The Global Language of Business

GS1 Barcode Syntax Resource User Guide

Guidance on the GS1 Barcode Syntax Resource for the GS1 Community

Release 1.0.1, Final, Nov 2022

Document Summary

Document Item	Current Value
Document Name	GS1 Barcode Syntax Resource User Guide
Document Date	Nov 2022
Document Version	1.0
Document Issue	1
Document Status	Final
Document Description	Guidance on the GS1 Barcode Syntax Resource for the GS1 Community

Contributors

Name	Organisation
Terry Burton	Terry Burton Consulting Ltd
Kevin Dean	GS1 Canada
Amber Walls	GS1 US
Phil Archer	GS1 Global Office
Deirdre Courtney	GS1 Global Office
Peta Ding	GS1 Global Office
Steven Keddie	GS1 Global Office
Neil Piper	GS1 Global Office

Log of Changes

Release	Date of Change	Changed By	Summary of Change
1.0	Aug 2022	Peta Ding	Final edits
1.0.1	Nov 2022	Peta Ding	Errata fix in section 3.1

Disclaimer

THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR PARTICULAR PURPOSE, OR ANY WARRANTY OTHER WISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims all liability for any damages arising from use or misuse of this document, whether special, indirect, consequential, or compensatory damages, and including liability for infringement of any intellectual property rights, relating to use of information in or reliance upon this document.

GS1 retains the right to make changes to this document at any time, without notice. GS1 makes no warranty for the use of this document and assumes no responsibility for any errors which may appear in the document, nor does it make a commitment to update the information contained herein.

GS1 and the GS1 logo are registered trademarks of GS1 AISBL.

Table of Contents

1 Introduction

Users of GS1 standards rely heavily on data encoded in barcodes to enable various business processes that serve countless needs across the supply chain and beyond. For the past 50 plus years, a lot of these needs have been sufficiently met by 1D (one-dimensional or linear) barcodes.

Today, however, the landscape for trade and services, as well as the activities that span the entire value chain, have evolved significantly. Across both physical and digital environments, industry stakeholders and end-users alike require and expect larger amounts of data to facilitate numerous operational requirements, to improve efficiencies, and to make informed decisions.

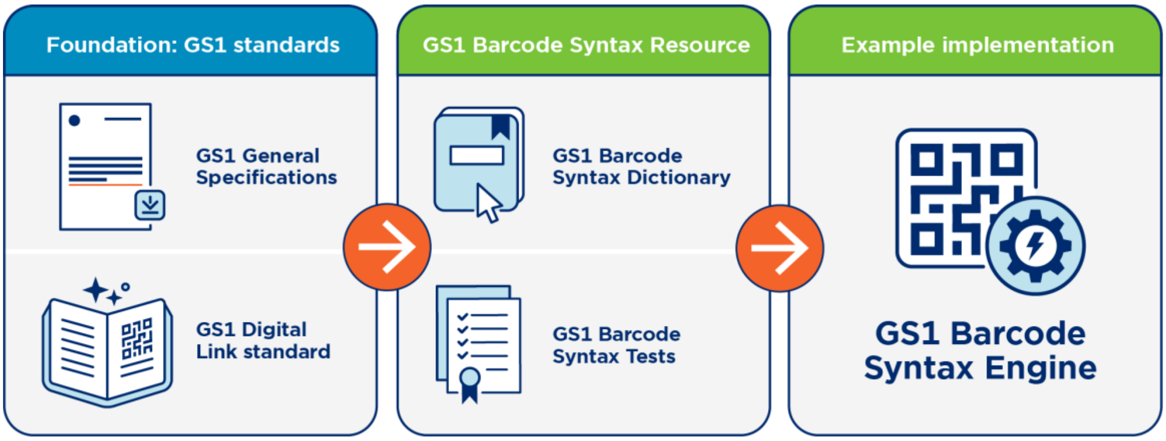
The global migration to 2D (two-dimensional or matrix) barcodes can fulfil many of these needs, as more data can be held in a significantly smaller space in comparison with 1D barcodes. However, the importance of high-quality data, which is trusted, timely and conformant to the specifications defined by GS1, cannot be underestimated. High-quality data that is structured and encoded in conformance with GS1 standards, is accessible in the right format, in the right barcode, and at the right time, enabling processes that are reliant on this information to operate seamlessly and efficiently. When data quality is poor and the data is not available as it’s required, or not available at all, processes, including those downstream, can be severely impacted.

The GS1 Barcode Syntax Resource is a collection of assets that enable the improvement of data quality for GS1 syntaxes encoded in 1D or 2D barcodes and provides the tools required by solution providers and users to correctly implement, apply and stay aligned to GS1 standards in a simple and consistent manner. The collection includes:

- GS1 Barcode Syntax Dictionary: contains details of the syntax requirements for GS1 Application Identifiers (AI), including how AI values may be decomposed into “component” parts.
- GS1 Barcode Syntax Tests: a set of routines that validate the different components of AI values.
- GS1 Barcode Syntax Engine: an example of the framework used to process AI syntax data using the GS1 Barcode Syntax Dictionary and GS1 Barcode Syntax Tests.

