011.

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

This document supersedes EN ISO/IEC 15438:2003.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

Endorsement notice

The text of ISO/IEC 15438:2006 has been approved by CEN as a EN ISO/IEC 15438:2010 without any modification.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

Contents

Page

Introduction vi
1 Scope 1
2 Normative references 1
3 Terms and definitions 2
4 Symbols, operations and abbreviated terms 3
4.1 Symbols 3
4.2 Mathematical operations 4
4.3 Abbreviated terms 4
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 PDF417 symbol parameters 7
5.2.2 Row parameters 7
5.2.3 Codeword sequence 7
5.3 Basic encodation 8
5.3.1 Symbol character structure 8
5.3.2 Start and stop characters 9
5.4 High level (data) encodation 10
5.4.1 Function codewords 10
5.4.2 Text Compaction mode 13
5.4.3 Byte Compaction mode 17
5.4.4 Numeric Compaction mode 19
5.4.5 Advice to select the appropriate compaction mode 21
5.4.6 Treatment of PDF417 reserved codewords 21
5.5 Extended Channel Interpretation 21
5.5.1 Encoding the ECI assignment number 22
5.5.2 Pre-assigned and default Extended Channel Interpretations 23
5.5.3 Encoding ECI sequences within compaction modes 23
5.5.4 Post-decode protocol 25
5.6 Determining the codeword sequence 25
5.7 Error detection and correction 26
5.7.1 Error correction level 26
5.7.2 Error correction capacity 26
5.7.3 Defining the error correction codewords 27
5.8 Dimensions 27
5.8.1 Minimum width of a module (X) 27
5.8.2 Row height (Y) 28
5.8.3 Quiet zones 28
5.9 Defining the symbol format 28
5.9.1 Defining the aspect ratio of the module 28
5.9.2 Defining the symbol matrix of rows and columns 28
5.10 Generating the error correction codewords 30
5.11 Low level encodation 31
5.11.1 Clusters 32
5.11.2 Determining the symbol matrix 32
5.11.3 Determining the values of the left and right row indicators 32

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

5.11.4 Row encoding ... 33
5.12 Compact PDF417 ... 33
5.13 Macro PDF417 ... 33
5.13.1 Compaction modes and Macro PDF417 ... 34
5.13.2 ECIs and Macro PDF417 ... 34
5.14 User guidelines ... 34
5.14.1 Human readable interpretation ... 34
5.14.2 Autodiscrimination capability ... 34
5.14.3 User-defined application parameters ... 34
5.14.4 PDF417 symbol quality ... 35
5.15 Reference decode algorithm ... 35
5.16 Error detection and error correction procedure ... 35
5.17 Transmitted data ... 35
5.17.1 Transmitted data in the basic (default) interpretation ... 35
5.17.2 Transmission protocol for Extended Channel Interpretation (ECI) ... 36
5.17.3 Transmitted data for Macro PDF417 ... 37
5.17.4 Transmission of reserved codewords using the ECI protocol ... 37
5.17.5 Symbology identifier ... 37
5.17.6 Transmission using older protocols ... 37

Annex A (normative) **Encoding/decoding table of PDF417 symbol character bar-space sequences** ... 39

Annex B (normative) **The default character set for Byte Compaction mode** ... 55

Annex C (normative) **Byte Compaction mode encoding algorithm** ... 56

Annex D (normative) **Numeric Compaction mode encoding algorithm** ... 58

Annex E (normative) **User selection of error correction level** ... 60
E.1 Recommended minimum error correction level ... 60
E.2 Other user consideration of the error correction level ... 60

Annex F (normative) **Tables of coefficients for calculating PDF417 error correction codewords** ... 61

Annex G (normative) **Compact PDF417** ... 66
G.1 Description ... 66
G.2 Print quality ... 66

Annex H (normative) **Macro PDF417** ... 67
H.1 Macro PDF417 overview ... 67
H.2 Macro PDF417 syntax ... 67
H.3 High level encoding considerations ... 70
H.4 Encodation example ... 70
H.5 Macro PDF417 and the Extended Channel Interpretation protocol ... 71
H.6 Macro PDF417 data transmission ... 72

Annex I (normative) **Testing PDF417 symbol quality** ... 75

Annex J (normative) **Reference decode algorithm for PDF417** ... 76
J.1 Initialisation ... 76
J.2 Reference decode algorithm for line decoding ... 76
J.3 Filling the matrix ... 78
J.4 Interpretation ... 79

Annex K (normative) **Error correction procedures** ... 80

Annex L (normative) **Symbology identifier** ... 82

Annex M (normative) **Transmission protocol for decoders conforming with original PDF417 standards** ... 83
M.1 Basic Channel mode ... 83
M.2 GLI encoded symbols ... 83
M.3 Macro PDF417 symbols ... 85
M.4 Transmission of reserved codewords using the original PDF417 protocol ... 86
M.5 Achieving compatibility between old and new PDF417 equipment ... 86

