INTERNATIONAL STANDARD

ISO/IEC 16023

First edition
2000-05-01

Information technology — International symbology specification — MaxiCode

Technologies de l'information — Spécification internationale des symboles — MaxiCode

Reference number
ISO/IEC 16023:2000(E)

Contents

Introduction 1

1 Scope 1

2 Normative References 1

3 Definitions and Mathematical Symbols 1

3.1 Definitions 1
3.1.1 Codeword 1
3.1.2 Extended Channel Interpretation (ECI) 1
3.1.3 Mode Indicator 1
3.1.4 Module 2
3.2 Mathematical Symbols and Operations 2

4 Requirements 2

4.1 Symbology Characteristics 2
4.1.1 Basic Characteristics 2
4.1.2 Summary of Additional Features 3
4.2 Symbol Description 4
4.2.1 Symbol Structure 4
4.2.2 Symbol Character and Module Sequence 5
4.3 General Encodation Procedures 6
4.4 Character Assignments 7
4.4.1 Codeword Representation 7
4.4.2 Default Character Interpretation 7
4.4.3 Code Sets 7
4.4.4 Symbology Control Characters 8
4.5 User Considerations for Encoding Data in a MaxiCode Symbol 9
4.5.1 User Selection of Error Correction Level 9
4.5.2 User Selection of Mode 9
4.5.3 User Selection of Extended Channel Interpretation 10
4.5.4 User Selection of Structured Append 10
4.5.5 User Assessment of Encodation Capacity 10
4.6 Extended Channel Interpretation 10
4.6.1 ECI and Modes 2 and 3 10
4.6.2 Encodation Modes and ECIs 10
4.6.3 Encoding ECIs in MaxiCode 11
4.6.4 ECIs and Structured Append 11
4.6.5 Post-Decode Protocol 11
4.7 Message Structure 11
4.7.1 Primary Message 11
4.7.2 Secondary Message 11
4.7.3 Structuring the Data 12
4.8 Modes 12
4.8.1 Mode 0: Obsolete 12
4.8.2 Mode 1: Obsolete 12
4.8.3 Modes 2 and 3: Structured Carrier Message 12
4.8.4 Mode 4: Standard Symbol 13
4.8.5 Mode 5: Full EEC 13
4.8.6 Mode 6: Reader Programming 13
4.8.7 Mode Indicators 13

4.9 Structured Append 13
4.9.1 Basic Principles 13
4.9.2 Structured Append and Modes 2 and 3 13
4.9.3 Structured Append in Modes 4 to 6 14
4.9.4 Buffered and Unbuffered Operation 14
4.10 Error Detection and Correction 14
4.10.1 Enhanced Error Correction (EEC) in the Primary Message 14
4.10.2 Error Correction in the Secondary Message 14
4.10.3 Generating the Error Correction Codewords 14
4.10.4 Error Correction Capacity 15
4.11 Dimensions 15
4.11.1 Symbol Dimensions 15
4.11.2 Hexagonal Module Dimensions 15
4.11.3 Dark Hexagon Dimensions and Tolerances 16
4.11.4 Finder Pattern Dimensions 16
4.11.5 Quiet Zones 17
4.11.6 Overall Symbol Size 17
4.11.7 Practical Printing Guidance 17
4.12 User Guidelines 17
4.12.1 Human Readable Interpretation 17
4.12.2 Autodiscrimination Capability 17
4.13 Symbol Quality 17
4.13.1 Obtaining the Test Image 17
4.13.2 Symbol Quality Parameters 17
4.13.3 Overall Symbol Grade 19
4.13.4 Process Control Measurements 19
4.14 Reference Decode Algorithm 19
4.15 Transmitted Data 22
4.15.1 Basic Interpretation 22
4.15.2 Protocol for Extended Channel Interpretation 22
4.15.3 Symbology Identifier 22
4.15.4 Transmitted Data Example 23

Annexe A (Normative) 24

MaxiCode Basic Character Encodation: Default Character Set 24

Annexe B (Normative) 26

Mode 2 and Mode 3: Structured Carrier Message 26
B.1 The Structure of the Primary Message 26
B.2 Modes 2 and 3 Messages Beginning with "[)>RS01GS" 27
B.2.1 Encoding 27
B.2.2 Decoding 27
B.3 Modes 2 and 3 Messages Not Beginning with "[)>RS01GS" 28
B.3.1 Encoding 28
B.3.2 Decoding 28
B.4 Modes 2 and 3 and Structured Append 28
B.4.1 Encoding Considerations 28
B.4.2 Decoding Considerations 28

Annexe C (Normative) 29

2D Matrix Bar Code Print Quality - Guideline 29
C.1 Obtaining the Test Image 29
C.2 Assessing Symbol Parameters 29
C.2.1 Decode 29
C.2.2 Symbol Contrast 30
C.2.3 “Print” Growth 30
C.2.4 Grid Nonuniformity 30
C.2.5 Unused Error Correction 31
C.3 Overall Symbol Grade 31

Annexe D (Normative) 31

Error Correction Algorithm 31

Annexe E (Normative) 32

Symbology Identifiers 32

Annexe F (Informative) 32

Use of Numeric Shift, Shift , Latch, and Lock-In Characters 32
F.1 Numeric Shift 32
F.2 Switching from Code Set A to Code Set B 33
F.3 Switching from Code Set B to Code Set A 33
F.4 Using Lock-In to Latch to Code Sets C, D or E 33
F.5 Illustrative Example 33

Annexe G (Informative) 34

User Assessment of Encodation Capacity 34

Annexe H (Informative) 35

A MaxiCode Encoding Example 35

Annexe J (Informative) 38

Practical Printing Considerations 38
J.1 12 dots per Millimeter 38
J.2 8 dots per Millimeter 39
J.3 Generic Rules for Other Pixel Sizes 40
J.4 Determining the Hexagon Font for a Given Dot Pitch 40

Annexe K (Informative) 41

Autodiscrimination Compatibility 41

Annexe L (Informative) 42

Useful Process Control Techniques 42
L.1 Symbol Contrast 42
L.2 Symbol Size 42
L.2.1 Checking Print Growth 42
L.2.2 Checking Finder Position and Orientation Patterns 42
L.2.3 Checking Overall Symbol Size 42
L.3 Symbol Distortion 43
L.4 Print Growth and Defects 43

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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 International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 16023 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

International Standard ISO/IEC 16023 was prepared by AIM International (as ANSI/AIM BC10) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Annexes A to E form a normative part of this International Standard. Annexes F to L are for information only.

Information technology — International symbology specification — MaxiCode

Introduction

MaxiCode is a fixed-size matrix symbology which is made up of offset rows of hexagonal modules arranged around a unique finder pattern.

Manufacturers of bar code equipment and users of the technology require publicly available standard symbology specifications to which they can refer when developing equipment and application standards. The publication of Symbology Specifications is designed to achieve this.

1 Scope

This specification defines the requirements for the symbology known as MaxiCode. It specifies the MaxiCode symbology characteristics, data character encodation, symbol formats, dimensions and print quality requirements, error correction rules, decoding algorithm, and user-selectable application parameters.

2 Normative References

This specification incorporates provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed below. The latest edition of the publication referred to applies.

EN796	Bar Coding : Symbology Identifiers
EN1556	Bar Coding : Terminology
ANSI X3.182	Bar Code Print Quality - Guideline (Same as EN1635 - Bar Coding : Test Specifications for Bar Code Symbols)
ANSI X3.4	Coded Character Sets - 7-bit American National Standard Code for Information Interchange (7-bit ASCII) (equivalent to the US national version of ISO 646)
ISO 3166	Codes for the Representation on Names of Countries
ISO/IEC 8859-1	Information Processing - 8-bit Single-byte Coded Graphic Character Sets - Part 1 (Latin Alphabet Number 1)

Guideline on Mode 0 for MaxiCode - AIM USA
ECI Assignments Document - AIM International.

3 Definitions and Mathematical Symbols

3.1 Definitions

For the purposes of this Standard the following definitions in EN1556 (Terminology) shall apply:

algorithm, application standard, ASCII, autodiscrimination, binary, bit, CCD, code page, code set, data character, data codeword, data region, data separator character, decode algorithm, decoder, error correction, finder pattern, human readable character, latch character, leading zeros, matrix symbology, modulo, numeric, omnidirectional, orientation pattern, overhead, pad character, pixel, quiet zone, reference decode algorithm, Reed-Solomon error correction, scanner, shift characters, structured append, symbol character, symbology, symbology identifier, X-dimension

The following definitions also apply to this specification. Although some of the terms below are defined in EN1556, the definitions which follow below are more appropriate for this specification.

3.1.1 Codeword

A symbol character value. An intermediate level of coding between source data and the graphical encodation in the symbol.

3.1.2 Extended Channel Interpretation (ECI)

A protocol used by some symbologies that allows the output data stream to have interpretations other than that of the default character set.

3.1.3 Mode Indicator

A group of modules, in MaxiCode, used to define the symbol structure, for example to specify the level of error correction employed in the symbol.

3.1.4 Module

A single cell in a matrix symbology used to encode one bit of data. In MaxiCode the module is a regular hexagonal shape.

3.2 Mathematical Symbols and Operations

For the purposes of this specification the mathematical symbols which follow shall apply:

c	codeword
H	vertical distance from the center of a module in the top row to the center of a module in the bottom row
L	distance from the center of the left-most module to the center of the right-most module in the top row
m	message character
n	total number of data codewords
s	symbol character
V	vertical height of a module
W	center to center distance between adjacent modules
X	horizontal width of a module
Y	vertical distance from the center line of modules in one row to the center line of modules in an adjacent row

For the purposes of this specification the mathematical operations which follow shall apply:

div	is the integer division operator
mod	is the integer remainder after division

4 Requirements

4.1 Symbology Characteristics

4.1.1 Basic Characteristics

MaxiCode is a matrix symbology with the following basic characteristics:

a. Encodable character set:

1. The default character set allows 256 international characters to be encoded:

    i. values 0-127, in accordance with ANSI X3.4, i.e. all 128 ASCII characters

    ii. values 128-255 in accordance with ISO 8859-1: Latin Alphabet No. 1

2. Numeric compaction allows 9 digits to be compacted in six codewords.

3. Various symbology control characters, for code switching and other control purposes, are included.

b. Codeword set:

1. The codeword set of 64 (2^6) values is used as an intermediate encodation layer between the data characters and symbol characters. The codewords form the basis for error correction calculations.

2. The codewords have the values 0-63; 000000 to 111111 in binary notation. Within each symbol character the most significant bit is the lowest numbered module as shown in Figures 1 and 5.

c. Representation of codewords in a MaxiCode symbol:

1. Each codeword is represented by 6 modules which are hexagonal in shape.

2. Information is represented in each module as a binary bit. A dark module is a one and a light module is a zero.

3. Generally the six modules are arranged in three rows of two modules, each ordered from upper right to lower left. Figure 1 identifies the modules of a typical symbol character.

4. Because of the structure of the MaxiCode symbol, symbol characters 1 - 9 and 137 - 144 have a different arrangement (See Figure 4).

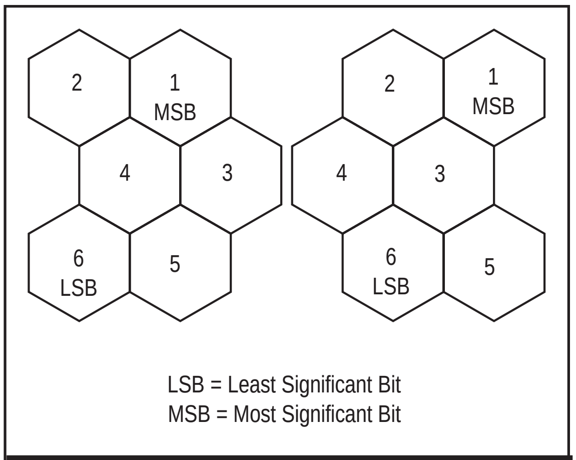


Figure 1: Typical Symbol Character of a MaxiCode Symbol

d. Symbol size:

1. Each MaxiCode symbol is of a fixed size, having 884 hexagonal modules arranged in 33 rows around a central finder pattern. Each row consists of a maximum of 30 modules.

2. Each symbol, including the quiet zone, is of a fixed physical size, nominally 28.14mm wide x 26.91mm high.

3. 864 modules (144 symbol characters) are available for data encodation and error correction. Two modules are unused.

4. Non-data overhead:

   i. 18 modules for orientation per symbol

   ii. equivalent of 90 modules for the finder pattern

e. Maximum data capacity:

1. Alphanumeric characters: 93
2. Numeric characters: 138

f. Error correction:

   50 or 66 codewords per MaxiCode symbol.

g. Code type: matrix

h. Orientation independence: Yes

4.1.2 Summary of Additional Features

The following summary is of additional features which are inherent or optional in MaxiCode:

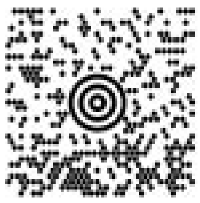
a. Finder Pattern: (Inherent) MaxiCode symbols have a central unique finder pattern, of three concentric dark rings, which is used to locate the MaxiCode symbol within a field of view (see Section 4.2.1.1). The finder pattern and fixed symbol size makes the MaxiCode symbology suitable for high speed scanning applications.

b. Error Correction: (Inherent) MaxiCode symbols have error correction codewords, based on Reed-Solomon error correction algorithms, which can be used not only to detect errors but to correct erroneously decoded or missing codewords (see Section 4.5.1). A user may select one of two error correction levels.

c. Modes: (Inherent) This mechanism allows various structures of the symbol. Seven modes are specified (including 2 obsolete modes, see Section 4.8).

d. Extended Channel Interpretations: (Optional) This mechanism enables characters from other character sets (e.g. Arabic, Cyrillic, Greek, Hebrew) and other data interpretations or industry-specific requirements to be represented.

e. Structured Append: (Optional) This allows files of data to be represented in up to 8 MaxiCode symbols. The original data can be correctly reconstructed regardless of the order in which the symbols are scanned (see Section 4.9).



"THIS IS A 93 CHARACTER CODE SET A MESSAGE THAT FILLS A MODE 4, UNAPPENDED, MAXICODE SYMBOL..."

Figure 2: MaxiCode Symbol (Actual Size)

4.2 Symbol Description

4.2.1 Symbol Structure

Each MaxiCode symbol consists of a central finder pattern surrounded by a square array of offset rows of hexagonal modules. The 33 rows in the symbol alternate between 30 and 29 modules in width. The symbol shall be surrounded on all four sides by a quiet zone border. Figure 2 illustrates a MaxiCode symbol.