Through the adoption of the GS1 Barcode Syntax Resource, the GS1 community can easily create and process conformant GS1 syntaxes and work towards the global migration to 2D barcodes, no matter what their chosen solution may be and where they are on their 2D journey.

Figure 1-1 The relationship between GS1 standards and the GS1 Barcode Syntax Resource components


Foundation: GS1 standards
GS1 General Specifications
GS1 Digital Link standard
GS1 Barcode Syntax Resource
GS1 Barcode Syntax Dictionary
GS1 Barcode Syntax Tests
Example implementation
GS1 Barcode Syntax Engine

1.1 Background

When GS1 standards are interpreted incorrectly or implemented poorly, their value is diminished, as the application will not work as intended or simply will not work at all. Non-conformant GS1 data can lead to a lack of interoperability, causing cascading issues and inefficiencies.

However, the correct adoption and use of GS1 standards is not always easy or straight forward. There are usually multiple GS1 and ISO/IEC standards and guidelines to be used and understood.

Finding the correct references when developing a solution can be difficult and keeping systems aligned when GS1 standards are updated can be challenging.

If GS1 standards are difficult to implement, users may be inclined to limit their adoption (e.g., GTIN only and no other data), or seek non-GS1 solutions (e.g., proprietary identifiers). Additionally, when implementation difficulties lead to the misunderstanding and incorrect use of GS1 standards, especially during proof of concept or pilot phases, the full potential and return on investment may not be entirely realised or understood, which can also impact the possibility of scaled implementation. This is particularly important now as industries embark on the move to 2D barcodes to fulfil the need for additional data.

To ensure the value and purpose of GS1 standards is neither hindered nor limited for the GS1 community, the specifications must be easy to adopt, simple to comply with and convenient to stay aligned with, regardless of the user's requirement, choice of solution or intended application.

1.2 Impacted users and stakeholders

The GS1 Barcode Syntax Resource aims to benefit all users who are involved with barcodes approved for use in the GS1 system, including (but not limited to) those who:

- Create and print barcodes
- Manage data within barcodes
- Scan and process barcodes, and use the encoded data
- Create consumer experiences based on barcode interaction
- Build the systems that enable these activities

For industry stakeholders, such as brands, retailers, marketplaces, distributors, manufacturers, healthcare and service providers or transport and logistics operators, GS1 standards enable them to meet the ever-growing requirements for accurate and timely data. Whether that data is for improved inventory management, enhancing recall readiness, championing sustainability and ethical sourcing, building brand trust, addressing product authentication, improving patient and clinical safety outcomes or any other use case related to enriched information – the benefits of having the right data, whenever and wherever it's needed, is extensive across all markets and sectors. To implement GS1 standards at scale however, the entire ecosystem must have fit-for-purpose solutions, which can provide a stable environment for interoperability, no matter where and how business is conducted.

For solution providers (SPs), who are responsible for supplying the industry and the broader GS1 user community with these hardware and software solutions, as well as the updates required to transition from 1D to 2D, it can be cumbersome and confusing to integrate GS1 standards with their products and services correctly. This means that the solution may not necessarily work for the intended purpose or throughout all scan environments and may also be difficult to keep aligned for all users.

The GS1 Barcode Syntax Resource provides SPs with a format that is best suited to and understood by them, to integrate GS1 standards to their solution correctly, easily and consistently. This enables industry to experience a truly interoperable, conformant and standardised implementation of GS1 standards for all their operational data needs, as well as the resources needed to enable the global migration to 2D.

For GS1 Member Organisations (MOs), the GS1 Barcode Syntax Resource will enhance support for their communities as it provides a simple way to demonstrate "how" GS1 standards can be implemented correctly, removing the need for complex conversations and references to multiple GS1 standards and guidelines. Alongside the well-established explanations of "what" is needed, "why" and in some instances "when" it's needed, the GS1 Barcode Syntax Resource provides straightforward and actionable details for "how" GS1 General Specifications and GS1 Digital Link standards can be integrated to their solutions or projects.

Furthermore, it is expected that GS1 services such as GS1 Activate, the GS1 Registry Platform (GRP), the GS1 Application Identifier (AI) browser and check digit/character tools, will also leverage the GS1 Barcode Syntax Resource in the same way as SPs, through the development of wrappers to support other platforms in future, for the benefit of the entire GS1 community.