Annex N (informative) **Algorithm to minimise the number of codewords** ... 89

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

Annex O (informative) **Guidelines to determine the symbol matrix** **91**
O.1 Parameters affecting the determination of the matrix **91**
O.2 Guidelines should any parameters not be achieved **94**

Annex P (informative) **Calculating the coefficients for generating the error correction codewords – worked example** **95**

Annex Q (informative) **Generating the error correction codewords - worked example** **96**

Annex R (informative) **Division circuit procedure for generating error correction codewords** **99**

Annex S (informative) **Additional guidelines for the use of PDF417** **100**
S.1 Autodiscrimination compatibility **100**
S.2 Pixel-based printing **100**

Bibliography **102**

Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

Introduction

The technology of bar coding is based on the recognition of patterns of bars and spaces of defined dimensions. There are various methods of encoding information in bar code form, known as symbologies, and the rules defining the translation of characters into bars and space patterns and other essential features are known as the symbology specification.

Manufacturers of bar code equipment and users of bar code technology require publicly available standard symbology specifications to which they can refer when developing equipment and application standards. It is the intent and understanding of ISO/IEC that the symbology presented in this International Standard is entirely in the public domain and free of all user restrictions, licences and fees.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

Information technology — Automatic identification and data capture techniques — PDF417 bar code symbology specification

1 Scope

This International Standard specifies the requirements for the bar code symbology known as PDF417. It specifies PDF417 symbology characteristics, data character encodation, symbol formats, dimensions, error correction rules, reference 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 15424, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

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 — Part 2: Optically readable media (ORM)*

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

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

[1] Published by AIM Global, 125 Warrendale-Bayne Road, Suite 100, Warrendale, PA 15086, USA.

Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

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
basic channel model
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

NOTE A decoder complying with this model operates in Basic Channel Mode.

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

3.3
cluster
any of three mutually exclusive subsets of PDF417 symbol characters; the symbol characters in a given cluster conform with particular structural rules which are used in decoding the symbology

3.4
compaction mode
any of three data compaction algorithms in PDF417 (Text, Numeric and Byte Compaction modes) which are used to map 8-bit data bytes efficiently to PDF417 codewords

3.5
e-distance
distance from leading edge of an element to leading edge of the next similar element, or from trailing edge to trailing edge

3.6
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 error correction level, to be corrected

3.7
Extended Channel Interpretation
ECI
procedure within some symbologies, including PDF417, 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.

3.8
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

NOTE A decoder complying with this model operates in Extended Channel Mode.

3.9
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

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

3.10
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 PDF417-specific precursor to the symbology-independent ECI system.

3.11
Macro PDF417
procedure in the PDF417 symbology logically to distribute data from a computer file across a number of related PDF417 symbols

NOTE 1 The procedure considerably extends the data capacity beyond that of a single symbol.

NOTE 2 This procedure is similar to the Structured Append feature in other symbologies.

3.12
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

3.13
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

3.14
Row Indicator codeword
PDF417 codeword adjacent to the start or stop character in a row, which encodes information about the structure of the PDF417 symbol in terms of the row identification, total number of rows and columns, and the error correction level

3.15
Symbol Length Descriptor
first codeword in a PDF417 symbol, which encodes the total number of data codewords in the symbol

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.

A symbol aspect ratio (height to width) of a PDF417 symbol

b element width in a symbol character

c number of columns in the symbol in the data region (excluding start, stop and row indicator codewords)

d data codeword including all function codewords

E error correction codeword

e edge to similar edge dimension in a symbol character

F row number

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

f number of substitution errors

H height of symbol including quiet zone

K cluster number

k number of error correction codewords

L left row indicator

l number of erasures

m number of source data codewords prior to the addition of the Symbol Length Descriptor and any pad codewords

n total number of data codewords including Symbol Length Descriptor and any pad codewords

p pitch or width of a symbol character

Q_H horizontal quiet zone

Q_V vertical quiet zone

R right row indicator

r number of rows in the symbol

s error correction level

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, add the value of the divisor to make it 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.

ECI Extended Channel Interpretation

GLI Global Label Identifier

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

5 Requirements

5.1 Symbology characteristics

5.1.1 Basic characteristics

PDF417 is a bar code symbology with the following basic characteristics:

a) Encodable character set:

1) Text Compaction mode (see 5.4.1.5) permits all printable ASCII characters to be encoded, i.e. values 32 - 126 inclusive in accordance with ISO/IEC 646 (IRV), 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 (at error correction level 0): 925 data codewords which can encode:

1) Text Compaction mode: 1 850 characters (at 2 data characters per codeword).

2) Byte Compaction mode: 1 108 characters (at 1,2 data characters per codeword).

3) Numeric Compaction mode: 2 710 characters (at 2,93 data characters per codeword)

At the minimum recommended error correction level, there is a maximum of 863 data codewords which can encode:

4) Text Compaction mode: 1 726 characters (at 2 data characters per codeword).

5) Byte Compaction mode: 1 033 characters (at 1,2 data characters per codeword).

6) Numeric Compaction mode: 2 528 characters (at 2,93 data characters per codeword).

d) Symbol size:

1) Number of rows: 3 to 90.