4.2.1.1 Finder Pattern

The finder pattern is made up of 3 dark concentric rings and 3 included light areas, centered on the virtual module specified in Section 4.11.4. Figure 3 shows the finder pattern relative to the adjacent module pattern.

4.2.1.2 Orientation Patterns

The orientation information is provided by 6 patterns of 3 modules. The precise location of the Orientation Patterns are given in Figures 3 - 5.

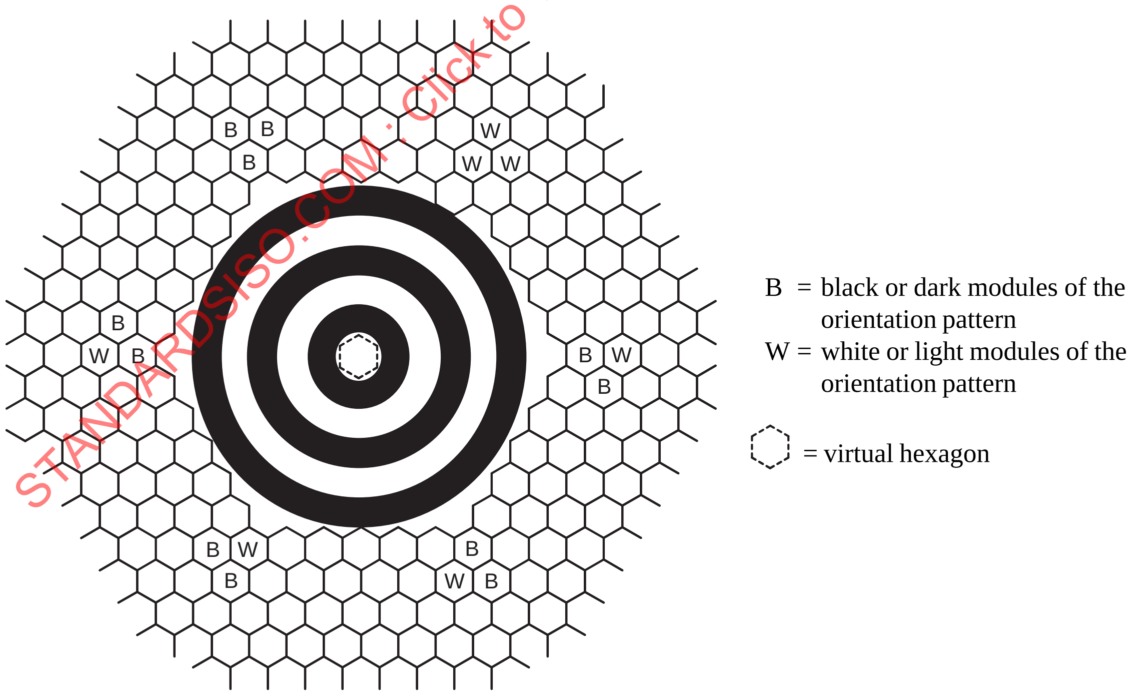

B
B
B
W
W
W
B
W
B
B
W
B
B
W
B
B
W
B
B = black or dark modules of the orientation pattern
W = white or light modules of the orientation pattern
= virtual hexagon

Figure 3: MaxiCode Symbol Construction - Finder Pattern and Orientation Modules

4.2.2 Symbol Character and Module Sequence

A MaxiCode symbol contains 144 symbol characters in the primary and secondary messages. The symbol characters are ordered according to the following rules:

a. Symbol characters in the primary message (1 to 20) are arranged around the finder pattern as shown in Figure 4. Symbol characters in the secondary message (21 to 144) are arranged in a boustrophedonic pattern beginning at the upper left corner proceeding left to right in the first row, right to left in the second row and so forth (See Figure 4).

b. Each hexagonal module is numbered. Figure 5 illustrates the sequence of hexagonal modules within a symbol. Generally, the hexagonal modules of a symbol character are contiguous and are numbered from right to left and top to bottom within the symbol character. In all cases, the lowest numbered module in a symbol character is the Most Significant Bit (See Figure 1). In all cases, module M is the Nth bit of symbol character C (where N=1 is the Most Significant Bit up to N = 6 is the Least Significant Bit) where:

$C = ((M - 1) \text{ div } 6) + 1$

$N = ((M - 1) \text{ mod } 6) + 1$

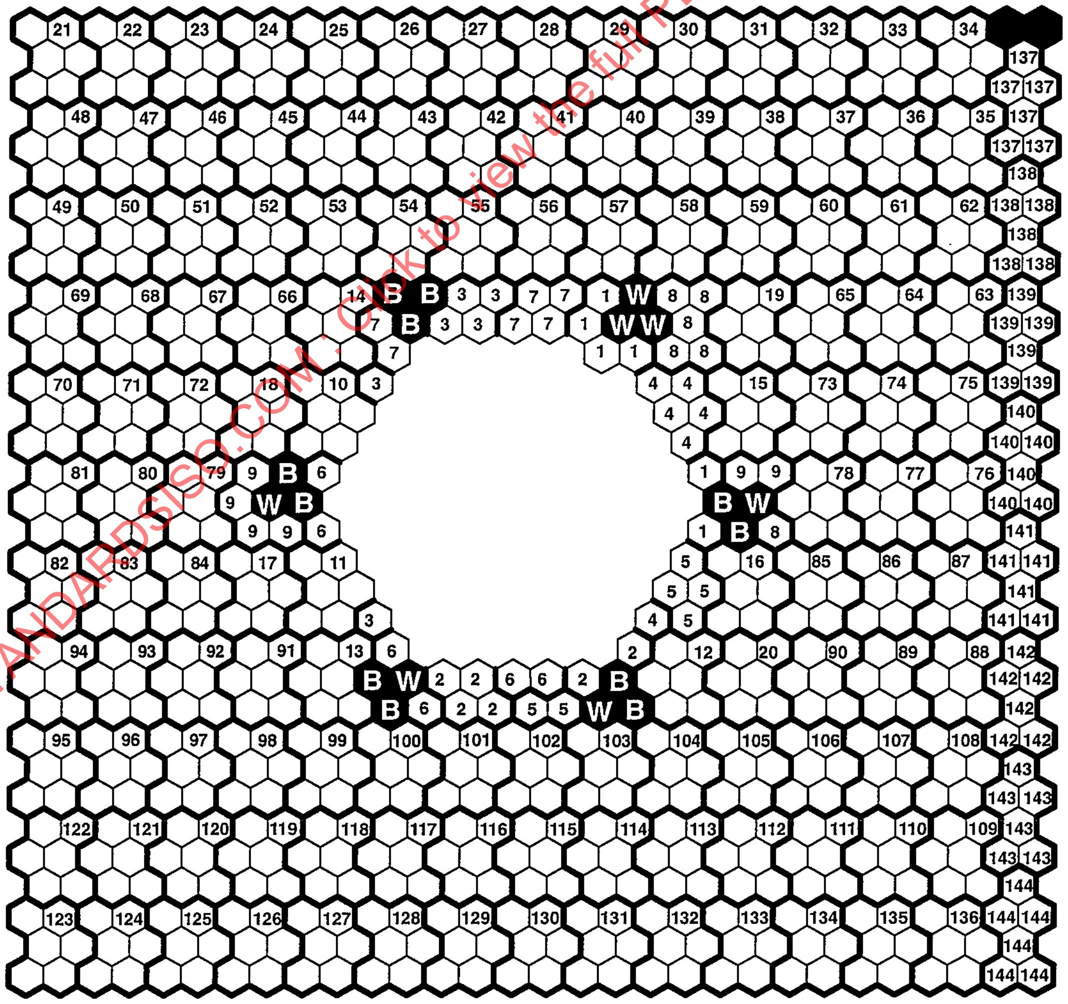


Figure 4: MaxiCode Symbol Character Sequence

c. Modules 1 to 120, i.e. 20 symbol characters, shall contain information from the Primary Message, including data, error correction and mode information. Modules 121 to 864, i.e. 124 symbol characters, shall contain information from the Secondary Message.

The two rightmost modules in the top row are not utilized (See Figure 5). They shall be encoded as dark modules.

4.3 General Encodation Procedures

The following steps are required to convert data into the encoded form represented in a MaxiCode symbol. The following sections of this specification specify the rules and procedures.

1. For transportation applications, determine if a structured carrier message is appropriate; if so, special encoding rules apply to the primary message.

2. Data from a 256 character set may be encoded in MaxiCode. This data needs to be presented in a data stream reading from left to right.

3. Each character of data is translated into a codeword (0-63). Additional codewords are

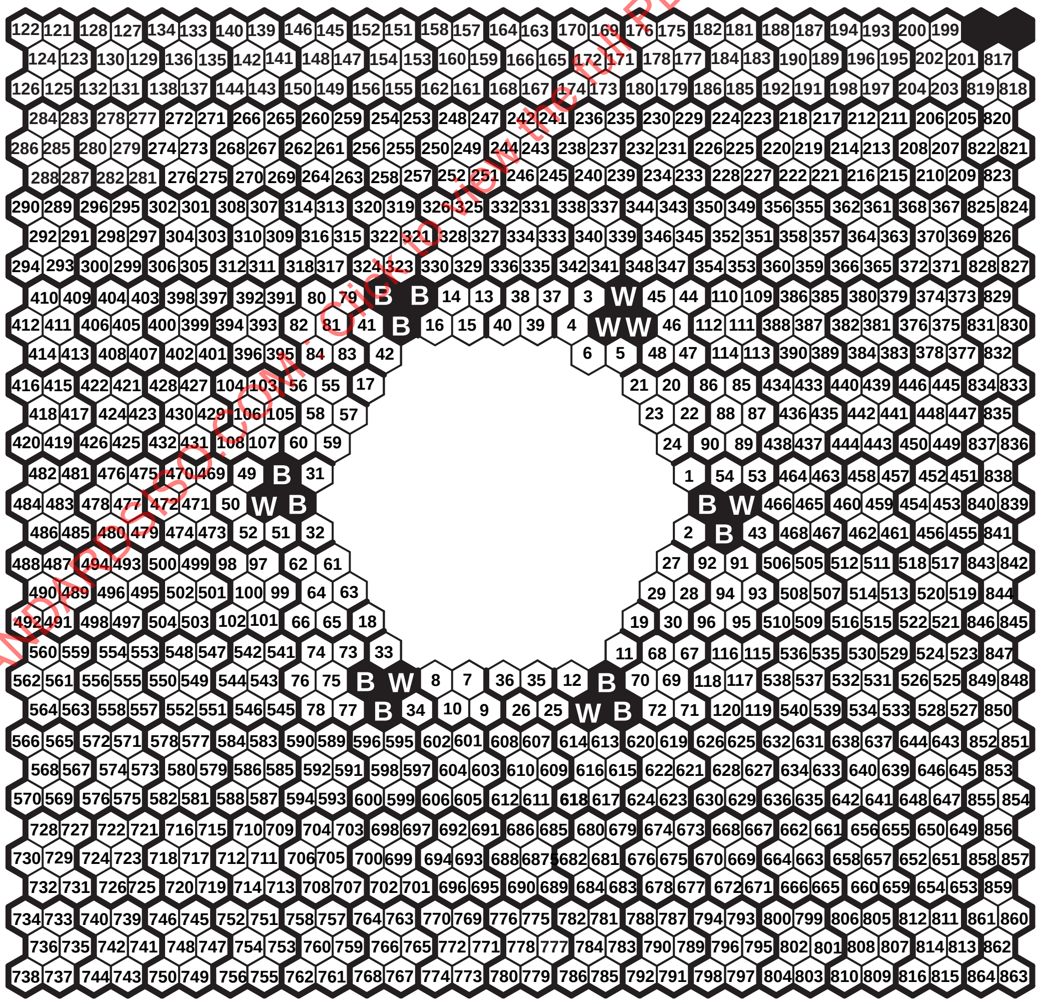


Figure 5: MaxiCode Module Sequence

inserted in the process to switch between subsets of character sets.

4. The user or application selects one of two error correction levels.

5. Pad characters are added as needed to fill out the capacity of the symbol.

6. The codeword stream is subdivided into two messages: Primary and Secondary.

7. The error correction codewords are generated for the Primary Message and for the Secondary Message. The result of this process expands the codeword stream by the number of error correction codewords, either 50 or 66.

8. The codeword stream is converted into two bit streams for the Primary and Secondary Messages.

9. The primary and secondary bit streams are mapped bit by bit to the sequence of hexagonal modules in the MaxiCode symbol (See Figure 5).

4.4 Character Assignments

MaxiCode has a 64 codeword set which is used to encode up to 256 different character values, to provide numeric compaction and to encode particular structured messages (see Section 4.7.3, for a description of Modes 2 and 3). The rules for encoding data are defined in the following sections.

In order to encode all 256 characters, five Code Sets (A to E) are defined. Annexe A shows the encodable character set and the arrangement of characters in the five Code Sets. Characters are grouped into Code Sets according to likely use. Code Set A contains the characters assessed to be most commonly used; in many applications it may not be necessary to switch from this basic character set. To select other data characters, it is necessary to use Latch and Shift characters (see Section 4.4.4.1 to 4.4.4.5).

4.4.1 Codeword Representation

The codewords, or symbol character values, in MaxiCode range from 0 to 63. The binary equivalent of the codeword (i.e. 000000 to 111111) shall be directly represented in six hexagonal modules in the MaxiCode symbol.

The sequence of modules in a codeword is normally represented in a MaxiCode symbol character as illustrated in Figure 1, and the entire data shall follow the module pattern sequence as defined in Figure 5.

4.4.2 Default Character Interpretation

The default character interpretation for character values 0 to 127 shall conform to ANSI X3.4. The default character interpretation for character values 128 to 255 shall conform to ISO 8859-1: Latin Alphabet No. 1. The graphical representation of data characters shown throughout this document complies with the default interpretation. This interpretation can be changed using Extended Channel Interpretation (ECI) escape sequences, see Section 4.6. The default interpretation corresponds to ECI 000003.

4.4.3 Code Sets

4.4.3.1 Code Set A

Code Set A is the default code set at the start of every MaxiCode symbol.

Code Set A contains all the standard uppercase alphabetic characters, the numerals 0 to 9, 15 common punctuation symbols, the space character, and the control characters [CR], [FS], [GS] and [RS] used for data syntax. In addition, it contains 8 symbology control characters.

