



5.1 Structure

There are seven different GS1 DataBar Symbols and these are further classified into **three groups** based on the structure of the symbol.

The First Group

- GS1 DataBar Omnidirectional
- GS1 DataBar Stacked Omnidirectional
- GS1 DataBar Stacked
- GS1 DataBar Truncated

These GS1 DataBar Symbols encode Application Identifier (01) and each one contains four data characters and two finder patterns. They are capable of being scanned in four separate segments, each consisting of a data character and an adjacent finder pattern. The two finder patterns together encode a modulo 79 check value for data security.

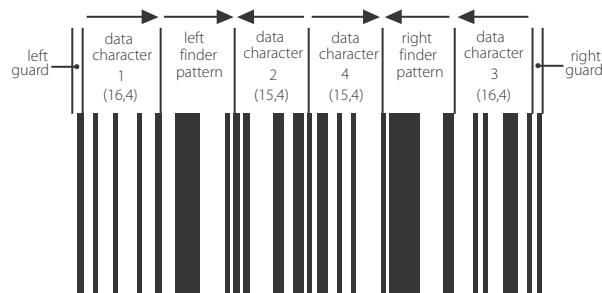


Figure 20 - Group One Symbol Structure

The left and right Guard Bar Patterns consist of a narrow space and narrow bar. The symbols do not require Quiet Zones.

The Second Group

- GS1 DataBar Limited

This GS1 DataBar Symbol encodes Application Identifier (01) but the first digit can only be zero or one.

The following figure shows the structure of the GS1 DataBar Limited Bar Code. A GS1 DataBar Limited Symbol contains two data characters and a check character. The check character encodes a modulo 89 check value for data security.

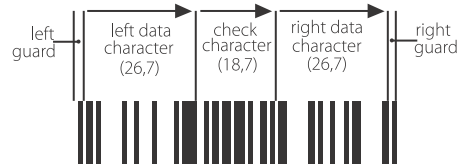


Figure 21 - GS1DataBar Limited Bar Code Structure

The left and right Guard Bar Patterns consist of a narrow space and narrow bar. The GS1 DataBar Limited Bar Code does not require Quiet Zones.

The Third Group

- GS1 DataBar Expanded
- GS1 DataBar Expanded Stacked

These symbols are variable in length and capable of encoding up to 74 numeric or 41 alphabetic characters. They can encode the Application Identifier (01) and additional information such as Batch Number using the appropriate Application Identifiers.

The figure below shows the structure of a six-segment GS1 DataBar Expanded Symbol. GS1 DataBar Expanded Symbols contain a check character, 3 to 21 data characters and 2 to 11 finder patterns, depending on the symbol length. GS1 DataBar Expanded is capable of being scanned in separate segments, each segment consisting of a data character or check character and the adjacent finder pattern. The check character encodes a modulo 211 check value for data security.

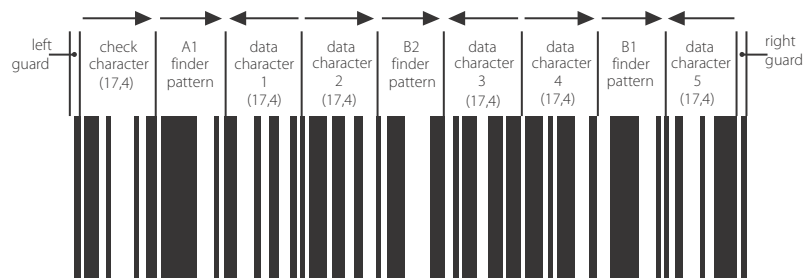


Figure 22 - GS1 DataBar Expanded Structure

The left and right Guard Bar Patterns consist of a narrow space and narrow bar. GS1 DataBar Expanded versions do not require Quiet Zones.



5.2 Encodation

- Encodable character set:
 - GS1 DataBar Omnidirectional, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Stacked, GS1 DataBar Truncated and GS1 DataBar Limited: digits 0 through 9 (with the restriction of GS1 DataBar Limited to 0 or 1 in the first digit)
 - GS1 DataBar Expanded versions: a subset (Table 1) of the International Standard ISO/IEC 646, consisting of the upper case and lower case letters, digits, spaces, and 20 selected punctuation characters in addition to the Function 1 Symbol Character (FNC1)
- Symbol character structure: Different (n,k) symbol characters are used for each member of the family, where each symbol character is n modules in width and is composed of k bars and k spaces.
- Code type: continuous, linear bar code symbology
- Maximum numeric data capacity (including implied Application Identifiers (AIs) where appropriate, but not including any encoded FNC1 characters):
 - GS1 DataBar Omnidirectional, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Stacked, GS1 DataBar Truncated and GS1 DataBar Limited: AI (01) plus a 14-digit numeric item identification
 - GS1 DataBar Expanded versions: 74 numeric or 41 alphabetic characters
- Error detection:
 - GS1 DataBar Omnidirectional, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Stacked, GS1 DataBar Truncated: mod 79 checksum
 - GS1 DataBar Limited: mod 89 checksum
 - GS1 DataBar Expanded versions: mod 211 checksum
- Character self-checking
- Bidirectionally decodable
- Quiet Zones: none required

Additional Features

Additional GS1 DataBar features include:

- Data compaction: Each member of the GS1 DataBar family has data compaction methods optimised for the data strings that it will encode. GS1 DataBar Expanded versions are also optimised for specific sequences of Application Identifiers (AIs) that are commonly used.
- Component linkage: All GS1 DataBar Symbols include a linkage flag. If the linkage flag is 0, then the GS1 DataBar Symbol stands alone. If the linkage flag is 1, then a 2D Composite Component and its separator pattern is printed above the GS1 DataBar Symbol with the separator pattern aligned and contiguous to the GS1 DataBar symbol.
- Edge to similar edge decoding; All GS1 DataBar family data characters, finder patterns, and symbol Check Characters can be decoded using edge-to-edge measurements.
- GS1-128 Symbol emulation: Readers set to the GS1-128 Symbol emulation mode transmit the data encoded within the GS1 DataBar Symbol as if the data were encoded in one or more GS1-128 Symbols.
- Large data characters: GS1 DataBar Symbol's data characters do not directly correspond to the encoded data character. The symbol's data characters encode thousands of possible combinations to increase the encoding





efficiency. They are then combined mathematically to form the encoded data string. Details can be found in ISO/IEC 24724: Information technology, automatic identification and data capture techniques Reduced Space Symbology (RSS) bar code symbology specification (formerly RSS, now GS1 DataBar)

5.3 Compressed Element String Sequences

While GS1 DataBar Expanded Symbols can encode any sequence of Application Identifier (AI) data up to the maximum capacity of the symbol, certain sequences of AI Element Strings have been selected for special compression in GS1 DataBar Expanded versions. If the application requires the use of the AI Element Strings in one of these sequences and they are used in the predefined sequence, a smaller symbol will result.

The selected sequences are two types: fixed length, where the sequence of selected AI Element Strings is the only data encoded, and open-ended, where the sequence occurs at the start of the symbol's data, and other AI Element Strings may be added following the sequence. If the data to be encoded in a GS1 DataBar Expanded Symbol starts with a sequence defined as fixed length but is followed by additional AI Element Strings, all the data will be encoded normally without special compression.

5.3.1 Fixed-Length Sequences

AI (01) - Weight with Limited Range

This sequence consists of the two Application Identifier (AI) Element Strings AI (01), followed by AI (3103), AI (3202), or AI (3203) for weight. The AI (01) Element String must start with an Indicator value of 9 for variable measure. Using AI (3103) (weight in grams), the special compression can only be applied up to a maximum weight of 32.767 kg. Using AI (3202) (weight in 0.01 lbs.) the special compression can only be applied up to a maximum weight of 99.99 lbs. Using AI (3203) (weight in 0.001 lbs.) the special compression can only be applied up to a maximum weight of 22.767 lbs. If the weight is in excess of these values, the sequence defined in "AI (01) - Weight and Optional Date" on page 35 still enables special compression to be performed.

AI (01) - Weight and Optional Date

This sequence consists of the two or three Application Identifier (AI) Element Strings AI (01), AI (310n), or AI (320n) for weight (n ranging from 0 to 9) and optionally AI (11), AI (13), AI (15), or AI (17) for date. The AI (01) Element String must start with an Indicator value of 9 for variable measure. If the date is not needed, this sequence still gives additional compression when the weight is outside the ranges required by the AI (01) and weight with limited range sequence above.





5.3.2 Open-Ended Sequences

AI (01) and Price

This sequence consists of the two Application Identifier (AI) Element Strings, AI (01), followed by AI (392x) for price or AI (393x) for price with ISO currency code (where x is in the range of 0 to 3). The AI (01) Element String must start with an Indicator value of 9 for variable measure. For example, this sequence is used for an AI (01) Element String, price and weight, because the fixed-length sequence AI (01) and weight does not give additional compression if the AI Element String for price is added to the end since the length of the sequence is fixed.

AI (01)

Any sequence that starts with AI (01) will have special compression applied to the AI (01). So when the data includes AI (01), it should always be the first Element String encoded.

5.4 Data Transmission and Symbology Identifier Prefixes

5.4.1 Default Transmission Mode

The GS1 System requires the use of Symbology Identifiers. GS1 DataBar family symbols are normally transmitted using Symbology Identifier prefix "Je0" (See "Symbology Identifiers" on page 6.) For example, a GS1 DataBar Symbol encoding AI (01) Element String 09312345678907 produces the transmitted data string "Je00109312345678907". Data transmission follows the same principles that apply to the concatenation of AI Element Strings in any bar code that encodes GS1 Application Identifiers (see Chapter 9 Page 71).

If a 2D Composite Component accompanies a GS1 DataBar family linear symbol, the AI Element String data from the 2D Composite Component immediately follows the linear component's data. However, readers have an option to transmit only the linear component data and ignore the 2D Composite Component.

5.4.2 GS1-128 Symbol Emulation Mode

Readers also have an option for GS1-128 Symbol emulation mode. This mode emulates the GS1-128 Symbol for data transmission. This mode is used for applications programmed for GS1-128 but not yet programmed to recognise the Symbology Identifier prefix "Je0." The Symbology Identifier for GS1-128 emulation mode is "JC1."





GS1 DataBar Expanded Symbols that exceed 48 data characters are transmitted as two messages so as not to exceed the maximum GS1-128 Symbol message length. Each of the two messages has a Symbology Identifier prefix of "]C1" and does not exceed 48 data characters. The two messages are split at a boundary between two Element Strings. This mode is inferior to the normal transmission mode as message integrity may be lost when a message is split.