2) Number of columns: 1 to 30.

3) Width in modules: 90X to 583X including quiet zones.

4) Maximum codeword capacity: 928 codewords.

5) Maximum data codeword capacity: 925 codewords.

Since the number of rows and the number of columns are selectable, the aspect ratio of a PDF417 symbol may be varied when printing to suit the spatial requirements of the application.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

e) Selectable error correction: 2 to 512 codewords per symbol (see 5.7).

f) Non-data overhead:

   1) Per row: 73 modules, including quiet zones.

   2) Per symbol: a minimum of 3 codewords, represented as symbol characters.

g) Code type: continuous, multi-row two-dimensional.

h) Character self-checking: Yes.

i) Bi-directionally decodable: Yes.

5.1.2 Summary of additional features

Additional features which are inherent or optional in PDF417 are summarised below:

a) **Data compaction**: (inherent) 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 5.4.1.5 to 5.4.4).

b) **Extended Channel Interpretations**: (optional) These mechanisms allow up to 811 800 different data character sets or interpretations to be encoded (see 5.5).

c) **Macro PDF417**: (optional) This mechanism allows files of data to be represented logically and consecutively in a number of PDF417 symbols. Up to 99 999 different PDF417 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**: (inherent) PDF417 can be decoded by measuring elements from edge to similar edge (see 5.3.1).

e) **Cross row scanning**: (inherent) The combination of the following three characteristics in PDF417 facilitates cross row scanning:

   — being synchronised horizontally, or self clocking

   — row identification

   — being synchronised vertically, 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) **Error correction**: (inherent) A user may define one of 9 error correction levels. All but Level 0 not only detect errors but also can correct erroneously decoded or missing codewords (see 5.7).

g) **Compact PDF417**: (optional) In relatively ‘clean' environments, it is possible to reduce some of the row overhead to improve the symbol density (see 5.12).

NOTE In earlier specifications of PDF417, Compact PDF417 was called Truncated PDF417. Compact PDF417 is the preferred term to avoid confusion with the more general use of the term ‘truncated'.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

5.2 Symbol structure

5.2.1 PDF417 symbol parameters

Each PDF417 symbol consists of a stack of vertically aligned rows with a minimum of 3 rows (maximum 90 rows). Each row shall include a minimum of 1 symbol character (maximum 30 symbol characters), excluding start, stop and row indicator columns. The symbol shall include a quiet zone on all four sides. Figure 1 illustrates a PDF417 symbol encoding the text: PDF417 Symbology Standard.

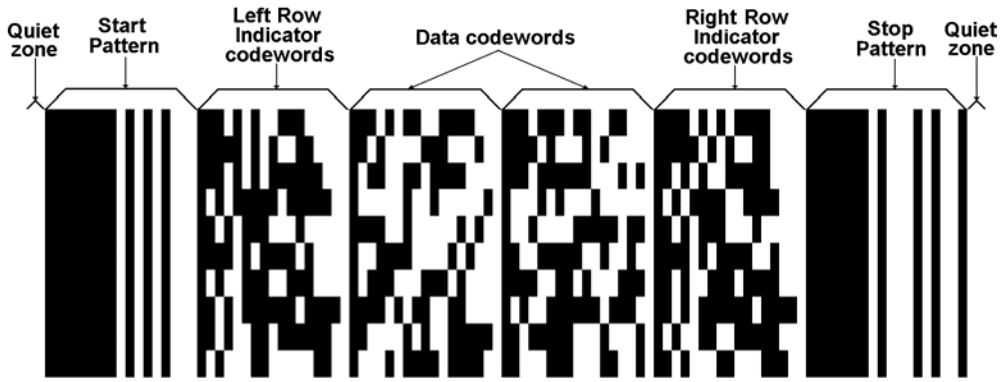

Figure 1 — PDF417 symbol structure

5.2.2 Row parameters

Each PDF417 row shall comprise:

a) leading quiet zone

b) start character

c) left row indicator symbol character

d) 1 to 30 symbol characters

e) right row indicator symbol character

f) stop character

g) trailing quiet zone

NOTE The number of symbol characters (or codewords) defined in item ‘d' above is equal to the number of data columns in the PDF417 symbol.

5.2.3 Codeword sequence

A PDF417 symbol may contain up to 928 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, the Symbol Length Descriptor, shall always encode the total number of data codewords in the symbol, including the Symbol Length Descriptor itself, data codewords and pad codewords, but excluding the number of error correction codewords.

b) The data codewords shall follow, from the most significant encodable character. Function codewords may be inserted to achieve data compaction.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

c) Pad codewords to enable the codeword sequence to be represented in a rectangular matrix. Pad codewords may also be used to fill additional complete rows to achieve an aspect ratio desired or as specified by the application.

d) An optional Macro PDF417 Control Block.

e) Error correction codewords for error detection and correction.

The codewords are arranged with the most significant codeword adjacent to the Symbol Length Descriptor, and are encoded from left to right and from top row to bottom. Figure 2 illustrates in layout format the sequence for a symbol like that shown in Figure 1. In Figure 2 an error correction level of 1 has been used and one pad character was needed to completely fill the symbol matrix.