1.3 Summary

The GS1 Barcode Syntax Resource provides a harmonised framework for GS1 Application Identifier (AI) data validation and conversion, in accordance with the conformance specifications and rules defined by GS1 General Specifications and GS1 Digital Link standards. This framework offers a simple and streamlined approach to adopting GS1 standards, by enabling any project or solution that processes GS1 data syntaxes as used in 1D and 2D barcodes within the GS1 system, to implement the resource, either through direct integration to their application's code, or transliteration into a project's build system.

The GS1 Barcode Syntax Resource is available for use by free and proprietary software alike (under the Apache 2.0 licence), in line with the voluntary and royalty-free use of GS1 standards. As GS1 standards and the related conformance specifications or rules are updated with approved General Specifications Change Notifications (GSCN) through the Global Standards Management Process (GSMP), the resource will be updated by GS1 Global Office (GO), enabling solutions to integrate the updated components and remain aligned to changes, where relevant for the application.

2 Document purpose and scope

Although the GS1 Barcode Syntax Resource is intended for use by technical users such as software developers, this document aims to provide both technical and non-technical users with a clear explanation of the resources available for incorporation into their applications and processes. Technical users can also refer to the in-built technical documentation provided directly in the source files.

By empowering the GS1 user community with an understanding of the GS1 Barcode Syntax Resource, the tools can be shared within their organisations, as well as with their partners and/or their stakeholders, to facilitate the correct implementation and use of GS1 standards for all parties involved.

- **What is the GS1 Barcode Syntax Resource?**

  An explanation of the GS1 Barcode Syntax Resource and the different components that form the resource library.

- **How does the GS1 Barcode Syntax Resource work?**

  Each component of the GS1 Barcode Syntax Resource contains an explanation of how it works. The format of each component is also included to provide further context.

- **How can the GS1 Barcode Syntax Resource be used?**

  Explains how the GS1 Barcode Syntax Resource can be accessed, implemented and how to stay updated.

This user guide is not intended to be used as technical documentation for development or programming purposes, however, it may act as a bridge between non-technical and technical users of GS1 standards.

2.1 References to GS1 standards

Throughout this user guide, there are various references to “GS1 standards” or “GS1 conformance specifications and rules”. Both terms are used to collectively represent the following key sources of information:

- GS1 General Specifications: the core standards document of the GS1 system, which describes how GS1 barcodes and identification keys should be used.
- GS1 Digital Link URI standard: extends the power and flexibility of GS1 identifiers by making them part of the web, enabling the use of a single barcode for traditional purposes such as POS, as well as access to digital information about the entity being identified.

3 What is the GS1 Barcode Syntax Resource?

The GS1 Barcode Syntax Resource is a collection of assets that aims to help the GS1 user community to improve data quality and accurately convert between the different GS1 data syntaxes used with 1D and 2D barcodes in the GS1 system. By offering the tools required by solution providers to easily adopt and correctly implement GS1 standards, as well as a consistent way to stay aligned to GS1 conformance specifications and rules as they are updated, industry and the broader GS1 user community can leverage the full benefits of using any barcode or syntax from the GS1 system and work towards the global migration to 2D barcodes.

Consisting of the following three components, the GS1 Barcode Syntax Resource can be integrated directly to an application's code base or simply used as a reference for transliteration into third-party code as required by the solution's build system:

- GS1 Barcode Syntax Dictionary
- GS1 Barcode Syntax Tests
- GS1 Barcode Syntax Engine

Serving as a foundation that aims be to be helpful, but non-prescriptive, the GS1 Barcode Syntax Resource is a lightweight but versatile suite of tools that enables GS1 standards to be implemented easily and consistently for new and existing solutions. Depending on the user's requirements, the assets can be implemented in full, or in part, offering a foundation for application or user specific requirements to be built upon.

3.1 Supported data syntaxes

The GS1 Barcode Syntax Resource is simple and concise, for the purpose of facilitating the various procedures required to process, validate, and convert between different types of GS1 data syntaxes. This enables all GS1 data syntaxes to be backwards compatible, regardless of which syntax is required by the user or their solution.

The supported GS1 data syntaxes currently include the following which are approved for use in 1D or 2D barcodes within the GS1 system:

- **Plain syntax**: GS1 data structure containing a GS1 identification key with no additional characters or syntactic features. This format is used for EAN/UPC family barcodes, when only a GTIN is to be encoded, without any GS1 Application Identifiers (AI) or any other attribute data.

  ```
  9526064055028
  ```

- **GS1 element string syntax**: a syntax for expressing GS1 identifier keys and attributes using GS1 Application Identifiers (AIs) surrounded by parentheses to aid readability, for input or display within user interfaces e.g., barcode generator software; parentheses are NEVER encoded directly to the barcode.

  ```
  (01)09526064055028(17)250521(10)ABC123(21)456DEF
  ```

- **Barcode message format**: an unbracketed GS1 element string syntax that represents the GS1 element string syntax created by common barcode generation applications, for encoding directly to a barcode; the GS1 Barcode Syntax Resource represents the Function 1 Symbol Character (FNC1)[1] with a ^ symbol, however this can be represented differently by each solution.

  ```
  ^01095260640550281725052110ABC123^21456DEF
  ```