4.4.3.2 Code Set B

Code Set B contains all the lowercase alphabetic characters and additional punctuation characters. In addition, it includes the control characters [FS], [GS], [RS], and [DEL] used for data syntax and 12 symbology control characters.

4.4.3.3 Code Set C

Code Set C contains multilingual uppercase alphabetic characters and additional punctuation and other graphic characters. It also includes the control characters [FS], [GS] and [RS] used for data syntax and 10 of the characters (values 128 to 137) not assigned a graphic representation in ISO 8859. In addition, it includes 7 symbology control characters.

4.4.3.4 Code Set D
Code Set D contains multilingual lowercase alphabetic characters, and additional punctuation. It also includes the control characters [FS], [GS] and [RS] used for data syntax and 11 characters (values 138 to 148) not assigned a graphical interpretation in ISO 8859. In addition, it includes 7 symbology control characters.

4.4.3.5 Code Set E
Code Set E contains the 31 ASCII control characters, currency indicators and other graphical symbols. It includes 11 characters (values 149 to 159) which do not have a graphical representation in ISO 8859. In addition, it has 9 symbology control characters.

4.4.4 Symbology Control Characters
MaxiCode has 15 symbology control characters which are special non-data characters with no ASCII character equivalents. These characters are used to instruct the decoder to perform certain functions or to send specific data to the host computer as described in Section 4.4.4.1 to 4.4.4.8. Table 1 defines the complete list of assigned symbology control characters. Annexe F provides guidance on optimum use of Latch, Shift, and Lock-In characters.

4.4.4.1 Latch Characters
A Latch Character may be used to switch from one Code Set to another Code Set. All codewords which follow a Latch Character shall be interpreted according to the new Code Set. The Code Set remains in effect until another Latch Character or shift is encountered.
Latch Characters are in all code sets, but are only used to switch to Code Sets A or B.

4.4.4.2 Shift Characters
A Shift Character is used to switch from one Code Set to another Code Set for the single character following the Shift Character. Subsequent character encodation shall revert to the Code Set defined prior to the Shift Character.
Shift Characters are in all code sets. When in Code Sets A or B it is possible to shift to all others.

Function Name and Purpose	Short Name	Codeword Value in Code Set					Refer to Section
		A	B	C	D	E	
Latch: to switch to and remain in a new code set	Latch A		63	58	58	58	4.4.4.1
	Latch B	63		63	63	63	
Shift: to switch to a new code set for one character and to return	Shift A		59				4.4.4.2
	Shift B	59					
	Shift C	60	60		60	60	
	Shift D	61	61	61		61	
	Shift E	62	62	62	62		
Lock-In: to extend a shift to behave as a latch and remain in a new Code Set	Lock-In C			60			4.4.4.3
	Lock-In D			61			
	Lock-In E					62	
Double Shift: to shift for two characters	2 Shift A		56				4.4.4.4
Triple Shift: to shift for three characters	3 Shift A		57				4.4.4.5
Numeric Shift: to compact numeric strings efficiently	NS	31	31	31	31	31	4.4.4.6
Extended Channel Interpretation: to switch to a new Extended Channel Interpretation	ECI	27	27	27	27	27	4.4.4.7
Pad: to fill out a symbol and to signalPad structured append	33	33 55	58		284.4.4.8 29		

Table 1: MaxiCode Symbology Control Characters

4.4.4.3 *Lock-In Character*

A Lock-In Character, following a Shift Character for the same Code Set has the effect of behaving as a Latch Character. The Code Set remains in effect until a Latch Character is invoked.

4.4.4.4 *Double Shift Characters*

A Double Shift Character (referred to as [2 Shift A]) is used to switch from Code Set B to Code Set A for the next two characters following the [2 Shift A] Character. Subsequent characters shall revert to Code Set B.

4.4.4.5 *Triple Shift Character*

A Triple Shift Character [3 Shift A] is used to switch from Code Set B to Code Set A for the next three characters following the [3 Shift A] Character. Subsequent characters shall revert to Code Set B.

4.4.4.6 *Numeric Shift*

Numeric Shift allows 9 digit strings to be encoded into 6 codewords. A Numeric Shift Character [NS] indicates that the next five codewords, equivalent to 30 bits, encodes 9 numeric digits in binary format. Subsequent character encodation shall revert to the Code Set defined prior to the Numeric Shift Character. For longer numeric strings it is possible to mix numeric compaction using Numeric Shift(s) and conventional encodation. Annexe F provides more detailed advice about Numeric Shift for any length of digit string.

4.4.4.7 *Extended Channel Interpretation Character*

An Extended Channel Interpretation (ECI) character is used to change from the default interpretation used to encode data. The Extended Channel Interpretation protocol is common across a number of symbologies and is defined more fully in Section 4.6.

The ECI character shall be followed by one, two, three, or four codewords which identify the ECI being invoked. The new ECI remains in place until the end of the encoded data, or until another ECI character is used to invoke another interpretation.

4.4.4.8 *Pad Character*

The Pad character in the first position is used for structured append (see Section 4.9); otherwise it is used to fill out remaining data capacity of the symbol.

4.5 *User Considerations for Encoding Data in a MaxiCode Symbol*

A MaxiCode symbol has a fixed number of modules and codewords. The 144 codewords can be used to encode the mode, data, symbology control functions, and error correction. It is also possible to use structured append to combine up to eight MaxiCode symbols. A number of symbology parameters may be pre-determined by the application. These include the level of error correction and the mode. Other parameters are more associated with the data, including the use of particular character sets, the need for data to conform to particular application standards or message syntax (e.g. EDIFACT), and the degree of switching between Code Sets. MaxiCode encodation should be done automatically, but some general guidelines on the capacity of a symbol to encode data are given in Section 4.5.5 and Annexe G.

4.5.1 *User Selection of Error Correction Level*

MaxiCode symbols offer two levels of error correction (which are specified in Section 4.10). In an application it is sufficient to understand that these two levels require different numbers of codewords, offer different levels of error recovery and are selected by the choice of mode. The basic features are set out in Table 2.

Feature	Error Correction Level	
	Standard	**Enhanced**
Total number of codewords	144	144
Those available for data encodation	93	77
Codeword used for specifying mode	1	1
Those used for error correction	50	66
Number of erroneously decoded codewords which can be corrected	22	30

Table 2: Features of Error Correction

4.5.2 *User Selection of Mode*

MaxiCode symbols offer five modes of encodation (which are specified in Section 4.8). Generally the modes are used to define the format of the message and the level of error correction.

4.5.3 User Selection of Extended Channel Interpretation

The use of an alternative Extended Channel Interpretation to identify a particular code page or more specific data interpretation requires additional codewords to invoke the feature. The use of the Extended Channel Interpretation protocol (see Section 4.6) provides the capability to encode data from alphabets other than the Latin alphabet (ISO 8859-1 Latin Alphabet No. 1) supported by the default interpretation.

4.5.4 User Selection of Structured Append

There may be application requirements where the number of MaxiCode symbols has to be restricted to one symbol, or a fixed or maximum number of symbols, or has the freedom to extend to the maximum of eight linked symbols. See Section 4.9 for the specification of Structured Append. The limiting number of MaxiCode symbols may be specified for an application. Two codewords are required to define a MaxiCode symbol as part of a Structured Append.

4.5.5 User Assessment of Encodation Capacity

MaxiCode symbols have a limited data capacity as indicated in Table 2. See Annexe G for User Assessment of Encodation Capacity.

4.6 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. Four broad types of interpretations are supported in MaxiCode:

a. international character sets (or code pages)
b. general purpose interpretations such as encryption and compaction
c. user defined interpretations for closed systems
d. control information for structured append in unbuffered mode.

The Extended Channel Interpretation protocol is fully specified in the *Extended Channel Interpretation (ECI) Assignments* document. The protocol provides a consistent method to specify particular interpretations on byte values before printing and after decoding.

The Extended Channel Interpretation is identified by a 6-digit number which is encoded in the MaxiCode symbol by the ECI character followed by one to four codewords.

A specific Extended Channel Interpretation may be invoked anywhere in the encoded message, except where special rules apply in Mode 2 and 3 symbols (see Section 4.6.1).

The Extended Channel Interpretation can only be used with readers enabled to transmit symbology identifiers. Readers that are not enabled to transmit the symbology identifer shall not transmit the data from any symbol containing an ECI. An exception can be made if the ECI(s) can be handled entirely within the reader.

4.6.1 ECI and Modes 2 and 3

Modes 2 and 3 are used to encode a Structured Carrier Message in the Primary Message (see Section 4.8.3).

In modes 2 and 3, ECIs may only be invoked within the Secondary Message.

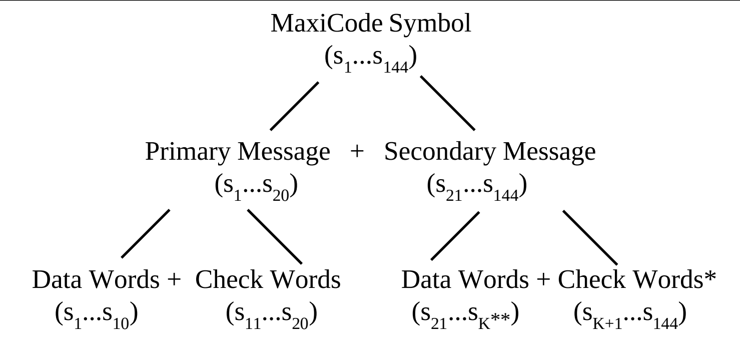
4.6.2 Encodation Modes and ECIs

The encodation mode used is determined strictly by the 8-bit data values being encoded and does not depend on the Extended Channel Interpretation in force. For example, a sequence of values in the range 48 to 57 (decimal) would be most efficiently

ECI Assignment Value	Codeword Sequence	Codeword Values
000000 to 000031	C_0C_1	[27] [0bbbbb]
000000 to 001023	$C_0C_1C_2$	[27] [10bbbb] [bbbbbb]
000000 to 032767	$C_0C_1C_2C_3$	[27] [110bbb] [bbbbbb] [bbbbbb]
000000 to 999999	$C_0C_1C_2C_3C_4$	[27] [1110bb] [bbbbbb] [bbbbbb] [bbbbbb]

Where: b...b is the binary value of the ECI assignment number

Table 3: Encoding ECI Assignment Numbers


MaxiCode Symbol
$(s_1 \ldots s_{144})$
Primary Message + Secondary Message
$(s_1 \ldots s_{20})$
$(s_{21} \ldots s_{144})$
Data Words + Check Words
$(s_1 \ldots s_{10})$
$(s_{11} \ldots s_{20})$
Data Words + Check Words*
$(s_{21} \ldots s_{K^{**}})$
$(s_{K+1} \ldots s_{144})$

* Within the Secondary Message, check words are computed over odd and even subsets.
** "K" = 104 for Standard Error Correction, 88 for Extended Error Correction

Figure 6: Symbol Structure and Symbol Character Designations

encoded in numeric mode even if the sequence is not to be interpreted as numbers.

4.6.3 Encoding ECIs in MaxiCode

The ECI assignment is invoked using codeword 27, the ECI character. One to four additional codewords are used to encode the ECI Assignment number. The encodation rules are defined in Table 3.

Note: On decoding, the binary pattern of the C1 codeword (i.e. the codeword following codeword 27) determines the length of the ECI sequence. The number of one bits before the first zero bit determines the number of additional codewords used to define the ECI assignment number. The bit sequence after the leading zero bit is the ECI number in binary notation. The lower numbered ECI assignments may be encoded in multiple ways, but the shortest way is preferred.

4.6.4 ECIs and Structured Append

ECIs may be encoded anywhere in the message encoded in a single or Structured Append (see Section 4.9) set of MaxiCode symbols, but cannot be within the primary message for modes 2 and 3. Any ECI invoked shall apply until the end of the encoded data, or until another ECI is encountered. Thus the interpretation of the ECI may straddle two or more symbols.

4.6.5 Post-Decode Protocol

The protocol for transmitting ECI data shall be as defined in Section 4.15.2. When using ECIs, symbology identifiers (see Section 4.15.3) shall be fully implemented and the appropriate symbology identifier transmitted as a preamble.

4.7 Message Structure

MaxiCode symbols are divided into a primary and a secondary message, each of which contains data and error correction codewords. These messages are structured as shown in Figure 6.

4.7.1 Primary Message

The Primary Message includes 20 symbol characters, designated as 1 (s_1) through 20 (s_{20}) in Figure 4 and represented by modules 1 to 120 in Figure 5. Of these symbol characters:

10 (s_1 through s_{10}) are used to encode data (including the symbol mode).
10 (s_{11} through s_{20}) are used for error correction.

The four LSBs of symbol character s_1, modules 3 through 6, encode the mode (see Section 4.8). The other two modules in s_1 shall be set to zero unless they are part of a Structured Carrier Message (see Annexe B). Enhanced Error Correction (see Section 4.10.1) is always applied to the Primary Message.

In Modes 2 and 3, the primary message is a strictly formatted Structured Carrier Message as specified in Annexe B. In Modes 4 through 6, the primary message contains the mode symbol character followed by 9 symbol characters that begin the encoding of the symbol's data message.

4.7.2 Secondary Message

The Secondary Message includes 124 symbol characters, designated as 21 (s_{21}) through 144 (s_{144}) in Figure 4 and represented by modules 121 to 864 in Figure 5. One of two levels of error correction shall apply to the Secondary Message, either Enhanced Error Correction (EEC) or Standard Error Correction (SEC) (see Section 4.10.2).

If EEC is used:

68 symbol characters (s_{21} through s_{88}) are used to encode data, and

56 symbol characters (s_{89} through s_{144}) are used for error correction.

If SEC is used:

84 symbol characters (s_{21} through s_{104}) are used to encode data, and

40 symbol characters (s_{105} through s_{144}) are used for error correction.

In both cases, error correction for the Secondary Message is implemented in two interleaved Reed-Solomon subsets, designated the “odd” and the “even” subsets. The odd-numbered symbol characters s_{21}, s_{23}, s_{25},..., s_{143} comprise in sequence the odd subset, while the even-numbered symbol characters s_{22}, s_{24}, s_{26},..., s_{144} comprise in sequence the even subset. Within each subset the error correction codewords are derived from, and ultimately provide error correction for, the data codewords in that same subset.