	L_1	d_{15}	d_{14}	R_1	
	L_2	d_{13}	d_{12}	R_2	
	L_3	d_{11}	d_{10}	R_3	
S	L_4	d_9	d_8	R_4	S
T	L_5	d_7	d_6	R_5	T
A	L_6	d_5	d_4	R_6	O
R	L_7	d_3	d_2	R_7	P
T	L_8	d_1	d_0	R_8	
	L_9	E_3	E_2	R_9	
	L_{10}	E_1	E_0	R_{10}	

Figure 2 — PDF417 Example of Symbol Layout Schematic

where L, R, d and E are as defined in clause 4

d_{15} = Symbol Length Descriptor (in this example, with a value of 16)

d_{14} to d_1 = encoded representation of data

d_0 = pad codeword

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

5.3 Basic encodation

5.3.1 Symbol character structure

Each PDF417 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. PDF417 symbol characters can be decoded by measuring the e-distances within the character.

Each symbol character is defined by an 8-digit bar-space 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.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

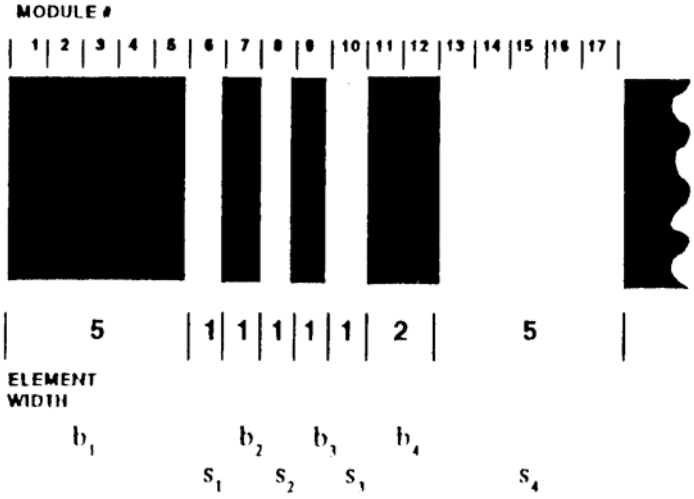

Figure 3 — A PDF417 symbol character

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

The codewords are represented by three mutually exclusive symbol character sets, or clusters. Each cluster encodes the 929 available PDF417 codewords into different bar-space patterns so that one cluster is distinct from another. The cluster numbers are 0, 3, and 6. The cluster definition applies to all PDF417 symbol characters, except for start and stop characters.

The cluster number K is defined by the following formula:

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

Where b_1, b_2, b_3 and b_4 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 characters

The start and stop characters shall be composed as defined in Table 1 and illustrated in Figure 4:

Table 1 — Bar-space sequence for Start and Stop Characters

Character	Bar-space sequence								
	B	S	B	S	B	S	B	S	B
Start	8	1	1	1	1	1	1	3	
Stop	7	1	1	3	1	1	1	2	1

NOTE 1 The PDF417 stop and start characters are unique in having elements more than 6 modules wide.

NOTE 2 The stop character has one extra single module bar element.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

The start and stop characters shall have the same bar-space sequence for all rows.

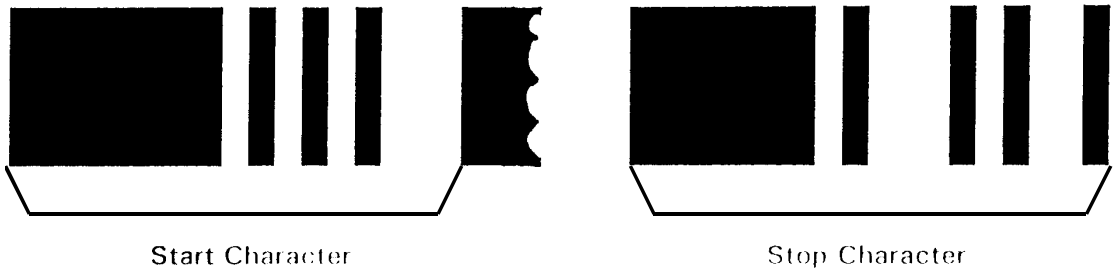

Figure 4 — PDF417 Start and Stop Characters

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 efficient high level encoding. Three modes are defined below, each of which defines a particular efficient mapping between user defined data and codeword sequences. PDF417 has three data compaction modes:

— Text Compaction mode (see 5.4.1.5).

— 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 PDF417 symbol.

900 codewords (0 to 899) 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.

PDF417 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 and 5.4.1.4).