- **Barcode message scan data**: a format which is output by the barcode reader when a barcode is scanned; this includes the symbology identifier (e.g.,]d2 is for a GS1 DataMatrix) and group separator character (GS or ASCII 29).

  ```
  ]d2010952606405502817250521 10ABC123{GS}21456DEF
  ```

- **GS1 Digital Link URI syntax**: a web URI syntax for expressing GS1 identifier keys and attributes in a format using GS1 Application Identifiers and GS1 Application Identifier data fields as specified in the GS1 Digital Link standard. The "example.com" domain name is used to

[1] A Function 1 Symbol Character (FNC1) is a symbology character used in some GS1 data carriers for specific purposes

indicate that a GS1 Digital Link URI can be based on any internet domain name (GS1 recommends brand owners use their own if possible).

```
https://example.com/01/09526064055028/10/ABC123/21/456DEF?17=250521
```

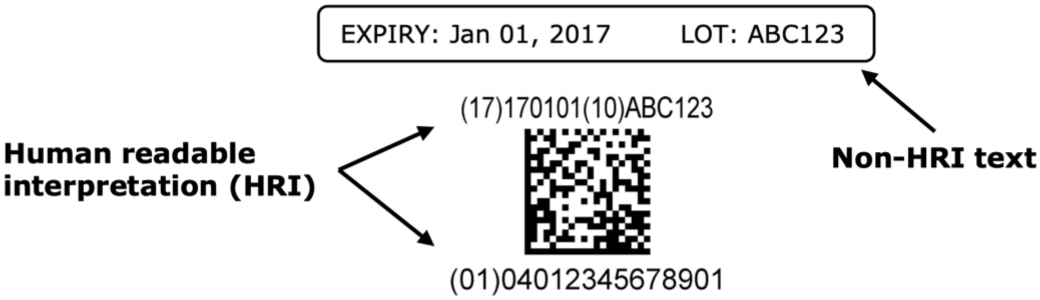
The applications which demonstrate the GS1 Barcode Syntax Engine present GS1 Digital Link URIs in canonical form, which includes the domain name “id.gs1.org”.

```
https://id.gs1.org/01/09526064055028/10/ABC123/21/456DEF?17=250521
```

- **Human readable interpretation (HRI)**: characters, such as letters and numbers, which can be read by a human and are encoded in GS1 AIDC data carriers confined to a GS1 standard structure and format. The HRI is the encoded data, shown with parentheses around each AI to aid readability, but they are NEVER encoded directly to the barcode. Start, stop, shift and function characters, as well as the symbol check character, are not shown in the HRI.
- **Non-HRI**: Characters such as letters and numbers that can be read by persons and may or may not be encoded in GS1 AIDC data carrier and are not confined to a structure and format based on GS1 standards (e.g., a date code expressed in a national format that could be used to encode a data field in a GS1 AIDC data carrier, brand owner name, consumer declarations etc.).

```
GTIN (01)09526064055028

EXPIRY (17)250521

BATCH/LOT (10)ABCDE123

SERIAL (21)123ABC456
```

Figure 3-1 Example of HRI and non-HRI text



3.2 GS1 Barcode Syntax Dictionary

The GS1 Barcode Syntax Dictionary is a simple text file that lists all currently assigned GS1 Application Identifiers (AI) and the components required to form a valid GS1 data syntax. Each AI can reference one or more GS1 Barcode Syntax Tests that enable the analysis of the different data components to check if it is valid against GS1 conformance specifications and rules for the GS1 data syntax.

As GS1 standards are updated or introduced, changes to AIs and any related conformance specifications or rules will be reflected within the GS1 Barcode Syntax Dictionary, by GS1 Subject Matter Experts (SMEs). In this way, users of GS1 standards can ensure their solutions or projects that process GS1 data syntaxes can easily stay up to date with each newly approved and published GSCN or GS1 standard, by adopting the refreshed GS1 Barcode Syntax Dictionary as it becomes available. Further information on staying up to date can be found in the section for How to utilise the GS1 Barcode Syntax Resource.