The data symbol characters in the Secondary Message (preceeded by 9 symbol characters from the Primary Message for modes 4, 5, and 6) shall encode a data message using the code sets and control characters presented in Sections 4.4.3 and 4.4.4.

4.7.3 Structuring the Data

Annexe H presents a specific MaxiCode encoding example which illustrates many steps in the procedure described here.

Initially, the desired data message string m_1 through m_i shall be encoded into a sequence of 6-bit codewords c_1 through c_j using the code sets and symbology control characters (see Sections 4.4.3 and 4.4.4) as needed for the application. Special encoding rules shall be followed for destination and class of service data within a Structured Carrier Message with a Numeric (Mode 2) or Alphanumeric (Mode 3) postal code (see Section 4.8.3 and Annexe B). Annexe F provides guidance for finding the most efficient use of the symbology control characters.

In Modes 2 and 3, the data message codewords are placed exclusively into the secondary message from symbol character 21 upward. In Modes 4 through 6, the first 9 message codewords are placed into the primary message in symbol characters 2 through 10, and then the remainder are placed into the secondary message from symbol character 21 upward. When the message codewords do not exactly fill out all the message regions of a symbol, then codewords representing the Pad character are appended as needed. The structure of a MaxiCode symbol is such that a symbology control character and the subsequent message character(s) that it acts upon can appear in different messages and often appear in different error correction subsets, but this has no effect on the message encoding.

Once the data codewords are subdivided into messages and then into error correction subsets, error correction shall be applied as indicated in Section 4.10. Finally, both data and error correction codewords are inserted into the graphical symbol following the placements shown in Figures 4 and 5.

4.8 Modes

MaxiCode has modes which are used to define the structuring of the data and error correction within a symbol. The mode shall be encoded as part of the Primary Message (see Section 4.7.1).

4.8.1 Mode 0: Obsolete

Mode 0 is obsolete. It is superseded by Modes 2 and 3. See the document, *Guideline on Mode 0 for MaxiCode*.

4.8.2 Mode 1: Obsolete

Mode 1 is obsolete. It is superseded by Mode 4.

4.8.3 Modes 2 and 3: Structured Carrier Message

Modes 2 and 3 are designed for use in the transport industry. They encode the destination address and the class of service as defined by the carrier. The structure of the message is defined in Annexe B. The first 120 bits are used to encode the Structured Carrier Message with Enhanced Error Correction (EEC). The rest of the symbol is free for other uses and employs Standard Error Correction (SEC).

4.8.4 Mode 4: Standard Symbol
Mode 4 indicates that the symbol employs EEC in the Primary Message and SEC for the Secondary Message. This mode provides 93 codewords for data encodation.

4.8.5 Mode 5: Full EEC
Mode 5 indicates that the symbol employs EEC for both the Primary and Secondary Message. This mode provides 77 codewords for data encodation.

4.8.6 Mode 6: Reader Programming
Mode 6 indicates that the symbol encodes a message used to program the reader system and that SEC is used for the Secondary Message. When a Mode 6 symbol is read no data shall be transmitted.

4.8.7 Mode Indicators
The Mode shall be encoded in the first symbol character, using only modules 3 to 6. Modes shall be encoded as defined in Table 4.

Mode	Description	Module Numbers 3456
0	Obsolete	0000
1	Obsolete	0001
2	Structured Carrier Message - Numeric Postal Code	0010
3	Structured Carrier Message - Alphanumeric Postal Code	0011
4	Standard Symbol, SEC	0100
5	Full EEC Symbol	0101
6	Reader Program, SEC	0110

Note: All modes and bit patterns not defined in this table are reserved for future use.

Table 4: MaxiCode Modes

4.9 Structured Append
4.9.1 Basic Principles
Up to eight MaxiCode symbols may be appended in a structured format.

If a symbol is part of a Structured Append this shall be indicated by a sequence of two symbol characters in particular positions in the symbol, depending on the Mode.

The Structured Append indicator sequence consists of:

a. The first data symbol character which shall be a Pad (symbol character value 33).

b. The second data symbol character which shall indicate the position of the symbol within the set of MaxiCode symbols in the Structured Append format; i.e. in the format of m of n symbols.

The first three bits of the second codeword identify the position of the particular symbol as the binary value of (m-1). The last three bits identify the total number of symbols, as the binary value of (n-1), to be concatenated in the Structured Append format. The 3-bit patterns shall conform to those defined in Table 5.

Symbol Position	Bits 123	Total Number of Symbols	Bits 456
1	000	N/A	N/A
2	001	2	001
3	010	3	010
4	011	4	011
5	100	5	100
6	101	6	101
7	110	7	110
8	111	8	111

Table 5: Structured Append Symbol Position

Example:

To indicate the third symbol of a set of seven, this shall be encoded thus:

Third position:	010
Total 7 symbols:	110
Bit pattern:	010110
Codeword:	22

4.9.2 Structured Append and Modes 2 and 3
For Mode 2 or Mode 3, the structured append

indicator sequence shall be placed in the first two data symbol characters in the Secondary Message, i.e. symbol characters s_{21} and s_{22}. Symbol character s_{23} starts normal data encodation in Code Set A. When Mode 2 or Mode 3 is used in Structured Append all of these symbols shall be in the same mode, and the primary message shall be repeated in each symbol.

4.9.3 Structured Append in Modes 4 to 6

For Modes 4 to 6, the structured append indicator sequence shall be placed in the first and second data symbol characters in the Primary Message, i.e. symbol characters s_2 and s_3. Symbol character s_4 starts normal data encodation in Code Set A.

4.9.4 Buffered and Unbuffered Operation

The message within a structured append sequence can be buffered in the reader in its entirety and transmitted after all of the symbols have been read. Alternatively, the reader may transmit the decoded data in each symbol as it is read. In this unbuffered operation, the ECI protocol for structured append defines a control block that shall be prefixed to the beginning of each transmission.

4.10 Error Detection and Correction

MaxiCode symbols employ Reed-Solomon error correction at one of two levels:

Standard Error Correction (SEC)
Enhanced Error Correction (EEC)

The subdivision of the MaxiCode symbol into Primary and Secondary Messages and the subdivision of the Secondary Message into two interleaved subsets enables error correction to be applied. Error correction is applied separately to each of the three subdivisions.

For a given sequence of data codewords (i.e. the Primary Message or a subset in the Secondary Message) error correction codewords shall be computed using the Reed Solomon Error Control Code algorithm.

The polynomial arithmetic for MaxiCode is calculated using bit-wise modulo 2 arithmetic and word-wise modulo 1000011 (decimal 67) arithmetic. This is arithmetic based on the Galois Field of 2^6 with 1000011 representing the field's prime modulus polynomial: x^6+x+1.

4.10.1 Enhanced Error Correction (EEC) in the Primary Message

Enhanced Error Correction (EEC) shall be used in the Primary Message, requiring 10 error correction codewords.

The generator polynomial g(X) for EEC in the Primary Message is:

$$g(x) = (x - 2^1)(x - 2^2) \ldots (x - 2^{10}) =$$

$$x^{10} + 31x^9 + 28x^8 + 39x^7 + 42x^6 + 57x^5 + 2x^4 + 3x^3 + 49x^2 + 44x + 46$$

4.10.2 Error Correction in the Secondary Message

The Secondary Message employs one of the following error correction levels:

a. Enhanced Error Correction (EEC), requiring 28 error correction codewords per subset.

b. Standard Error Correction (SEC), requiring 20 error correction codewords per subset.

The generator polynomial g(x) for EEC in the Secondary Message is:

$$g(x) = (x - 2^1)(x - 2^2) \ldots (x - 2^{28}) =$$

$$x^{28} + 22x^{27} + 45x^{26} + 53x^{25} + 10x^{24} + 41x^{23} + 55x^{22} + 35x^{21} + 10x^{20} + 22x^{19} + 29x^{18} + 23x^{17} + 13x^{16} + 61x^{15} + 45x^{14} + 34x^{13} + 55x^{12} + 40x^{11} + 37x^{10} + 46x^9 + 49x^8 + 34x^7 + 41x^6 + 9x^5 + 43x^4 + 7x^3 + 20x^2 + 11x + 28$$

The generator polynomial g(x) for SEC in the Secondary Message is:

$$g(x) = (x - 2^1)(x - 2^2) \ldots (x - 2^{20}) =$$

$$x^{20} + 23x^{19} + 44x^{18} + 11x^{17} + 33x^{16} + 27x^{15} + 8x^{14} + 22x^{13} + 37x^{12} + 57x^{11} + 36x^{10} + 15x^9 + 48x^8 + 22x^7 + 17x^6 + 38x^5 + 33x^4 + 31x^3 + 19x^2 + 23x + 59$$

4.10.3 Generating the Error Correction Codewords

The error correction codewords are the remainder after dividing the n data codewords by the generator polynomial g(x) of degree k. In MaxiCode this is done separately for each of the error correction subdivisions.

The data codewords are the coefficients of the terms of a polynomial with the coefficient of the highest term being the first codeword and the coefficient of the lowest term being the last codeword before the first error correction codeword. The highest order coefficient of the remainder is the first error correction codeword and the zero power coefficient of the remainder is the last error correction codeword out of the register.

The error correction codewords can be generated by using the division circuit as shown in Figure 7. The registers b_0 through b_{k-1} are initialized as zeroes. There are two phases to generate the encoding. In the first phase, with the switch in the down position the symbol data is passed both to the output and the circuit. The first phase is completed after n clock pulses. In the second phase (n+1 ... n+k clock pulses), with the switch in the up position, the error correction codewords are generated by flushing the registers in order while keeping the data input at zero. The codewords output from the shift register are in the order that they are to be placed in the symbol. The secondary message is interleaved therefore the codewords will not be placed in consecutive symbol characters.

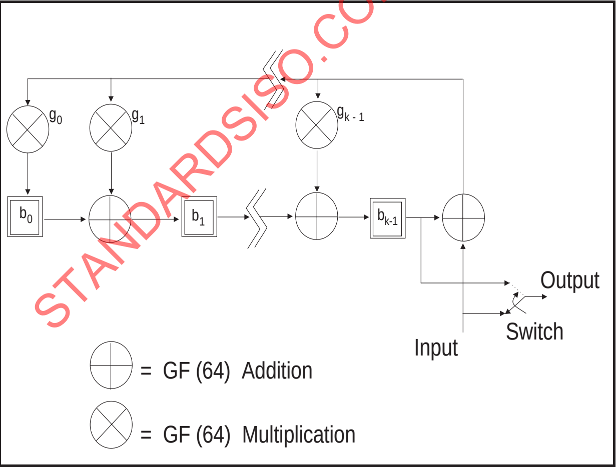


Figure 7: Error Correction Codeword Encoding Circuit

4.10.4 Error Correction Capacity

The error correction codewords can correct two types of erroneous codewords, erasures (erroneous codewords at known locations) and errors (erroneous codewords at unknown locations). An erasure is an unscanned or undecodable symbol character. An error is a misdecoded symbol character. The number of erasures and errors correctable is given by the following formula:

$e + 2t \le d - 2$

where:

e	=	Number of erasures
t	=	Number of errors
d	=	Number of error correction codewords

However, if most of the error correction capacity is used to correct erasures, the possibility of undetected errors is increased. Whenever there are fewer than ten errors and the number of erasures is more than half the number of error correction codewords, then:

$e + 2t \le d - 4$

Note that the equation states that four error correction characters may need to be reserved if there are a large number of erasures. Otherwise, the symbol cannot be decoded without risking a misdecode.

4.11 Dimensions

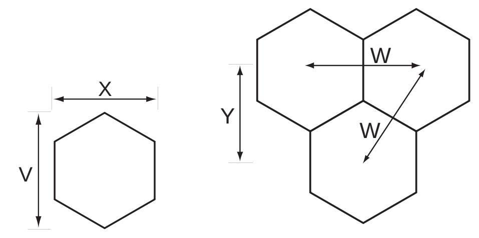
4.11.1 Symbol Dimensions

L is the width of the symbol measured from the center of the left-most module to the center of the rightmost module in the top row of the symbol. The acceptable range of L is 24.00 mm to 27.00 mm.

H is the height of the symbol measured from the center of the top row to the center of the bottom row. The nominal value of H depends on L: H = 0.9556L.

4.11.2 Hexagonal Module Dimensions

The hexagonal modules in a MaxiCode symbol are nested in offset rows, containing 29 or 30 modules. Four measurements define the size and placement of the modules in relation to each other (see Figure 8).



V is the vertical height of a module
W is the center to center distance between adjacent modules
X is the horizontal width of a module
Y is the vertical distance from the center line of modules in one row to the center line of modules in an adjacent row

Figure 8: MaxiCode Module Dimensions

In order to prevent problems caused by the buildup of tolerances within the symbol the module dimensions shall be based on the L dimension of the symbol. The achievable L, given the printing technology and the range of values specified in Section 4.11.3, shall be determined (see Annexe J). From that, the nominal dimensions for V, W, X and Y are computed. The relationships and dimensions for a module are defined in Table 6.

4.11.3 Dark Hexagon Dimensions and Tolerances
The dimensions given in Table 7 are for the hexagon module in the grid as illustrated in Figure 5. To enhance decoding, dark hexagons should not abut contiguously but instead should be distinct from one another. Actual dimensions of dark hexagons should be as defined in Table 7.
Practical printing guidance including dark hexagon font development is included in Annexe J.

	Nominal	Tolerance
width	(X - 0.12 mm)	±0.12 mm
height	(V - 0.12 mm)	±0.12 mm

Table 6: Dark Module Size

Dimension	Relationships to Other Dimensions	Dimension based on L = 25.50 mm (mm)
W	W = L/29	0.88
V	V = (2 / √3)W	1.02
X	X = W	0.88
Y	Y = (1.5 / √3)W	0.76
H	H = 32Y	24.37

Table 7: MaxiCode Module Dimensions Based on L

4.11.4 Finder Pattern Dimensions
The finder pattern shall be as defined in Section 4.2.1.1 and illustrated in Figure 3. The dimensions of the finder pattern shall be as defined below (and illustrated in Figure 9).