At present codewords 903 to 912, 914 to 917, and 919 are reserved. Table 2 defines the complete list of assigned and reserved function codewords. Their functions are defined in 5.4.1.1 to 5.4.1.5. See 5.4.6 for the treatment of reserved codewords.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

Table 2 — Assignments of PDF417 function codewords

Codeword	Function	Refer to subclause
900	mode latch to Text Compaction mode	5.4.1.1
901	mode latch to Byte Compaction mode	5.4.1.1, 5.4.3.1
902	mode latch to Numeric Compaction mode	5.4.1.1
903 to 912	Reserved	
913	mode shift to Byte Compaction mode	5.4.1.1
914 to 917, 919	Reserved	
918	linkage flag to associated linear component, in a composite symbol (other than an EAN.UCC Composite symbol)	5.4.1.5
920	linkage flag to associated linear component, in an EAN.UCC Composite symbol	5.4.1.5
921	reader initialisation	5.4.1.4
922	terminator codeword for Macro PDF control block	5.13
923	sequence tag to identify the beginning of optional fields in the Macro PDF control block	5.13
924	mode latch to Byte Compaction mode (used differently from 901)	5.4.1.1, 5.4.3.1
925 to 927	identifier for an Extended Channel Interpretation (ECI)	5.5
928	Macro marker codeword to indicate the beginning of a Macro PDF Control Block	5.13

5.4.1.1 Function codewords for mode switching

In one PDF417 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 for this purpose. 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 3 — 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 to be used to switch to the defined mode.

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

Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

Table 4 — Mode Transition Table, Showing Codewords and Their Function

Original Mode	Destination Mode		
	Text	Byte	Numeric
Text	900 mode latch	913 mode shift 901 mode latch 924 mode latch	902 mode latch
Byte	900 mode latch	901 mode latch 924 mode latch	902 mode latch
Numeric	900 mode latch	901 mode latch 924 mode latch	902 mode latch

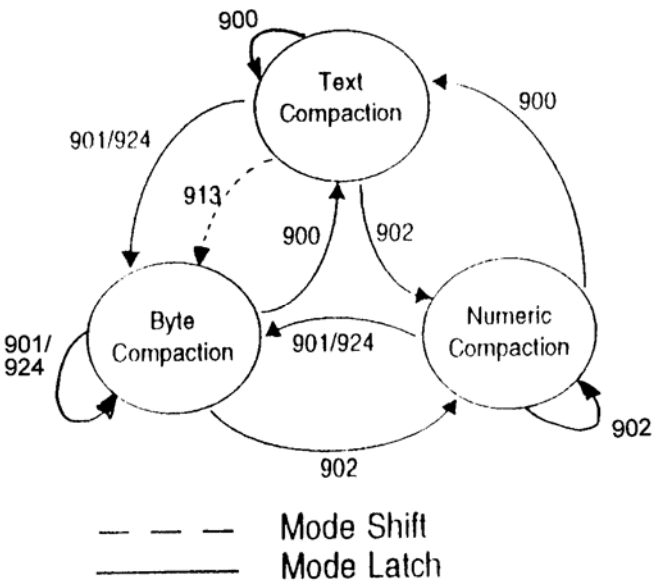

Figure 5 — Available Mode Switching

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 Macro PDF417

Macro PDF417 symbols (see 5.13) shall use codeword 928 at the start of the Macro PDF417 Control Block. Codewords 922 and 923 are used for special functions in Macro PDF417.

5.4.1.4 Function codeword for reader initialisation

Codeword 921 shall be used to instruct the reader to interpret the data contained within the symbol as programming for reader initialisation. Codeword 921 shall appear as the first codeword after the Symbol

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

Length Descriptor. In the case of a Macro PDF417 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 linkage flags in composite symbols

Codeword 920 shall be used as a linkage flag to signal the presence of an associated EAN.UCC 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 two-dimensional composite component and transmit only the data from the associated linear component.

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.

The default compaction mode for PDF417 in effect at the start of each symbol shall always be Text Compaction mode Alpha sub-mode (uppercase alphabetic). 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 5.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

Table 5 — 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
as = shift to alpha
ll = latch to lower
ml = latch to mixed
pl = latch to punctuation
ps = shift to punctuation

NOTE The Char columns above show the default interpretation of 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)

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

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 5. For each pair of base 30 values, the first or left value shall be designated the more significant value *h*, the other shall be designated the less significant value *l*.

The encoded PDF417 codeword is defined using the following formula:

$$d = h \times 30 + l$$

where: *d* is as defined in clause 4

The formula shall also apply to the base 30 values for shifts and latches within the Text Compaction mode. Appropriate latch and shift values shall be used between sub-modes. If the encoding of the character sequence does not result in an even number of base 30 values, see 5.4.2.4 for the specific mechanism to use.

The following example illustrates how compaction is achieved in Text Compaction mode.