To access the GS1 Barcode Syntax Dictionary, please visit https://ref.gs1.org/tools/gs1-barcode-syntax-resource/syntax-dictionary/

3.2.1 How does the GS1 Barcode Syntax Dictionary work?

Each AI entry in the GS1 Barcode Syntax Dictionary references several different structural components of the required AI data syntax to assess if the data complies with the conformance specifications and rules for the referenced AI, as defined by GS1 standards:

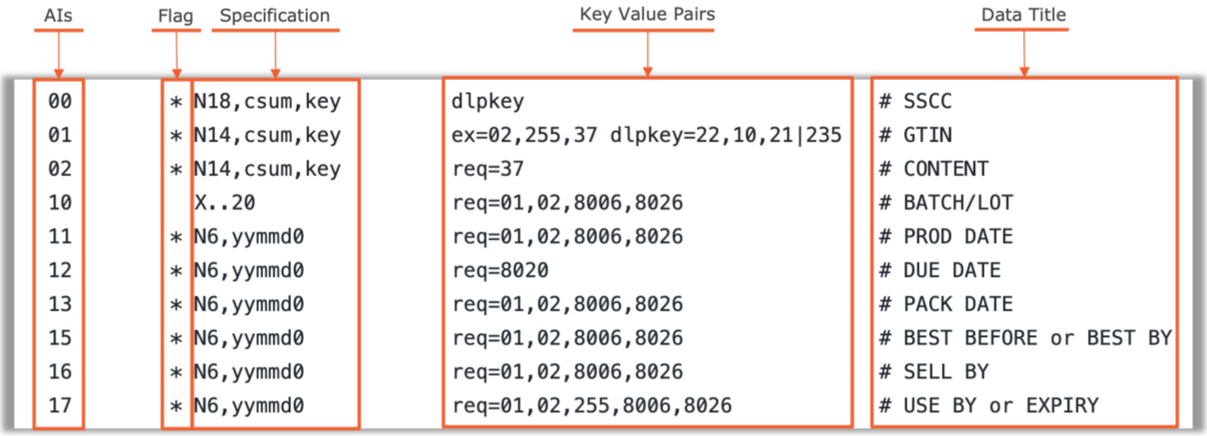
- GS1 AI Definitions (as defined by the GS1 General Specifications)
  - Data types: numeric or alphanumeric characters from the ISO/IEC 646 code set 82 or 39
  - Data format: field length and whether format is fixed or variable length
  - Data structure: requirements to include GS1 Company Prefixes (GCP) for primary keys, check digits/character pairs, data ranges (e.g., ISO country codes) or optional data components (e.g., serial component for GDTI)
  - Data relationships: mandatory or invalid pairs of AIs
- GS1 Digital Link URI syntax rules (as defined by the GS1 Digital Link URI standard)
  - Primary key AIs
  - Valid key qualifiers

The GS1 Barcode Syntax Dictionary provides the basis for implementing foundational GS1 conformance rules but is not intended to be a specification that captures all GS1 validation rules, nor does it include any application-specific rules. For example, when a GTIN AI (01) begins with 9, and it is designated as a variable measure trade item, it must be paired with AIs (30) OR (3nnn) OR (8001). These types of application specific rules require additional meaning for the AI value beyond its basic syntax and adds a level of coding intricacy that defies the scope of the tool. The GS1 Barcode Syntax Dictionary uses simple text declarations rather than code, to ensure it is purposefully compact and simple to enable adaptability and avoid any complexity that may be required for application-specific processing.

3.2.2 Format of the GS1 Barcode Syntax Dictionary

The GS1 Barcode Syntax Dictionary lists the numerical AI value, with segments detailing the required syntax, related GS1 Barcode Syntax Tests, its associations with other AIs and its GS1 Digital Link structure requirements (if present), as well as the AI data title. Each AI entry is read from left to right, following the order outlined in Figure 3-2.

Figure 3-2 Explanation of the segments contained in the GS1 Barcode Syntax Dictionary



```
00      * N18,csum,key     dlpkey                              # SSCC
01      * N14,csum,key     ex=02,255,37 dlpkey=22,10,21|235    # GTIN
02      * N14,csum,key     req=37                              # CONTENT
10        X..20            req=01,02,8006,8026                 # BATCH/LOT
11      * N6,yymmd0        req=01,02,8006,8026                 # PROD DATE
12      * N6,yymmd0        req=8020                            # DUE DATE
13      * N6,yymmd0        req=01,02,8006,8026                 # PACK DATE
15      * N6,yymmd0        req=01,02,8006,8026                 # BEST BEFORE or BEST BY
16      * N6,yymmd0        req=01,02,8006,8026                 # SELL BY
17      * N6,yymmd0        req=01,02,255,8006,8026             # USE BY or EXPIRY
```

AIs

Single (e.g., 01) or a range (e.g.,3100-3105) of AIs, listed in numerical order

Flag

“*” indicates a pre-defined length AI which does not require a FNC1 group separator character, as defined by GS1 standards. If no “*” is present, the AI requires a FNC1 group separator to indicate the end of the AI data field.