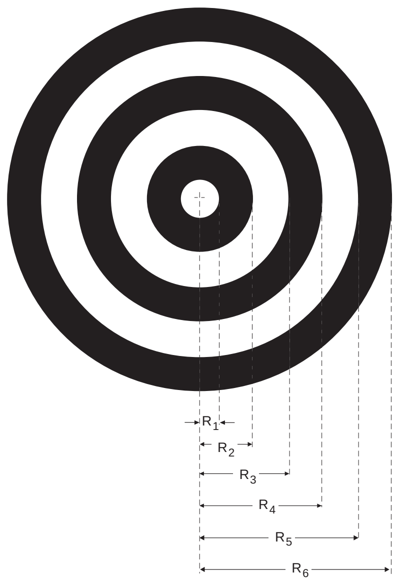


Figure 9: MaxiCode Finder Pattern Dimensions

The finder pattern shall be concentric with the center of the module location which is 16Y above the center of modules in the bottom-most row of the symbol, and 14W to the right from the center of the left-most module of the row.
The thickness of both light and dark rings is

generally uniform. The finder pattern is defined by the radii of the transitions from light to dark and dark to light, R_1 to R_6 where R_i is based on a smooth curve averaging pixel roughness. The dimensions of these radii are listed in Table 8.

Dimension	Nominal Dimension (mm)
R_1	0.51
R_2	1.18
R_3	1.86
R_4	2.53
R_5	3.20
R_6	3.87

Table 8: Finder Pattern Dimensions

An internal clear area surrounding the finder pattern is defined by the arrangement of the modules. This area should be kept free of all extraneous marks.

See Annexe J for practical printing guidance.

4.11.5 Quiet Zones
MaxiCode symbols require the following quiet zones as measured from the outside edges:

1W on the left and right sides
1Y on the top and bottom

The quiet zone dimensions shall apply between the MaxiCode symbol and any adjacent printing.

4.11.6 Overall Symbol Size
A MaxiCode symbol, including quiet zones, is 32X wide by (34Y + V) high. Each MaxiCode symbol shall be in the range of 26.48mm wide by 25.32mm high to 29.79mm wide by 28.49mm high.

4.11.7 Practical Printing Guidance
Printing systems may not be capable of achieving the nominal dimensions shown. Practical printing guidance is provided in Annexe J.

4.12 User Guidelines
4.12.1 Human Readable Interpretation
Because MaxiCode symbols are capable of encoding many characters, a human readable interpretation of the data characters may not be practical. As an alternative, descriptive text rather than the encoded text may accompany the symbol. The character size and font are not specified and the message may be printed anywhere in the area surrounding the symbol. The human readable interpretation should not interfere with the symbol itself nor the quiet zones.

4.12.2 Autodiscrimination Capability
MaxiCode can be used in an autodiscrimination environment with a number of other symbologies (see Annexe K).

4.13 Symbol Quality
MaxiCode symbols shall be assessed for quality using the 2D matrix bar code symbol print quality guidelines presented in Annexe C, as augmented and modified below.

4.13.1 Obtaining the Test Image
A grey-scale image of the symbol being tested shall be obtained with at least 16 dpmm (400 dpi) resolution. It is recommended that a desktop flat-bed page scanner be used to obtain the image, taking care to set it for a flat linear 8-bit grey-scale response. A precision video camera-based setup as described in Annexe C.1 may alternately be used, but feed-through page scanners and hand-driven scanners do not assure the spatial accuracy needed for symbol quality testing.

4.13.2 Symbol Quality Parameters
4.13.2.1 Decode
The reference decode algorithm presented in Section 4.14 shall be applied to the test image. If it results in a successful decode of the entire data message, then Decode passes with a grade of “A” (4.0), otherwise it fails with a grade of “F” (0.0).

4.13.2.2 Symbol Contrast
As part of the reference decode, a 4 dpmm (100 dpi) grey-scale image is derived from the test image. Determine the dark and light reflectance values and symbol contrast grade as detailed in Annexe C.2.2.

4.13.2.3 “Print” Growth
“Print” Growth shall be assessed by checking the

area filled by isolated dark modules relative to the area of the ideal hexagonal module in the symbol. This measurement is performed by counting the contiguous dark or light bits in a 16 dpmm (400 dpi) image which has been binarized using the threshold determined in Step 3 of the reference decode algorithm.

If the binarized image has a pixel resolution of R dpmm, and the reference decode reveals an average center-to-center module spacing of X, then the allocated area for each hexagonal module in the grid is $HEX_{ALLOC} = (X * R)^2 * \sqrt{3}/2$ pixels.

From the reference decode, the dark hexagonal modules which are surrounded on all six sides by light modules can be determined and located in the image. The number of contiguous dark pixels in the region of each such isolated module shall be counted, then those counts averaged to obtain the average dark module area HEX_{DARK}. Normalizing HEX_{DARK} to the allocated area HEX_{ALLOC} creates a dimensionless area measure of the dark modules $A = HEX_{DARK} / HEX_{ALLOC}$.

The optimal dark hexagon size is 75% of the allotted area and can be achieved by undercutting the print pattern (see Annexe J). As the size of the dark hexagons deviates from optimal, reading performance decreases. Thus, nominal printing of MaxiCode should produce isolated dark modules with $A_{NOM} = 0.75$, while the maximum allowed area is $A_{MAX} = 0.95$ and the minimum is $A_{MIN} = 0.55$. Determine:

$$D_{NOM} = (A_{NOM})^{½} = 0.8660$$
$$D_{max} = (A_{MAX})^{½} = 0.9747$$
$$D_{min} = (A_{MIN})^{½} = 0.7416$$
$$D = (A)^{½}$$

Apply the method described in Annexe C.2.3 to obtain the grade for Print Growth.

4.13.2.4 Axial and Local Nonuniformity

Step 11 in the reference decode algorithm creates by inverse Fast Fourier transform an image of the module centers. The precise locations of these sampling points relative to those of its neighbors are the basis for assessing axial and local nonuniformity.

The fixed scaling of MaxiCode symbols, combined with the way that the sampling grid is derived, both alters and expands the useful symbol quality assessments which can be made about axial nonuniformity. A "global" nonuniformity is defined for MaxiCode which simply assures that the symbol scaling along each of the three polygonal axes falls within absolute limits. In addition, a "local" axial nonuniformity is defined which guards against grid distortions along any axis which would imperil a symbol's readability.

Along each of MaxiCode's three main axis, the spacing between each pair of adjacent sampling points is measured. Average spacing X_{AVG}, maximum spacing X_{MAX}, and minimum X_{MIN} are determined and used for grading two independent quality parameters as follows:

4.13.2.4.1 Axial Nonuniformity

MaxiCode is a fixed size symbology with a nominal center-to-center data module spacing of 0.88 mm. So that the overall symbol size stays within the dimensional limits of Section 4.11.1, each of the three average grid spacings shall be graded as:

A (4.0)	if $0.820 \text{ mm} \le X_{AVG} \le 0.940 \text{ mm}$
F (0.0)	otherwise, i.e. $X_{AVG} < 0.820$ mm or $X_{AVG} > 0.940$

The symbol's Axial Nonuniformity grade is the lowest obtained for any of its three axes.

4.13.2.4.2 Local Nonuniformity

The extent to which the maximum and minimum spacings of the module sampling points along each axis differ from their average indicate the severity of variations or discontinuities in a symbol's hexagonal grid. For each axis independently then, the Local Nonuniformity shall be graded:

A(4.0) if $X_{MAX} - 0.10 \text{ mm} \le X_{AVG} \le X_{MIN} + 0.10 \text{ mm}$
B(3.0) if $X_{MAX} - 0.15 \text{ mm} \le X_{AVG} \le X_{MIN} + 0.15 \text{ mm}$
C(2.0) if $X_{MAX} - 0.20 \text{ mm} \le X_{AVG} \le X_{MIN} + 0.20 \text{ mm}$
D(1.0) if $X_{MAX} - 0.25 \text{ mm} \le X_{AVG} \le X_{MIN} + 0.25 \text{ mm}$
F(0.0) if $X_{MAX} - 0.25 \text{ mm} > X_{AVG} > X_{MIN} + 0.25 \text{ mm}$

The symbol's Local Nonuniformity grade is the lowest obtained for any of its three axes.

4.13.2.5 Unused Error Correction

For error correction, a MaxiCode symbol is separated into the primary message subset and two interleaved subsets in the secondary message. Each of these three Reed-Solomon subsets shall be graded

Grade	Reference Decode	Symbol Contrast	"Print" Growth	Axial Nonuniformity	Local Nonuniformity	Unused Error Correction
A(4.0)	Passes	$SC \geq 70\%$	$-0.50 \leq D' \leq 0.50$	$0.820 \leq X_{AVG} \leq 0.940$	X_{MAX} - 0.10 mm $\leq X_{AVG} \leq X_{MIN}$ + 0.10 mm	$UEC \geq 0.62$
B(3.0)		$SC \geq 55\%$	$-0.70 \leq D' \leq 0.70$		X_{MAX} - 0.15 mm $\leq X_{AVG} \leq X_{MIN}$ + 0.15 mm	$UEC \geq 0.50$
C(2.0)		$SC \geq 40\%$	$-0.85 \leq D' \leq 0.85$		X_{MAX} - 0.20 mm $\leq X_{AVG} \leq X_{MIN}$ + 0.20 mm	$UEC \geq 0.37$
D(1.0)		$SC \geq 20\%$	$-1.00 \leq D' \leq 1.00$		X_{MAX} - 0.25 mm $\leq X_{AVG} \leq X_{MIN}$ + 0.25 mm	$UEC \geq 0.25$
F(0.0)	Fails	$SC < 20\%$	$-1.00 > D' > 1.00$	otherwise	X_{MAX} - 0.25 mm $> X_{AVG} > X_{MIN}$ + 0.25 mm	$UEC < 0.25$

Table 9: Summary of MaxiCode Symbol Print Quality Parameters

independently as specified in Annexe C.2.5, then the Unused Error Correction grade shall be the lowest achieved by any of the subsets.

4.13.3 Overall Symbol Grade

The overall print quality grade for a MaxiCode symbol is the lowest of the six grades achieved above. Table 9 summarizes the test criteria grades.

4.13.4 Process Control Measurements

Several tools and methods are available which can perform useful measurements for monitoring and controlling the process of creating MaxiCode symbols. These include:

1. Symbol contrast readings from a linear bar code verifier.
2. A template overlay to visually check finder pattern growth, the orientation patterns, and overall symbol size.
3. A printed "zipper" pattern to visually indicate local grid nonuniformities.
4. Visual inspection for dark module growth and defects.

These tools and methods are described in Annexe L.

4.14 Reference Decode Algorithm

This reference decode algorithm finds the symbols in an image and decodes them. This is the decode algorithm used in the determination of symbol quality.

1. Locate potential finder patterns using a coarse locating algorithm such as the following template method.

   A one-dimensional template representing a line signal going through the center of the finder pattern is constructed as a square-wave template of limited length, depicted in Figure 10 in which a dotted line indicates the center of the finder pattern.

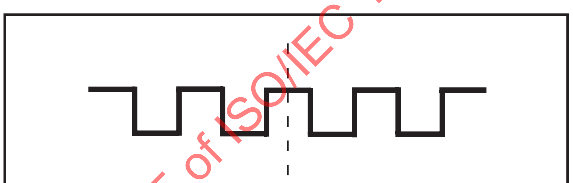


Figure 10: Square Wave Template

The finder pattern defined in MaxiCode is considered a composition of alternate three black rings and three white rings. The synthetic template going through the center of the finder pattern starts with a high intensity representing one side of the largest white ring bounded by the largest black ring and ends with a high intensity representing the other side of the same white ring. Given a MaxiCode image, a line signal starting from the first row of the image is extracted and compared against the template by calculating the cross-correlation coefficient. A signal intensity detection is employed to detect a low intensity of a width consistent with the expected size of the finder pattern, e.g. 3 pixels. If the low intensity is followed by a high intensity, it potentially signals a start of the finder pattern. Therefore the matching process does not start until such low intensity is detected. The correlation coefficient is recorded for every comparison in a line as the template moves along and the maximum is determined and compared to a threshold of 70%. Every line signal in the image is sequentially tested for a closest match.

If such a match happens, a line signal along the image column at the center of the matching signal is extracted for the similar template matching. If the correlation coefficient in the column matching does not show an acceptable match, the same procedures along the rows are resumed. Matching in both row and column indicates the location of a finder pattern in the given image. If the last row is reached before a finder pattern is found, this indicates there is no finder pattern in the given image.

2. Re-scale the image to a fixed horizontal and vertical resolution of 4 dpmm.

3. Collect the reflectance values of all the pixels in the immediate square area of the candidate finder pattern. Determine the black/white threshold using the methods described in Annexes C.2.2 and C.2.3.

4. Based on the threshold, polarize the module values on a scale of -127 to 128 with the light/dark threshold set at zero.

5. Verify that the finder pattern candidate identified is a valid finder pattern as follows:
   (a) Generate the fine correlation kernel by substituting the values from Table 8 and RES = 4 dpmm into Formula 1. This reduces to Formula 2.
   (b) Copy the finder pattern to a buffer ($I_{x,y}$) that is

$$K_{x,y} = \begin{cases} \cos\left[\pi * \dfrac{\text{Radius}_{x,y}}{2 * R_1 * \text{RES}}\right] & \text{for Radius}_{x,y} < R_1 * \text{RES} \\ -\sin\left[5 * \pi * \dfrac{[\text{Radius}_{x,y} - (R_1 * \text{RES})]}{(R_6 - R_1) * \text{RES}}\right] & \text{for } R_1 * \text{RES} < \text{Radius}_{x,y} < R_6 * \text{RES} \\ 0 & \text{otherwise} \end{cases}$$

Where:

RES	=	resolution of the image in dpmm
Kernel Size	=	Xmax = Ymax = 2 * R_6 * RES
$\text{Radius}_{x,y}$	=	$[(x - \text{Xmax}/2)^2 + (y - \text{Ymax}/2)^2]^{1/2}$

Formula 1: Correlation Kernel

$$K_{x,y} = \begin{cases} \cos\left[0.77 * \text{Radius}_{x,y}\right] & \text{for Radius}_{x,y} < 2.04 \\ -\sin\left[1.17 * [\text{Radius}_{x,y} - 2.04]\right] & \text{for } 2.04 < \text{Radius}_{x,y} < 15.48 \\ 0 & \text{otherwise} \end{cases}$$

Where:

Kernel Size	=	Xmax = Ymax = 31
$\text{Radius}_{x,y}$	=	$[(x - \text{Xmax}/2)^2 + (y - \text{Ymax}/2)^2]^{1/2}$

Formula 2: Correlation Kernel for RES = 4 dpmm

$$P = \frac{\sum_x \sum_y (K_{x,y} * I_{x,y})}{\text{Contrast} * \sum_x \sum_y (K_{x,y} * K_{x,y})}$$

Where:

$I_{x,y}$	=	image data of potential finder pattern
Contrast	=	contrast of the image data (i.e. MAX[$I_{x,y}$] - MIN[$I_{x,y}$])
P	=	percent correlation

Formula 3: Correlation Equation

the same size as the kernel. Place the center of the finder pattern at (Xmax/2, Ymax/2).