EXAMPLE Data to be encoded: **PDF417**

Table 6 — Example of Text Compaction Encoding

Character Pairs	*h*	*l*	*h* x 30 + *l*	Codeword
P D	15	3	15 x 30 + 3	453
F **ml**	5	28	5 x 30 + 28	178
4 1	4	1	4 x 30 + 1	121
7 **ps**	7	29	7 x 30 + 29	239

NOTE 1 **ml** (latch to mixed sub-mode) is used to switch to encode the numeric characters

NOTE 2 **ps** is used as a pad value in this example, other shift and latch values can be used (see 5.4.2.4)

The data **PDF417** is represented by codewords 453, 178, 121, 239

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

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

— **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 5. Table 7 shows the transition table between Text Compaction sub-modes; Figure 6 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 pl
Punctuation	al			

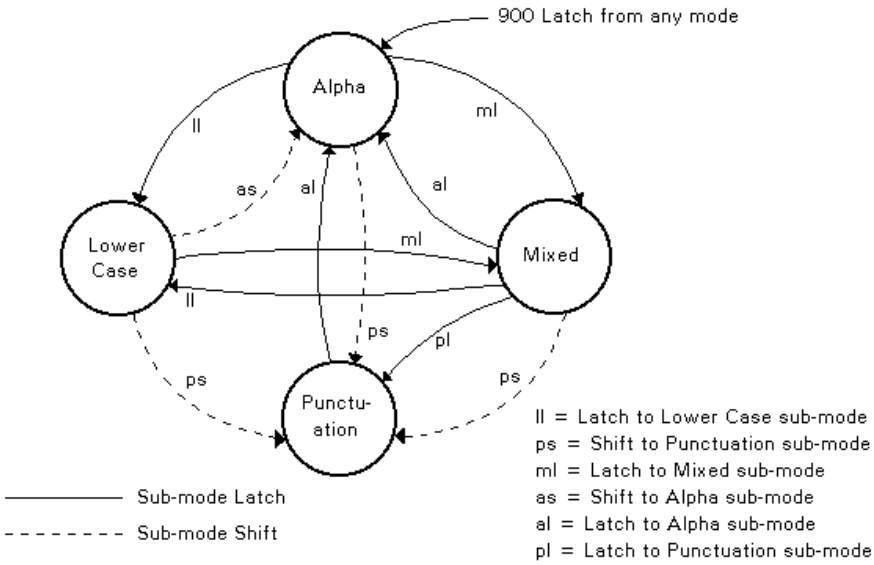

Figure 6 — Text Compaction Sub-Mode Switching

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 6. 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.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

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.

   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 therefore will 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 Macro PDF417 Control Block)

— 923 (Beginning of Macro PDF417 Optional Field)

— 922 (Macro PDF417 Terminator)

The last three codewords only occur within the Macro PDF417 Control Block of a Macro PDF417 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 previous PDF417 specifications, 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

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

PDF417 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.

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 Byte Compaction characters to be encoded is an integer multiple of 6

— **mode latch 901** shall be used when the total number of Byte Compaction characters to be encoded is not a multiple of 6

— **mode shift 913** can be used instead of codeword 901 when 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

1) Establish the total number of Byte Compaction characters.

2) If a perfect multiple of 6, mode latch 924 shall be used; else mode latch 901 shall be used.

3) Sub-divide the number of Byte Compaction characters 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.

4) 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).

5) 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.

6) Repeat from Step 3 as necessary.

7) For the remaining Byte Compaction characters when mode latch 901 is used, (i.e. when the last group is less than 6 Byte Compaction characters) 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).

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

5.4.3.4 Switching from Byte Compaction

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 Macro PDF417 Control Block)

— 923 (Beginning of Macro PDF417 Optional Field)

— 922 (Macro PDF417 Terminator)

The last three codewords only occur within the Macro PDF417 Control Block of a Macro PDF417 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.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

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:

Codewords = INT (number of digits / 3) +1

EXAMPLE For a 28 digit sequence

INT (28 / 3) + 1 = 9 + 1 = 10 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 Macro PDF417 Control Block)

— 923 (Beginning of Macro PDF417 Optional Field)

— 922 (Macro PDF417 Terminator)

The last three codewords only occur within the Macro PDF417 Control Block of a Macro PDF417 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.

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

5.4.5 Advice to select the appropriate compaction mode

All basic implementations for printing and scanning PDF417 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 5; 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 PDF417 reserved codewords

5.4.6.1 Overview

PDF417 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

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. Specifically:

— When a new signalling codeword (as opposed to a new compaction mode 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 both the original and this PDF417 standard, regardless of whether that equipment employs the original or the new transmission protocol.

— At the receiving system, the ECI decoder will process the signal ECIs (i.e. Macro Control Blocks and escaped uninterpreted codewords) before the encodable ECIs (such as character sets). Thus, the encoder should take into account the order of operations as follows:

1) The Macro 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 signalling ECI, 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 PDF417. ECIs are assigned by AIM Global, Inc.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

NOTE Originally a symbology specific scheme called Global Label Identifiers (GLIs) was defined for PDF417. Encoding and decoding ECIs is identical to earlier specifications for PDF417 GLIs. However, the transmission protocol for decoded messages according to earlier PDF417 specifications for GLIs is different from the transmission protocol for ECIs. There are also differences with respect to the use of interpretive ECIs with Macro PDF417. 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 PDF417:

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 Macro PDF417

e) transmission of uninterpreted PDF417 codewords

Transmission of the Extended Channel Interpretation protocol is fully specified in the AIM International standard 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 PDF417 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 were 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.