The FNC1 group separator character is represented as a ^ symbol when data is provided in the barcode message format:

```
^0109526064055028172505211OABC123^21456DEF
```

When data is provided in the barcode message scan data format the FNC1 group separator is represented as {GS}:

```
]d20109526064055028172505211OABC123{GS}21456DEF
```

Specification

Consists of multiple components, separated by “,”, which specify the character set, data format, and data structure required for the AI:

- Required character set:
  - "N": Numeric digits only
  - "X": Alphanumeric from GS1 encodable character set 82
  - "Y": Alphanumeric from GS1 encodable character set 39
- Required data format:
  - “N5” or “X5”: must be 5 digits or alphanumeric characters
  - “N..5” or “X..5”: between 1 and 5 digits or alphanumeric characters
  - “[N5]” : an optional component, if present, MUST be 5 digits
  - “[X..5]”: an optional component, between 1 and 5 alphanumeric characters
- Required structure:
  - csum: ensures data has a valid numeric check digit
  - yymmdd: ensures data represents a meaningful date in YYMMDD format
  - yymmd0: ensures data represents a meaningful date in YYMMDD format, additionally permitting “00” to represent an unspecified day
  - iso3166999: ensures data represents a 3-digit ISO 3166 country code, or the value “999”

Note: This is not a complete list of required data structures, please refer to the section for GS1 Barcode Syntax Tests for further information.

Key Value Pairs

Enables associations of AIs, to ensure mandatory or invalid AI pairs, including primary key and key qualifier sequences for GS1 Digital Link URI syntax, can be validated.

- “ex=02,255,37”: Invalid to pair with any of the listed AIs, and vice versa

  e.g., SHALL NOT be paired with AI (02), (255) or (37)
- “req=01,02,8006,8026”: Mandatory association with one of the listed AIs

  e.g., SHALL be paired with AI (01) or (02) or (8006) or (8026)
- “req=01,8006 req=21”: Mandatory association with one of the listed AIs, for each set of required AIs

  e.g., SHALL be paired with AI (01) or (8006) and SHALL also be paired with AI (21)

- "dlpkey": GS1 Digital Link primary key, with no qualifiers defined and accepted

  e.g., AI SHALL be present to form a valid GS1 Digital Link URI, with any additional AIs structured as data attributes only

- "dlpkey=22,10,21": GS1 Digital Link primary key, with ordered, optional qualifier AIs allowed

  e.g., AI SHALL be present to form a valid GS1 Digital Link URI, with AIs (22), (10) or (21) associated as key qualifiers, strictly in the given order

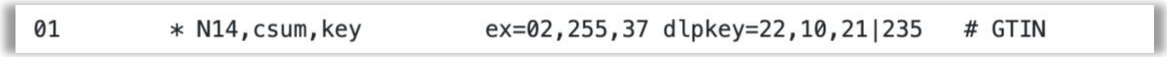
- "dlpkey=22,10,21|235": As above, but contains alternative sequences

  e.g., If AIs (22), (10) or (21) are associated as key qualifiers of the primary key, AI (235) SHALL NOT also be included as a key qualifier

Data Title

Preceded by a "#", data titles reflect the non-HRI text associated with each AI, as outlined by GS1 Application Identifier definitions.

Figure 3-3 Extract of the GS1 Barcode Syntax Dictionary entry for AI (01) with segments explained below



```
01        * N14,csum,key        ex=02,255,37 dlpkey=22,10,21|235   # GTIN
```

AI: 01 for Global Trade Item Number (GTIN)

Flag: * present, indicates a pre-defined length element string which does not require a FNC1 separator character

Specification: checks if data is 14 digits only, includes a valid check digit, and if the data string is correctly structured as a primary key that begins with a GS1 Company Prefix (GCP[2]).

Key value pairs: checks if data contains AIs (02) or (255) or (37) which are defined as invalid when combined with AI (01); and if the data string is a GS1 Digital Link URI syntax, a valid key-qualifier sequence is checked to ensure that AIs (22) or (10) or (21) are correctly represented as key qualifiers, in the correct order; and if AI (235) is present with AIs (22) and (10), it is represented as an attribute, rather than as a key qualifier when encoded alone with AI (01).

Data title: enables the non-HRI data title "GTIN" to be displayed if required by the user.

3.3 GS1 Barcode Syntax Tests

GS1 Barcode Syntax Tests are a set of files, written in the C programming language, a machine-independent type of code that provides instructions to perform a series of analytical actions. For software developers and programmers, GS1 Barcode Syntax Tests are formally known as "linters".

The actions described in each of the GS1 Barcode Syntax Tests referenced in the GS1 Barcode Syntax Dictionary are intended to check if data input by a user or an application is valid as per the GS1 conformance specifications and rules defined by GS1 standards. For context, these data strings may be input in a number of different ways. A user may input a data string to software, to generate a barcode to be placed on pack. The input data could also be a string transmitted from a barcode scanner. Or potentially, as a GS1 Digital Link URI input to a web application, from a QR Code scanned by a mobile device.

The manner in which data is input may vary, however the purpose of the GS1 Barcode Syntax Tests is to ensure data is validated so that it can be corrected, before being stored, communicated, or transformed into other GS1 data syntaxes. If any errors are present, information on the first piece of invalid data found can be returned to the user. This is the added value and deep validation provided by the GS1 Barcode Syntax Tests beyond the format specifications provided by the GS1 General Specifications.