(c) Calculate the percent correlation as shown in Formula 3.

(d) If P > 85% the correlation passes

6. Blank the finder pattern with pixel values of zero.

7. Create a second image that is only the edge locations between light and dark. The edges shall be set to zero and the other regions shall be greater than zero.

8. If using a fixed position reader, blank the outside of a circle centered on the enhanced image with a pixel value calculated as the average of white hexagon samples inside the circle. The diameter of the circle is 95% of the symbol dimension (height).

9. Transform the space domain of the image to the frequency domain through a two-dimensional Fast Fourier transformation. The brightest point will be at (0,0) in the transform plane, corresponding to the DC component in the image. The six points surrounding the central point represent spacing, direction and intensity of the edges between hexagons.

10. Locate and filter bright points by eliminating any frequency domain points that do not correspond to the desired spacing and direction of hexagon boundaries. Since the image is real valued, the frequency domain is point symmetrical about the origin. Thus, only three bright points in the half transform plane are actually identified.

Note: As this is done in the frequency domain, there is no actual analysis of the spatial distribution of the modules. The bright spots in the frequency domain correspond to the harmonics of the spacing of the edges.

11. Perform two-dimensional inverse Fast Fourier transformation to return to the space domain, thereby restoring an image of hexagon centers.

Note: In the new image, the centers of the hexagons have high magnitude. The actual magnitude of the white centers at the hexagon centers is dependent on how many edges are in its neighborhood.

12. Determine the symbol orientation as follows. The MaxiCode hexagons produce three axes spaced 60 degrees apart. The direction of these three axes is established by the brightest points in the frequency domain. Based on the information on the bright points, the location of the orientation patterns can be computed. The information in the orientation patterns will determine the orientation of the image.

13. Convert the hexagon information into bit information and arrange these into a temporary sequential bit stream.

14. Separate the bit stream into the Primary and Secondary Messages, creating bit streams 1 - 120 and 121 to 864.

15. Apply error correction to the bit stream for the Primary Message.

    a. Compute syndromes.

    b. Compute the error locator polynomial using the Berlekamp-Massey algorithm.

    c. Compute error locations using Chien search.

    d. Compute the correct value of each erroneous codeword using Forney's algorithm.

    e. If a correctable number of errors has been detected from steps 15c and 15d, decode the data from the codewords. An alternate method is defined in Annex D.

16. Identify the error correction level used in the Secondary Message from the Mode bits.

17. Apply error correction to the bit stream for each segment of the Secondary Message using the substeps of step 15 above.

4.15 Transmitted Data

This section describes the standard transmission protocol for compliant readers. These readers may be programmable to support other transmission options.

4.15.1 Basic Interpretation

Encoded data shall not be transmitted from symbols in Mode 6. Otherwise all codewords shall be converted to a data stream as follows:

a. All error correction codewords shall be discarded.

b. All Shift and Latch symbology control characters shall perform their function to switch to other Code Sets. [NS] shall convert the following five codewords (equivalent to 30 binary bits) to 9 numeric digits.

c. The ECI codeword causes the next one to four codewords to be converted to a six digit number preceded by a backslash. (see Section 4.15.2).

d. All data shall be translated into user data as 8-bit bytes.

e. In modes 4 and 5, the data is transmitted in the same sequence as it is encoded in the symbol. In modes 2 and 3, the sequence of transmitted data follows the rules in Annexe B

4.15.2 Protocol for Extended Channel Interpretation

In systems where ECIs are supported, the use of a symbology identifier prefix is required with every transmission. Whenever an ECI codeword is encountered, it shall be transmitted as the escape character 92_{DEC} (or $5C_{HEX}$), which represents the character backslash (\) or reverse solidus in the default encodation. The next codeword(s) are converted into a 6-digit value, inverting the rules defined in Table 3. The 6-digit value is transmitted as the appropriate ASCII values (48-57).

Application software recognizing \nnnnnn should interpret all subsequent characters as being from the ECI defined by the 6-digit sequence. This interpretation remains in effect until the end of the encoded data or until another ECI sequence is encountered.

If the backslash (byte 92_{DEC}) needs to be used as encoded data, transmission shall be as follows. Whenever (ASCII 92_{DEC}) occurs as data, two bytes of that value shall be transmitted, thus a single occurrence is always an escape character and a double occurrence indicates true data.

Example:
Encoded data: A\\B\C
Transmission: A\\\\B\\C

Use of the symbology identifier assures that the application can correctly interpret the escape character.

4.15.3 Symbology Identifier

EN796 provides a standard procedure for reporting the symbology which has been read, together with options set in the decoder and any special features encountered in the symbol.

Once the structure of the data (including the possible use of any ECI) has been identified, the appropriate symbology identifier should be added by the decoder as a preamble to the transmitted data; if

ECIs are used the symbology identifier is required. See Annexe E for the symbology identifier and option values which apply to MaxiCode.

4.15.4 Transmitted Data Example

In this example, the two-character message " ¶ Ж " is to be encoded in Maxicode's Mode 4. " ¶ " is represented by a byte value of 182 in Maxicode's default character set (ECI 000003, which is equivalent to ISO 8859-1). "Ж" is a Cyrillic character not available in ECI 000003, but which can be represented in ISO 8859-5 (ECI 000007) by the same byte value of 182. The complete message can therefore be represented by inserting a switch to ECI 000007 after the first character, as follows:

The symbol encodes the message <¶> <Switch to ECI 000007> <Ж> using the following series of Maxicode codewords (note that a shift to Code Set E, followed by a codeword of value 47, encodes a byte value of 182:

[Shift E] [47] [ECI] [7] [Shift E] [47],

with decimal values of

[62], [47], [27], [7], [62], [47].

The decoder transmits the following bytes (including the symbology identifier prefix with an option value of 2, which indicates use of the ECI protocol):

93, 85, 50,182, 92, 48, 48, 48, 48, 48, 55, 182

which, if viewed entirely in the default interpretation, would appear graphically as:

]U2¶\000007¶

Note that the decoder is responsible for signalling the switch to ECI 000007, but not for interpreting the result.

ECI-aware software in the receiving application would delete the ECI escape sequence \000007, and the Cyrillic character "Ж" would be represented in a system-dependent manner (e.g. by changing the font in a desktop-publishing file). The final result would match the original message of " ¶ Ж ".

Annexe A (Normative)

MaxiCode Basic Character Encodation: Default Character Set

Symbol Character Value		Code Set A		Code Set B		Code Set C		Code Set D		Code Set E	
Dec	**Binary**	**Char**	**Dec**	**Char**	**Dec**	**Char**	**Dec**	**Char**	**Dec**	**Char**	**Dec**
0	000000	CR	13	‘	96	À	192	à	224	NUL	0
1	000001	A	65	a	97	Á	193	á	225	SOH	1
2	000010	B	66	b	98	Â	194	â	226	STX	2
3	000011	C	67	c	99	Ã	195	ã	227	ETX	3
4	000100	D	68	d	100	Ä	196	ä	228	EOT	4
5	000101	E	69	e	101	Å	197	å	229	ENQ	5
6	000110	F	70	f	102	Æ	198	æ	230	ACK	6
7	000111	G	71	g	103	Ç	199	ç	231	BEL	7
8	001000	H	72	h	104	È	200	è	232	BS	8
9	001001	I	73	i	105	É	201	é	233	HT	9
10	001010	J	74	j	106	Ê	202	ê	234	LF	10
11	001011	K	75	k	107	Ë	203	ë	235	VT	11
12	001100	L	76	l	108	Ì	204	ì	236	FF	12
13	001101	M	77	m	109	Í	205	í	237	CR	13
14	001110	N	78	n	110	Î	206	î	238	SO	14
15	001111	O	79	o	111	Ï	207	ï	239	SI	15
16	010000	P	80	p	112	Ð	208	ð	240	DLE	16
17	010001	Q	81	q	113	Ñ	209	ñ	241	DC1	17
18	010010	R	82	r	114	Ò	210	ò	242	DC2	18
19	010011	S	83	s	115	Ó	211	ó	243	DC3	19
20	010100	T	84	t	116	Ô	212	ô	244	DC4	20
21	010101	U	85	u	117	Õ	213	õ	245	NAK	21
22	010110	V	86	v	118	Ö	214	ö	246	SYN	22
23	010111	W	87	w	119	×	215	÷	247	ETB	23
24	011000	X	88	x	120	Ø	216	ø	248	CAN	24
25	011001	Y	89	y	121	Ù	217	ù	249	EM	25
26	011010	Z	90	z	122	Ú	218	ú	250	SUB	26
27	011011	[ECI]		[ECI]		[ECI]		[ECI]		[ECI]	
28	011100	FS	28	FS	28	FS	28	FS	28	[Pad]	
29	011101	GS	29	GS	29	GS	29	GS	29	[Pad]	
30	011110	RS	30	RS	30	RS	30	RS	30	ESC	27
31	011111	[NS]		[NS]		[NS]		[NS]		[NS]	
32	100000	space	32	{	123	Û	219	û	251	FS	28
33	100001	[Pad]		[Pad]		Ü	220	ü	252	GS	29
34	100010	“	34	}	125	Ý	221	ý	253	RS	30
35	100011	#	35	~	126	Þ	222	þ	254	US	31
36	100100	$	36	DEL	127	ß	223	ÿ	255	{C159}	159
37	100101	%	37	;	59	ª	170	¡	161	NBSP	160
38	100110	&	38	<	60	¬	172	¨	168	¢	162
39	100111	‘	39	=	61	±	177	«	171	£	163
40	101000	(	40	>	62	²	178	¯	175	¤	164
41	101001	)	41	?	63	³	179	°	176	¥	165
42	101010	*	42	[	91		181	´	180		166
43	101011	+	43	\	92	¹	185	·	183	§	167
44	101100	,	44	]	93	º	186		184	©	169
45	101101	-	45	^	94	¼	188	»	187	SHY	173
46	101110	.	46	_	95	½	189	¿	191	®	174
47	101111	/	47	space	32	¾	190	{C138}	138	¶	182

Default Character Set (Concluded)

Symbol Character Value		Code Set A		Code Set B		Code Set C		Code Set D		Code Set E	
Dec	Binary	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec
48	110000	0	48	,	44	{C128}	128	{C139}	139	{C149}	149
49	110001	1	49	.	46	{C129}	129	{C140}	140	{C150}	150
50	110010	2	50	/	47	{C130}	130	{C141}	141	{C151}	151
51	110011	3	51	:	58	{C131}	131	{C142}	142	{C152}	152
52	110100	4	52	@	64	{C132}	132	{C143}	143	{C153}	153
53	110101	5	53	!	33	{C133}	133	{C144}	144	{C154}	154
54	110110	6	54	|	124	{C134}	134	{C145}	145	{C155}	155
55	110111	7	55	[Pad]		{C135}	135	{C146}	146	{C156}	156
56	111000	8	56	[2 Shift A]		{C136}	136	{C147}	147	{C157}	157
57	111001	9	57	[3 Shift A]		{C137}	137	{C148}	148	{C158}	158
58	111010	:	58	[Pad]		[Latch A]		[Latch A]		[Latch A]	
59	111011	[Shift B]		[Shift A]		space	32	space	32	space	32
60	111100	[Shift C]		[Shift C]		[Lock In C]		[Shift C]		[Shift C]	
61	111101	[Shift D]		[Shift D]		[Shift D]		[Lock In D]		[Shift D]	
62	111110	[Shift E]		[Shift E]		[Shift E]		[Shift E]		[Lock In E]	
63	111111	[Latch B]		[Latch A]		[Latch B]		[Latch B]		[Latch B]	

Notes:

1. The relationship between the ASCII decimal value and the code set/codeword value remains constant whatever Extended Channel Interpretation is in use.

2. Characters {C128} to {C159} in the table have no default graphical interpretation.

Annexe B (Normative)

Mode 2 and Mode 3: Structured Carrier Message

Mode 2 and Mode 3 shall be reserved for use as a destination sortation symbol for use by carriers in the transport industry. This annexe describes the structure of the primary message, the alternative structures of the secondary message, covering both encoding and decoding. Also covered are the rules of Modes 2 and 3 when used with the Structured Append.

The common data structure shall be as defined in Table B1.

B.1 The Structure of the Primary Message

All Mode 2 and Mode 3 symbols, even if part of a Structured Append, shall have a common structure for the Primary Message. The following rules shall be used to convert data to the bit value:

1. No symbology shift characters shall be used to switch to and from numeric compaction or alphabetic encodation in the Primary Message.
2. The 3-digit class of service shall be represented as its 10-bit binary equivalent.
3. The 3-digit ISO country code shall be represented as its 10-bit binary equivalent.
4. Postal code is 36-bits.
   a. If the mode is 2 then 6 bits (42 to 39,32 to 31) encode the length (up to 9). The remaining 30 bits is the binary representation of the numeric postal code. In the case of country code 840, if the "+4" is unknown, then fill with zeroes.
   b. If the mode is 3, encode up to six characters from the printable subset of Code Set A. Shorter codes shall be filled with trailing spaces. Longer codes shall be truncated.
5. The mode shall be set to Mode 2 if the postal code is numeric and 3 if the postal code is alphanumeric.

Bit Value	Encoded Data	Structure
6 - 3	Mode	binary 0 to 15
36 - 33, 30 - 7, 2 -1	postal code[1,2]	numeric postal code (up to 9 digits)
42 - 39, 32 - 31	postal code length	for numeric postal codes only
42 - 39, 36 - 7, 2 - 1	postal code[1,2]	alphanumeric postal code
54 - 53, 48 - 43, 38 - 37	country code[1,3]	3 digit numeric (ISO 3166)
60 - 55, 52 - 49	service class[1]	3 digit numeric
120 - 61	ECC codewords	
121 - 844	Data message as specified by application	
625 - 864	SEC Codewords	

Notes:
1. The symbology control character [NS] shall not be used to encode the class of service, ISO country code, or all-numeric postal code.
2. The structure of the postal code is determined by the mode.
3. The country code is from ISO 3166.