Table 8 — Encoding ECI assignment numbers

ECI assignment number	Codeword sequence	Codewords	Ranges
000000 to 000899	C_0 C_1	927 ECI_no	 C_1 = (0 to 899)
000900 to 810899	C_0 C_1 C_2	926 ECI_no div 900 - 1 ECI_no mod 900	 C_1 = (0 to 899) C_2 = (0 to 899)
810900 to 811799	C_0 C_1	925 ECI_no - 810 900	 C_1 = (0 to 899)

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

There are 811 800 possible ECI assignment numbers available in PDF417.

NOTE The encodation method is identical to the GLI scheme incorporated in the original AIM USA (1994) and AIM Europe (1994) PDF417 specifications.

The following example illustrates the encodation:

EXAMPLE ECI = 013579

Codewords: [926] [(13 579 div 900) - 1] [13 579 mod 900]

= [926] [15 - 1] [79]

= [926] [14] [79]

5.5.2 Pre-assigned and default Extended Channel Interpretations

The following ECIs, ECI 000000 to ECI 000003, have been pre-assigned to be backwards compatible with existing symbology specifications, including PDF417.

— ECI 000000 (equates to original GLI 0) represents the default encodation scheme of encoders compliant with the original PDF417 standards.

— ECI 000001 (equates to original GLI 1) represents the GLI encodation scheme of a number of symbologies with characters 0 to 127 being identical to those of ISO/IEC 646:1991, International Reference Version (equivalent to ANSI X3.4) and characters 128 to 255 being identical to those values of ISO/IEC 8859-1.

NOTE ECI 000000 (equivalent to GLI 0) and ECI 000001 (equivalent to GLI 1) require a return-to-GLI 0 logic at the beginning of each encoded symbol of a Macro PDF417 set of symbols. This protocol is not adopted for other Extended Channel Interpretations.

— ECI 000002 has an equivalent code table to ECI 000000, without the return-to-GLI 0 logic.

— ECI 000003 has an equivalent code table to ECI 000001, without the return-to-GLI 0 logic. ECI 000003 is the default encodation scheme for encoders fully compliant with this edition of this standard.

ECI 000000 and ECI 000001 shall not be encoded in the same PDF417 symbol or Macro PDF417 symbol set as other ECIs, except for user defined ECIs. ECI 000002 and ECI 000003 provide the compatible alternatives to ECI 000000 and ECI 000001 respectively. ECI 000000 and ECI 000001 should not be used in new applications.

5.5.3 Encoding ECI sequences within compaction modes

The general encodation principle is that ECIs are applied to the source data byte stream (to signal various interpretations) producing a modified byte stream that is encoded into PDF417 symbols using the symbology's compaction modes for efficiency. The ECI encoding, and symbology specific compaction, form two independent logical layers of the process.

Although ECI assignments and compaction modes may generally be intermixed, some combinations can produce illogical or ambiguous behaviour. The following sections define how ECIs may be incorporated without ambiguity by specifying the valid placements of ECI escape sequences.

5.5.3.1 ECIs and Text Compaction mode

An ECI escape sequence may be placed anywhere within Text Compaction mode. The sub-mode invoked immediately prior to the ECI escape sequence is preserved for the encodation immediately after it. Thus, sub-mode latches and shifts are preserved across an ECI escape sequence; and thus a sub-mode shift immediately before an ECI escape sequence is not ignored.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

5.5.3.2 ECIs and Byte Compaction mode using mode latch 924 and 901

If encoding in Byte Compaction mode using mode latch 924, an ECI escape sequence may be positioned by an encoder immediately following codeword 924, or at any 5-codeword boundary thereafter. This is necessary to provide an unambiguous position in the decoded byte stream for the decoder to place the escape sequence.

If the decoder is in the 924 version of Byte Compaction mode and finds an ECI escape sequence following a 5-codeword group, it shall output the six data bytes associated with the codewords before the escape sequence, output the escape sequence, and then continue collecting codewords for decoding in Byte Compaction mode. If the decoder encounters an ECI escape sequence at other than these prescribed locations, it shall treat the symbol as invalid.

If encoding in Byte Compaction mode using mode latch 901, an ECI escape sequence may be positioned:

— Immediately following codeword 901

— Immediately after any set of five codewords encoding six bytes

— Immediately after any of the trailing single-byte codewords at the end of the sequence

NOTE The decoder cannot assume that, just because the ECI escape sequence follows a set of five codewords, the five codewords encode six bytes, since an input stream of length 6N+5 (where N is an integer) will have a final set of five codewords that encode only five bytes, one byte per codeword. The decoder must, therefore, scan forward in the symbol past the ECI escape sequence to determine where the 901 mode terminates, as defined in 5.4.3.4. Based on this information, it can then determine how the group of five codewords have been encoded.

Figure 7 illustrates valid locations for ECI escape sequences when encoding in Byte Compaction mode. If the decoder encounters an ECI escape sequence within the 5-codeword group, it shall treat the symbol as invalid.