[2] The GS1 Barcode Syntax Resource is not intended check if a GCP is valid in terms of active licencing. There is however capability to connect to an external database to enable this functionality, if required by the user's chosen application.

Example of errors caught by the component specification in the GS1 Barcode Syntax Dictionary

- Invalid field length – too long or short
- Invalid characters – non-numeric or alphanumeric from code set 39 or 82

Example of errors caught by the GS1 Barcode Syntax Tests

- Incorrect check digit or check character pair
- Invalid dates – month and days
- Invalid times – hours and minutes
- Invalid IBAN country code
- Invalid ISO codes
- Invalid coupon codes

To view the full list of GS1 Barcode Syntax Tests, please visit https://ref.gs1.org/tools/gs1-barcode-syntax-resource/syntax-tests

3.3.1 How do GS1 Barcode Syntax Tests work?

GS1 Barcode Syntax Tests along with the GS1 Barcode Syntax Dictionary, are intended for use within the framework of a solution that processes or generates GS1 data strings. The framework provides the necessary processes to extract the relevant AI entries from the GS1 Barcode Syntax Dictionary, which then enables each GS1 Barcode Syntax Test to validate each part of the GS1 data string that has been input. Further information on a possible framework, is explained in the section for the GS1 Barcode Syntax Engine.

The data is analysed in the following sequence:

1. GS1 Application Identifiers (AI) initial data field checks
   a. For each AI detected within the GS1 data string, the data is validated to firstly check it corresponds to a known AI, unless “Permit Unknown AIs” is enabled
   b. Next, overall field length is checked to validate that data length is within limits (i.e., not too long, or too short); for barcode message or scan data format, AIs are checked to ensure they are correctly terminated by a FNC1 group separator where required
   c. For GS1 Digital Link URIs, additional checks are performed to ensure that the extracted AIs form a valid primary key to key-qualifier sequence; and that the query parameter structure is correct
2. Further AI component checks:
   a. Validate conformance of the allowed character set
   b. Apply each GS1 Barcode Syntax Test in turn (reading left to right) – using (01) as example
      i. Validates check digit
      ii. Validates if the key starts with a GCP
      iii. Validates no invalid pairs of AIs are present
      iv. Validates if used as a DL primary key, then which optional key qualifiers are allowed (including any alternate XOR requirements)
   c. Validate overall AI associations
      i. Ensure repeated AIs have the same value
      ii. Ensure invalid pairs of attributes are not present
      iii. Ensure any mandatory pairs are present

3.3.2 Format of the GS1 Barcode Syntax Test

Each GS1 Barcode Syntax Test is an individual C source code routine that implements a series of analytical steps to evaluate the component data. The source files include in-built technical documentation and quality control mechanisms (unit tests), enabling developers to use the files directly in their solutions, or as a reference for transliteration into third-party code as required. The following assets are provided:

- The source files of each GS1 Barcode Syntax Test, which are directly referenced in the GS1 Barcode Syntax Dictionary
- A header file which contains the GS1 Barcode Syntax Test function declarations and definitions of each error code
- Optional implementations for mapping GS1 Barcode Syntax Test names to functions and error codes to error messages

Each GS1 Barcode Syntax Test source file contains the following segments of code and in-built technical documentation for support:

Brief

Offers a description of what the GS1 Barcode Syntax Test is checking and its purpose.

Remark

Some GS1 Barcode Syntax Tests include a note on the GS1 standard or external standard (e.g., ISO/IEC) for users to cross reference, if further information on the conformance specification or rule exists. In some instances, where the referenced rule may require maintenance such as an update, the remark also refers to 'maintenance notes' found directly in the code.

For example, the GS1 Barcode Syntax Test "lint_cset82" which checks that data only includes characters from the GS1 AI encodable character set 82 table, includes references to the GS1 General Specifications, however, does not include any maintenance notes, as the table will not change.

Whereas the GS1 Barcode Syntax Test "lint_iso3166" which checks that data represents a valid ISO 3166 "num-3" country code, includes a maintenance note to link users to any potential updates of the ISO 3166 country code list.

Additionally, GS1 Barcode Syntax Tests that refer to a universally fixed data range will not include any remark. For example, the GS1 Barcode Syntax Test "lint_yymmdd" which checks data represents a meaningful date, refers to the fixed ranges for months and days, so no cross reference is required.

Functions

This section contains the analytical functions of the GS1 Barcode Syntax Test used to validate the different parts of each data field, with lines of code that define the data parameters, commands for returns, API headers and in some instances, lookup headers. Additionally, every GS1 Barcode Syntax Test includes functions for various "unit tests" that act as a form of quality control, to ensure the integrity of the GS1 Barcode Syntax Test, in case the code needs to be altered for direct integration to a solution or adapted to proprietary codes or systems.