Table B1: Structured Carrier Message

Example:
The following example illustrates how this is achieved:

* Class of service: 999
* ISO country code: 056 (for Belgium)
* Postal code: B1050
* Mode 3

Class 999 converts to binary as: 1111100111

Country code 056 converts to binary as: 111000

Adding leading zero bits to make a 10-bit string:

0000111000

The postal code is alphanumeric and is converted to a 36-bit binary string using Code Set A :

	codeword	binary
B	2	000010
1	49	110001
0	48	110000
5	53	110101
0	48	110000
space	32	100000

Mode 3 has the binary setting: 0011

The entire 60-bit string shall be converted into codewords, as illustrated in Figure B1.

B.2 Modes 2 and 3 Messages Beginning with "[)>$^{R}_{S}$01$^{G}_{S}$"

Messages which begin with the seven encoded data characters "[)>$^{R}_{S}$01$^{G}_{S}$" conform to particular open system standards.

Sample message:

[)>$^{R}_{S}$01$^{G}_{S}$96152382802$^{G}_{S}$840$^{G}_{S}$001$^{G}_{S}$1Z00004951$^{G}_{S}$
UPSN$^{G}_{S}$06X610$^{G}_{S}$159$^{G}_{S}$1234567$^{G}_{S}$1/1$^{G}_{S}$$^{G}_{S}Y^{G}_{S}$
634 ALPHA DRIVE$^{G}_{S}$PITTSBURGH$^{G}_{S}$PA$^{R}_{S}$^{E}o$_{T}$

In this format the identifier "[)>$^{R}_{S}$01$^{G}_{S}$" is followed by a date (yy), in this example "96".

B.2.1 Encoding

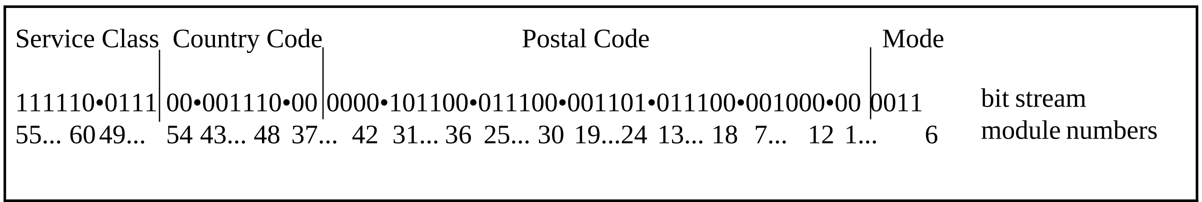
In messages of this type, data is encoded in a particular manner as follows:

1. The first nine data characters [)>$^{R}_{S}$01$^{G}_{S}$ yy are extracted to be encoded in the secondary message.
2. The next three data elements, representing respectively the postal code, country code, and service class and their separators ($^{G}_{S}$) that immediately follow the year are extracted from the source data.
3. The three $^{G}_{S}$ (decimal 29) characters are excluded from encoding. The three fields are encoded in the primary message as defined in Table B1 and the rules defined in Annexe B.1.
4. The remaining string of the data is then encoded in the secondary message after the header "[)>$^{R}_{S}$01$^{G}_{S}$yy".

It is recommended that this specific message handling method be isolated from the application in order to simplify the requirements for equipment users. That is, it should be embedded in the printer or printer driver. See Figure B2 for an example of this MaxiCode symbol.

B.2.2 Decoding

After the primary and secondary messages have been decoded, the original message is reconstructed by inserting the three primary data elements, each



Service Class	Country Code	Postal Code	Mode	
111110•0111	00•001110•00	0000•101100•011100•001101•011100•001000•00	0011	bit stream
55... 60	49... 54	43... 48 37... 42 31... 36 25... 30 19...24 13... 18 7... 12	1... 6	module numbers

Figure B1: Bit Assignments within the Structured Carrier Message

followed by ${}^{G}_{S}$ (decimal 29), in the specified order immediately following the nine character header which starts the secondary message. The entire message is then transmitted exactly as shown in the sample message above.

Mode: 2
Primary:
152382802
840
001
Secondary:
[)>${}^{R}_{S}$01${}^{G}_{S}$961Z00004951${}^{G}_{S}$UPSN${}^{G}_{S}$06X610${}^{G}_{S}$159${}^{G}_{S}$1234567${}^{G}_{S}$1/1${}^{G}_{S}{}^{G}_{S}$Y${}^{G}_{S}$
634 ALPHA DR${}^{G}_{S}$PITTSBURGH${}^{G}_{S}$PA${}^{R}_{S}{}^{E}_{O_T}$

Figure B2: Mode 2 Symbol Example

B.3 Modes 2 and 3 Messages Not Beginning with "[)>${}^{R}_{S}$01${}^{G}_{S}$"

In modes 2 and 3 primary messages are always data specific and encode postal code, country code and service class. These data elements along with the secondary message contents shall be supplied to the encoder in such a way that they can be distinguished one from the other. The three primary data elements should be supplied in the above order separated by ${}^{G}_{S}$ (decimal 29) immediately followed by the secondary message contents. In addition, each data element shall be of the correct type, e.g. postal codes in mode 2 must be all numeric. The standard message format is then:

postal_code${}^{G}_{S}$country_code${}^{G}_{S}$service_class${}^{G}_{S}$
secondary_message${}^{E}o_{T}$

Sample Message:

524032140${}^{G}_{S}$840${}^{G}_{S}$001${}^{G}_{S}$AIM USA${}^{G}_{S}$
634 ALPHA DRIVE${}^{G}_{S}$PITTSBURGH${}^{G}_{S}$PA${}^{E}o_{T}$

B.3.1 Encoding

The first three data elements are the postal code, country code and service class respectively, each delimited by ${}^{G}_{S}$ (decimal 29). These data elements are encoded in the primary message as described in Annexe B.1. The remainder of the message:

AIM USA${}^{G}_{S}$634 ALPHA DRIVE${}^{G}_{S}$
PITTSBURGH${}^{G}_{S}$PA${}^{E}o_{T}$

is encoded in the secondary message.

B.3.2 Decoding

After the primary and secondary messages have been decoded, the original message is reconstructed by placing the three primary data elements, each followed by ${}^{G}_{S}$ (decimal 29), in the specified order immediately preceding the secondary message string. The entire message is then transmitted exactly as shown in the sample message above.

B.4 Modes 2 and 3 and Structured Append

The basic rules for Structured Append and Modes 2 and 3 are specified in Section 4.9.2. Additional advice is provided below.

B.4.1 Encoding Considerations

The primary message (encoding postal code, country code, and service class) shall be repeated in each subsequent Mode 2 or Mode 3 symbol. Therefore, allowing for the Structured Append codewords, only the codewords from s_{23} onwards are available for the secondary message.

B.4.2 Decoding Considerations

When Structured Append is used with Modes 2 and 3 symbols, the Primary Message may be decoded from any of the symbols in the Structured Append.

When a Mode 2 or Mode 3 symbol with a Structured Append indicator in the first two positions of the secondary message is encountered by a reader, the reader shall reconstruct the entire data message as follows:

A. If the data "[)>${}^{R}_{S}$01" is decoded from codewords s_{23} to s_{30} (i.e. positions 3 to 10 of the secondary message) of the first symbol:
   1. Decode the first symbol as per Annexe B.2.2.

2. Decode subsequent symbols from codeword s_{23} onwards, ignoring the data in the Primary Message.
3. Reconstruct the data in the sequence of:
   a. the format header "[)>$^R{}_S$01$^G{}_S$yy"
   b. the data from the primary message, with the three $^G{}_S$ characters re-inserted in their correct position
   c. the data from the secondary message from the first Structured Append symbol
   d. the data from subsequent secondary message(s)

B. If the data "[)>$^R{}_S$01" is not decoded from codewords s_{23} to s_{30} (i.e. positions 3 to 10 of the secondary message) of the first symbol:
1. Decode the first symbol as per Annexe B.3.2
2. Decode subsequent symbols from codeword s_{23} onwards, ignoring the data in the Primary Message.
3. Reconstruct the data in the sequence of:
   a. the data from the primary message, with the three $^G{}_S$ characters re-inserted in their correct position
   b. the data from the secondary message from the first Structured Append symbol
   c. the data from subsequent secondary message(s)

A separator character $^G{}_S$ (decimal 29) shall follow each data element. This message is then transmitted as an ASCII character stream.

In all other cases the transmitted message must be an ASCII character stream representing the concatenation of the postal code, country code, service class and secondary message, in that order.

Annexe C (Normative)

2D Matrix Bar Code Print Quality - Guideline

This annexe presents a framework for assessing the print quality of a 2D matrix bar code symbol, which can be adapted to any matrix symbology. The method described here parallels in many ways the ANSI X3.182 guideline for assessing print quality of linear bar code symbols. It starts by obtaining a high resolution grey-scale image of the symbol under controlled illumination and viewing conditions. The stored image is then analyzed for the parameters of Decode, Symbol Contrast, “Print” Growth, Axial Nonuniformity, and Unused Error Correction. The final symbol assessment is the lowest grade achieved for these five parameters and any others specified for a given symbology or application.

The procedures presented here must necessarily be augmented by the reference decode algorithm and other measurement details within any companion symbology specification, and they may also be altered or overridden as appropriate by governing symbology or application specifications.

C.1 Obtaining the Test Image

A test image of the symbol shall be obtained in a configuration that mimics the typical scanning situation for that symbol, but with substantially higher resolution, uniform illumination, and at best focus. Specialized applications clearly must dictate the color and angle of symbol illumination as well as the required imaging resolution, but the following general test setup should work suitably for many open applications.

A standard monochrome video camera shall image the test symbol directly on axis with its center and normal to its plane. The lens used shall be appropriate to frame the entire symbol (including any required quiet zones) in good focus, and with a sufficiently small field of view to minimize optical distortions. Light illumination shall uniformly flood the symbol area from at least two sides with a 45 degree angle of incidence. Test images can be captured with 8-bit greyscale digitization using standard frame capture equipment, and the grey-scale shall be calibrated using targets of known diffuse reflectance.

Regardless of the exact optical setup, two principles should govern its selection. First, the test image’s greyscale shall be nominally linear and not be adjusted in any way to either enhance contrast or improve appearance. Second, the image resolution shall be adequate to produce consistent readings, which generally requires that the module widths and heights span at least five image pixels.

C.2 Assessing Symbol Parameters

C.2.1 Decode

A symbology’s reference decoding algorithm shall be applied to the test image. If it achieves a valid decode, the Decode grade is “A”, otherwise it is “F”.

The Decode parameter tests on a Pass/Fail basis whether the symbol, when optimally imaged, has all its features sufficiently correct to be readable.

Beyond this; the initial reference decode performs three additional tasks needed for subsequent measurement of the other symbol quality parameters. First, it locates and defines the area covered by the test symbol in the image. Second, it adaptively creates a grid mapping of the data module centers so as to sample them. Third, it performs error correction, detecting if symbol damage has consumed any of the error budget. These images, image coordinates, and error decoding each facilitate one or more of the following measurements.

C.2.2 Symbol Contrast

Within the grey-scale image, all of the image pixels which fall within the area of the test symbol, extending outward to the limits of any required quiet zones shall be sorted by their reflectance values to select the darkest 10% of the pixels and the lightest 10% of the pixels. Calculate the arithmetic mean of the reflectance of the darkest 10% and the arithmetic mean of the reflectance of the lightest 10%. The difference of the two means is the Symbol Contrast, SC.

The Symbol Contrast grade is determined by

A (4.0)	if $SC \geq 70\%$
B (3.0)	if $SC \geq 55\%$
C (2.0)	if $SC \geq 40\%$
D (1.0)	if $SC \geq 20\%$
F (0.0)	if $SC < 20\%$

Symbol Contrast tests that the two reflective states in the symbol, namely light and dark, are sufficiently and consistently distinct throughout the symbol.

C.2.3 "Print" Growth

Calculate a reflectance threshold halfway between the dark and light means from Annexe C.2.2. Create a secondary binary image distinguishing dark and light regions using the threshold.

The print growth parameter, the extent to which dark or light markings appropriately fill their module boundaries, is an important indication of process quality which affects reading performance. The particular graphical structures most indicative of element growth or shrinkage from nominal dimensions will vary widely between symbologies, and shall be defined within their specifications, but will generally be either fixed structures or isolated elements whose dimension(s) D is/are determined by counting pixels in the binary digitized image and most likely averaged. More than one dimension, for example both horizontal and vertical growth, may be specified and checked independently. Each checked dimension shall have specified both a nominal value D_{NOM} and maximum D_{MAX} and minimum D_{MIN} allowed values. Each measured D shall be normalized to its corresponding nominal and limit values:

$$\begin{aligned} D' &= (D - D_{NOM}) / (D_{MAX} - D_{NOM}) \quad \text{if } D > D_{NOM} \\ &= (D - D_{NOM}) / (D_{NOM} - D_{MIN}) \quad \text{otherwise.} \end{aligned}$$

Print growth is then graded according to:

A (4.0)	if $-0.50 \leq D' \leq 0.50$
B (3.0)	if $-0.70 \leq D' \leq 0.70$
C (2.0)	if $-0.85 \leq D' \leq 0.85$
D (1.0)	if $-1.00 \leq D' \leq 1.00$
F (0.0)	if $D' < -1.00$ or $D' > 1.00$

Print Growth tests that the graphical features comprising the symbol have not grown or shrunk from nominal so much as to hinder readability with less optimum imaging conditions than the test conditions.

C.2.4 Axial Nonuniformity

2D matrix symbols include data fields of modules nominally lying in a regular polygonal grid, and any reference decode algorithm must adaptively map the center positions of those modules to extract the data. Axial Nonuniformity measures and grades the spacing of the mapping centers, i.e. the sampling points, in the direction of each of the grid's major axes.