[901] ♦ □ □ □ □ □ ♦ □ □ □ □ □ ♦ □ ♦ □ ♦ □ ♦ □ ♦ □ ♦

[924] ♦ □ □ □ □ □ ♦ □ □ □ □ □ ♦

5 codeword group 5 codeword group

where □ = Byte Compaction mode codeword

♦ = Valid location for ECI escape sequence

Figure 7 — Valid Locations for ECI Escape Sequences in Byte Compaction Mode

5.5.3.3 ECIs and Byte Compaction using mode shift 913

If encoding in Byte Compaction mode using mode shift 913, an ECI escape sequence may be placed:

— immediately preceding codeword 913

— immediately following codeword 913

— immediately following the codeword after codeword 913

In the first two cases, the ECI escape sequence is output before the encoded byte, while in the last case, the escape sequence is output following the encoded byte.

5.5.3.4 ECIs and Numeric Compaction mode

An ECI escape sequence shall not be placed within a group of codewords being processed through the base 10 to base 900 conversion as defined in 5.4.4.2. It may only be placed within a Numeric Compaction mode

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010

region at a boundary between (the typically) 15-codeword groups. This is necessary to provide an unambiguous position in the decoded digit stream for the decoder to place the escape sequence.

Thus, an ECI escape sequence may only be placed:

— immediately after codeword 902

— after the 15th codeword

— after the 30th codeword

— etc.

If the encoder needs to place an ECI escape sequence at a location that does not result in a multiple of 15 codewords, it shall treat the numeric block before the ECI as a complete entity, as defined in 5.4.4.2 step 2. It shall re-invoke the Numeric Compaction mode by placing another codeword 902 in the stream followed by the ECI escape sequence.

If the decoder finds an ECI escape sequence on one of the boundary points defined above, it shall emit the data bytes associated with the codewords before the escape sequence (if any), then emit the escape sequence, and then continue collecting codewords for decoding in Numeric Compaction mode. If the decoder encounters an ECI escape sequence at other than the prescribed locations, it shall treat the symbol as invalid.

5.5.3.5 Combining ECIs

Two or more ECI escape sequences (e.g. assignment numbers) may be placed at any point where one ECI can be validly located; providing that no codewords, other than those used to encode the ECI escape sequence, are placed between them.

5.5.4 Post-decode protocol

The protocol for transmitting ECI data shall be as defined in 5.17.2. When transmitting ECIs, symbology identifiers (see 5.17.5) shall be fully implemented and the appropriate symbology identifier shall be transmitted as a preamble.

5.6 Determining the codeword sequence

The encoding process generates a sequence of codewords defined as:

$$d_{n-1} \ldots d_0$$

where: d = data codeword including the Symbol Length Descriptor and all function codewords

n = total number of data (and pad) codewords including the Symbol Length Descriptor, but excluding the error correction codewords

The Symbol Length Descriptor shall be the first data codeword and designated d_{n-1}. Its value shall be equal to the total number of data codewords $\boldsymbol{n}$; this count shall include the Symbol Length Descriptor itself, and thus shall be in the range of 1 to 926.

During the encoding process, sequences of codewords will be established. Like the original data itself, the most significant data shall appear first, for example textual and numeric data reads from the left to the right. The sequence of codewords shall be that the most significant data codeword containing encoded data is the one designated d_{n-2}. The final data (or pad) codeword is the one designated d_0.

The process used to determine the symbol matrix of rows and columns (see 5.9.2) can require the addition of trailing pad codewords to the end of the data codeword sequence.

AVA Documents
Document Sharing Hub

Full document details: BS EN ISO/IEC 15438:2010

5.7 Error detection and correction

Each PDF417 symbol contains at least two error correction codewords. The Error Correction codewords provide capability for both error detection and correction.

5.7.1 Error correction level

The error correction level for a PDF417 symbol is selectable at the time of symbol creation. Table 9 shows the number of error correction codewords for each error correction level.

Table 9 — Error Correction Levels and Error Correction Codewords

Error Correction Level	Total Number of Error Correction Codewords
0	2
1	4
2	8
3	16
4	32
5	64
6	128
7	256
8	512

5.7.2 Error correction capacity

Error correction can be used to compensate for defects in the label and misreads during the decode procedure. For any given error correction level, a particular number of error correction codewords is incorporated into the PDF417 symbol. The error correction codeword algorithm used allows two types of error to be recovered:

— an **erasure**, which is a missing or undecodable codeword at a known position,

— a **substitution error**, which is an erroneously decoded codeword at an unknown position.

The error correction scheme requires one error correction codeword to rectify an erasure and two to recover a substitution error. Thus a given error correction level can rectify any combination of substitution errors and erasures which satisfy the following equations:

$$l + 2f \leq 2^{s+1} - 2$$

where: l, f and s are as defined in 4.1

However, if most of the error correction capacity is used to correct erasures, the possibility of undetected errors is increased. For this reason, whenever there are fewer than 4 errors corrected (except when s = 0), the error correction capacity should be reduced as follows:

$$l + 2f \leq 2^{s+1} - 3$$

Documents
Document Sharing Hub
Full document details: BS EN ISO/IEC 15438:2010