The spacings between adjacent sampling points are independently sorted for each polygonal axis, then the average spacing X_{AVG} along each axis is computed. Axial Nonuniformity is a measure of how much the sampling point spacing differs from one axis to another, namely:

$$AN = abs(X_{AVG} - Y_{AVG}) / ((X_{AVG} + Y_{AVG})/2)$$

where abs() yields the absolute value. If a symbology has more than two major axes, then AN is computed for those two average spacings which

Grade	Reference Decode	Symbol Contrast	"Print" Growth	Axial Nonuniformity	Unused Error Correction
A(4.0)	Passes	$SC \geq 70\%$	$-0.50 \leq D' \leq 0.50$	$AN \leq 0.06$	$UEC \geq 0.62$
B(3.0)		$SC \geq 55\%$	$-0.70 \leq D' \leq 0.70$	$AN \leq 0.08$	$UEC \geq 0.50$
C(2.0)		$SC \geq 40\%$	$-0.85 \leq D' \leq 0.85$	$AN \leq 0.10$	$UEC \geq 0.37$
D(1.0)		$SC \geq 20\%$	$-1.00 \leq D' \leq 1.00$	$AN \leq 0.12$	$UEC \geq 0.25$
F(0.0)	Fails	$SC < 20\%$	$D' < -1.00$ or $D' > 1.00$	$AN > 0.12$	$UEC < 0.25$

Table C1: Summary of 2D Matrix Bar Code Print Quality Parameters

differ the most. Axial Nonuniformity is then graded as:

A (4.0)	if $AN \leq 0.06$
B (3.0)	if $AN \leq 0.08$
C (2.0)	if $AN \leq 0.10$
D (1.0)	if $AN \leq 0.12$
F (0.0)	if $AN > 0.12$

Axial Nonuniformity tests for uneven scaling of the symbol which would hinder readability at some non-normal viewing angles more than others.

C.2.5 Unused Error Correction

The correction capacity of Reed-Solomon decoding is expressed in the equation:

$$e + 2t \leq d - p$$

where:

e	is the number of erasures,
t	is the number of errors,
d	is the number of error correction codewords,
p	is the number of codewords reserved for error detection.

Values for d and p are defined by symbology specification (often depending on symbol size), while e and t are determined during the successful reference decode. The amount of Unused Error Correction is computed as:

$$UEC = 1.0 - (e+2t)/(d-p).$$

In symbols with more than one (e.g., interleaved) Reed-Solomon block, UEC shall be calculated for each block independently, then the lowest value graded as:

A (4.0)	if $UEC \geq 0.62$
B (3.0)	if $UEC \geq 0.50$
C (2.0)	if $UEC \geq 0.37$
D (1.0)	if $UEC \geq 0.25$
F (0.0)	if $UEC < 0.25$

The Unused Error Correction parameter tests the extent to which regional or spot damage in the symbol has eroded the reading safety margin that error correction provides.

C.3 Overall Symbol Grade

The overall symbol grade is the lowest of the parameter grades achieved above. Table C1 summarizes the test parameters and grade levels.

Annexe D (Normative)

Error Correction Algorithm

The calculation described below follows the Peterson-Gorenstein-Zierler Algorithm which corrects errors using the Reed-Solomon error correction codewords. Erasures will be corrected as errors by initially filling any erasure character positions with dummy values.

All calculations are done using the Galois Field GF(64) arithmetic operations. Addition and subtraction are equivalent to the binary exclusive-or operation. Multiplication and division can be performed using log and antilog tables. A subroutine to generate these tables is part of the MaxiCode error correction program found on the MaxiCode Development Diskette.

Construct the symbol character polynomial C(x) $= C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \ldots + C_1x^1 + C_0$ where the n coefficients are the codewords read with C_{n-1} being the first symbol character and where n is the total number of symbol characters.

Calculate i syndrome values S_0 through S_{i-1} by evaluating C(x) at $x = 2^k$ for k = 1 through i and where i is the number of error correction codewords in the symbol.

Form and solve j simultaneous equations with j unknowns L_0 through L_{j-1} using the i syndromes:

$$\begin{aligned} S_0L_0 + S_1L_1 + \ldots + S_{j-1}L_{j-1} &= S_j \\ S_1L_0 + S_2L_1 + \ldots + S_jL_{j-1} &= S_{j+1} \\ &\vdots \\ S_{j-1}L_0 + S_jL_1 + \ldots + S_{2j-2}L_{j-1} &= S_{2j-1} \end{aligned}$$

where j is i/2.

Construct the error locator polynomial:

$$L(x) = L_{j-1}x^j + L_{j-2}x^{j-1} + \ldots + L_0x + 1$$

from the j values of L obtained above. Evaluate L(x) at $x = 2^k$ for k = 0 through n-1 where n is the total number of symbol characters in the symbol. Whenever $L(2^k) = 0$, an error location is given by n-1-k. If more than j error locations are found, the symbol is not correctable.

Save the error locations in m error location variables E_0 through E_{m-1} where m is the number of error locations found. Form and solve m simultaneous equations with m unknowns X_0 through X_{m-1} (the error magnitudes) using the error location variables E and the first m syndromes S:

$$\begin{aligned} E_0X_0 + E_1X_1 + \ldots + E_{m-1}X_{m-1} &= S_0 \\ E_0{}^2X_0 + E_1{}^2X_1 + \ldots + E_{m-1}{}^2X_{m-1} &= S_1 \\ E_0{}^3X_0 + E_1{}^3X_1 + \ldots + E_{m-1}{}^3X_{m-1} &= S_2 \\ &\vdots \\ E_0{}^mX_0 + E_1{}^mX_1 + \ldots + E_{m-1}{}^mX_{m-1} &= S_{m-1}. \end{aligned}$$

Add the error magnitudes X_0 through X_{m-1} to the symbol character values at the corresponding error locations E_0 through E_{m-1} to correct the errors.

This algorithm, written in C, is available from AIM$_{®}^{USA}$ on the MaxiCode Developers Diskette, see Reference Document section.

Annexe E (Normative)

Symbology Identifiers

EN796 provides a uniform methodology for reporting the symbology read, options set in the reader and any special features of the symbology encountered.

The symbology identifier for MaxiCode is:

]U*m*

where:

] is the symbology identifier flag character (ASCII 93)

U is the symbology identification character for MaxiCode

m is a modifier character with one of the values defined in Table E1.

Option Value	Option
0	Symbol in mode 4 or 5
1	Symbol in mode 2 or 3
2	Symbol in mode 4 or 5 supporting ECI protocol
3	Symbol in mode 2 or 3 supporting ECI protocol in secondary message(s)
(Permissible values of m: 0,1, 2, 3)	

Table E1: Symbology Identifier Option Values for MaxiCode

Annexe F (Informative)

Use of Numeric Shift, Shift, Latch, and Lock-In Characters

Optimal encodation efficiency can be achieved by using the following guidelines. Source routines in C are provided on the MaxiCode Development Diskette both for message encoding, using these guidelines, and for message decoding.

F.1 Numeric Shift

If 9 or more digits occur in a string, consider using [NS] as follows.

1. Encode [NS].
2. Subdivide the string into its first 9-digit block.
3. Convert the 9-digit decimal number to a 30-bit binary value. If the number of bits in the binary string is less than 30, the appropriate number of non-significant zero bits shall be added to the most significant positions of the string.

4. If there are 9 or more digits remaining, continue from Step 1, else encode any remaining digits as characters in Code Set A.

Example:

Decimal Value: 123456789

Binary equivalent (most significant bit first):

111010110111100110100010101

Adding leading zeros:

000111010110111100110100010101

Symbol character modules:

000111
010110
111100
110100
010101

Codewords:

7
22
60
52
21

F.2 Switching from Code Set A to Code Set B

When in Code Set A, if the next characters are only in Code Set B, it is necessary to switch code sets.

1. Use [Latch B] if the next two, or more, characters are from Code Set B.

2. Use [Shift B] if only the next character is from Code Set B.

F.3 Switching from Code Set B to Code Set A

When in Code Set B, if the next characters are only in Code Set A it is necessary to switch code sets. (Note: There are five punctuation characters [ASCII Values 32, 44, 46, 47, and 58] which appear in both code sets.) Depending on the data characters which follow, particular switching characters provide most efficient encodation.

1. Use [NS] if the next nine, or more, characters are numeric digits (see Annexe F.1 for more detailed advice).

2. Use [Latch A] if the next four or more characters are from Code Set A.

3. Use [3 Shift A] if only the next three characters are from Code Set A.

4. Use [2 Shift A] if only the next two characters are from Code Set A.

5. Use [Shift A] if only the next character is from Code Set A and data continues thereafter in Code Set B.

F.4 Using Lock-In to Latch to Code Sets C, D or E

When in Code Sets A or B, if the next four or more characters are from one of Code Sets C, D, or E, switching to those characters should be achieved as follows:

1. Use the appropriate [Shift C, D, or E]

2. Immediately encode the appropriate [Lock-In C, D or E]

3. Encode the data characters from the selected Code Set.

4. At the end of encoding, if it is necessary to switch to another Code Set:
   a. If the characters are in Code Set A or B, use [Latch A or B] as appropriate to switch.

   b. If the characters are in Code Set C, D or E, use [Shift C, D, or E] to encode a single character or [Shift C, D, or E] and the appropriate [Lock-In] to encode more characters.

F.5 Illustrative Example

Encoding the following text as four lines of data using the switching rules to minimize space can be illustrated below:

Data	**Number of Data Characters**
Comité Européen de Normalisation	32
rue de Stassart 36	18
B-1050 BRUXELLES	16
TEL +3225196811	15
TOTAL	81

For simplicity of illustration the codewords are not shown, only the encoded data and the symbology control characters used to effect the switches.

Data and Symbology Characters	**Number of Codewords**
C[Latch B]omit[Shift D]é• [Shift A]Europ[ShiftD]éen•de•[Shift A]Normalisation {FS}	38
rue•de•[Shift A]Stassart•[Latch A]36 {FS}	21
B-1050•BRUXELLES {FS}	17
TEL•+[N/S]bbbbb1	12
TOTAL	88

In this example:

•	indicates the data character for space
[ ]	and the text between, identifies a symbology control character
{FS}	identifies Field Separator
bbbbb	indicates that 5 codewords have been used for numeric compaction

Annexe G (Informative)

User Assessment of Encodation Capacity

In any application it should be remembered that MaxiCode symbology has features which make it suitable for high speed omnidirectional scanning; with an encodation capacity greater than a linear bar code symbology, but with a smaller capacity than some other 2-Dimensional symbologies. Its features (as defined in Sections 4.1.1 and 4.1.2) should be taken into account in the design of any application.

The following guidelines should be used to assess the encodation capacity of one or more MaxiCode symbols permitted for an application. These guidelines have three uses:

For the printer manufacturer or printing software manufacturer to create symbol design algorithms.

For the designer of an application standard to ensure that the design parameters produce a viable solution.

For the user who has a particular constraint which must be met. For most users it is expected that printing equipment and software may allow user definable parameters to be input so that the MaxiCode symbol is automatically and correctly generated.

The guidelines are as follows:

1. The maximum number of codewords per MaxiCode symbol is 93 or 77 depending on the selected level of error correction.

2. If all data to be encoded is from the default interpretation and is in Code Set A, data is encoded into codewords on a 1:1 ratio.

3. If numeric data is 9 or more digits long, the use of [NS] increases the data encodation capacity. For example, if the data to be encoded is all-numeric, 138 or 113 digits can be encoded depending on the level of error correction selected.

4. If the default interpretation can be used, but the data requires the use of Code Sets B to E, switching will be necessary, consuming some codewords.

5. If the receiving application supports ECI protocols, then switching to a different ECI, which requires two or more codewords, may be more

efficient than staying in the default interpretation.

6. The overhead (e.g. the use of particular syntax etc.) for complying to a particular application standard needs to be taken into account.

Where the data cannot be encoded in the required number of MaxiCode symbols, the printing software should provide some output to the user to enable a revision of the parameters:

a. To use a greater number of symbols in Structured Append, up to the maximum of eight.

b. To reduce error correction from the higher to lower level.

c. In extreme cases, to review the data content.

Annexe H (Informative)

A MaxiCode Encoding Example

This Annexe examines the process of encoding a short test message "MaxiCode (19 chars)" in a MaxiCode symbol in two different ways, either to augment a Structured Carrier Message in Modes 2 and 3 or as the entire encoded message in Modes 3 to 5.

There are four major steps, two of them illustrated in Table H1. The input message is regarded as a sequence of 19 ASCII data characters which are designated "m_1" through "m_{19}" as shown in the leftmost column.

Step 1 - Generating a Data Codeword Sequence

Regardless of the planned symbol mode, Step 1 is the same: using entries from the encodation table in Annexe A, the data characters are interspersed as needed with code set shift and latch commands, creating the equivalent sequence of codewords "c_1" through "c_{23}" as shown also in Table H1.

MaxiCode message encoding starts in Code Set A, so the opening "M" is directly encoded by value 13 (c_1). Since the message moves into lowercase letters, it is necessary to latch into Code Set B (c_2), then encode the next three letters directly (c_3 - c_5). As the capital "C" is immediately followed by lowercase letters, a temporary shift to Code Set A (c_6) precedes the capital "C", then the encoding automatically reverts back to Code Set B for the next four characters (c_8 - c_{11}). A triple shift to A (c_{12}) is next used for encoding three successive data characters (c_{13} - c_{15}) in Code Set A, but then encoding reverts again to Code Set B for the next six characters (c_{16} - c_{21}). Finally another shift to Code Set A (c_{22}) is needed to encode the final closing parenthesis (c_{23}) in the data message.

Guidelines for choosing the shift and latch commands that efficiently pack messages into codeword sequences are given in Annexe F. The end result of message encoding is a sequence of codewords somewhat longer than the original data message but usually not exactly the right length to fill the data regions of a MaxiCode symbol. As needed then, [Pad] codewords are appended to the end of the codeword sequence to fill out all the data symbol character positions in the symbol.

Step 2 - Assigning Codewords to Symbol Character Positions.

Two user decisions govern how many data codewords fit into a MaxiCode symbol and where they are placed. First, the user may elect to devote MaxiCode's primary message region to a Structured Carrier Message (Modes 2 and 3) whereby only the secondary message is available for a string of data codewords with Standard Error Correction. Alternately, the primary message can be used to augment the message capacity in which case the user may select either Standard (Modes 4 and 6) or Enhanced (Mode 5) Error Correction in the secondary message. Symbol character assignments s_N for both cases are illustrated in Table H1, where "N" corresponds exactly to the character positions shown in Figure 4.

Symbol mode selection (see Table 4) is always indicated in the four least significant bits of symbol character "s_1". There are two distinct cases:

a. Modes 2 and 3

A Structured Carrier Message in Modes 2 and 3 fills symbol characters "s_2" through "s_{10}", plus the two most significant bits of "s_1", with compacted destination data as detailed in Annexe B. Thus the sequence of data message codewords is positioned exclusively in the secondary message, as shown in the middle columns of Table H1, starting from "s_{21}" upward and padded out to "s_{104}". For the purposes