The segment for data parameters, enables the commencement of the GS1 Barcode Syntax Test and if invalid data is detected, depending on the nature of the error, these parameters can enable the position of the invalid data to be indicated, where it is meaningful to do so.

Return commands define what information is returned to the user, once the GS1 Barcode Syntax Test has detected either valid or invalid data. If invalid data is found, the return commands provide an error code which enables a description of the error to be offered to the user.

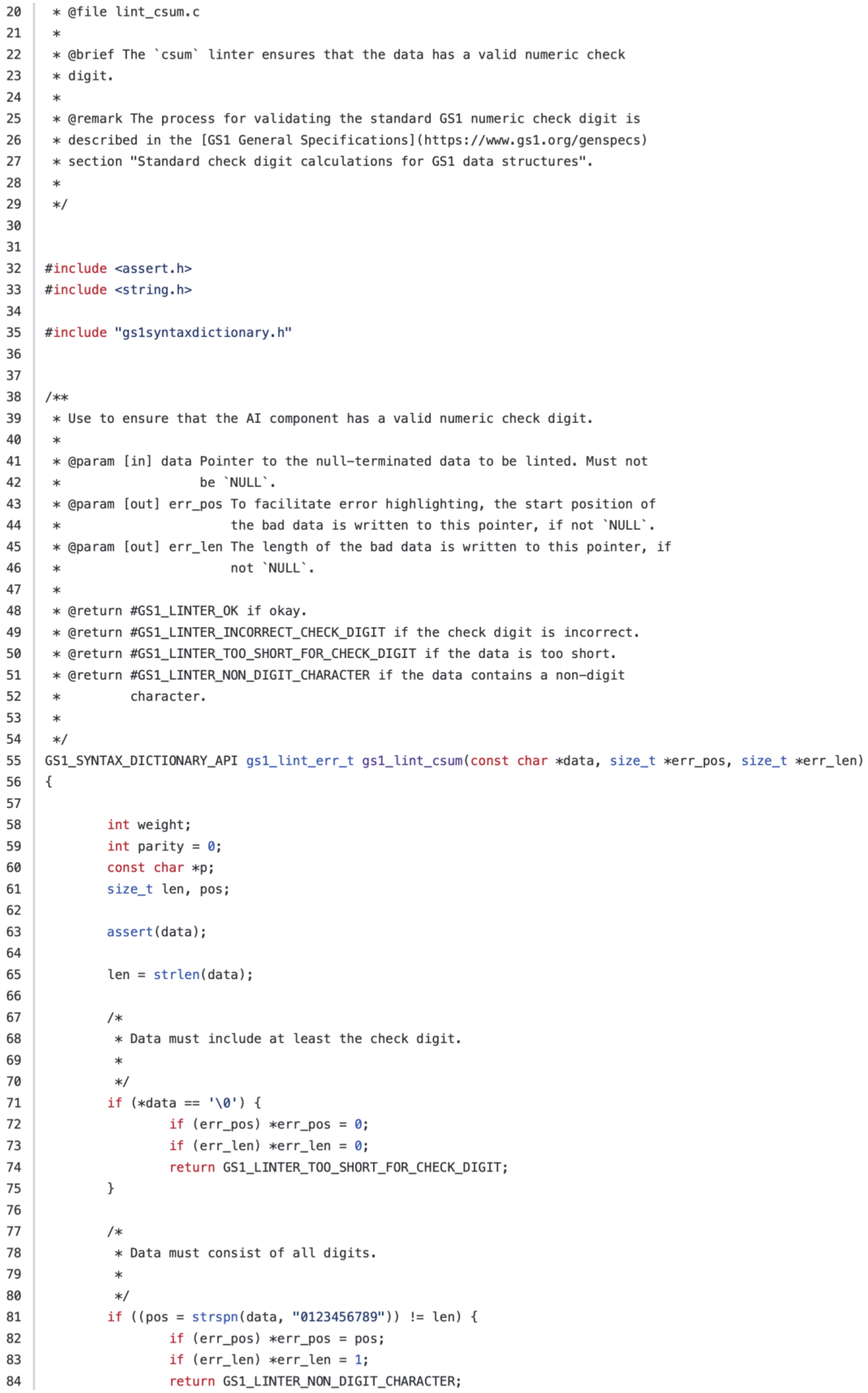
API headers enable the connection between the GS1 Barcode Syntax Dictionary and the GS1 Barcode Syntax Tests that are referenced for each AI entry.

Headers for lookup tables are included for some GS1 Barcode Syntax Tests, where users may have a requirement to connect to an external database. For example, the GS1 Barcode Syntax Test "lint_key" which checks if data contains a GCP to form a valid GS1 identification key, contains a lookup header that can enable an application to connect to an externally provided GCP database, if there is a requirement to check if a GCP is valid or is actively licensed.

Macro

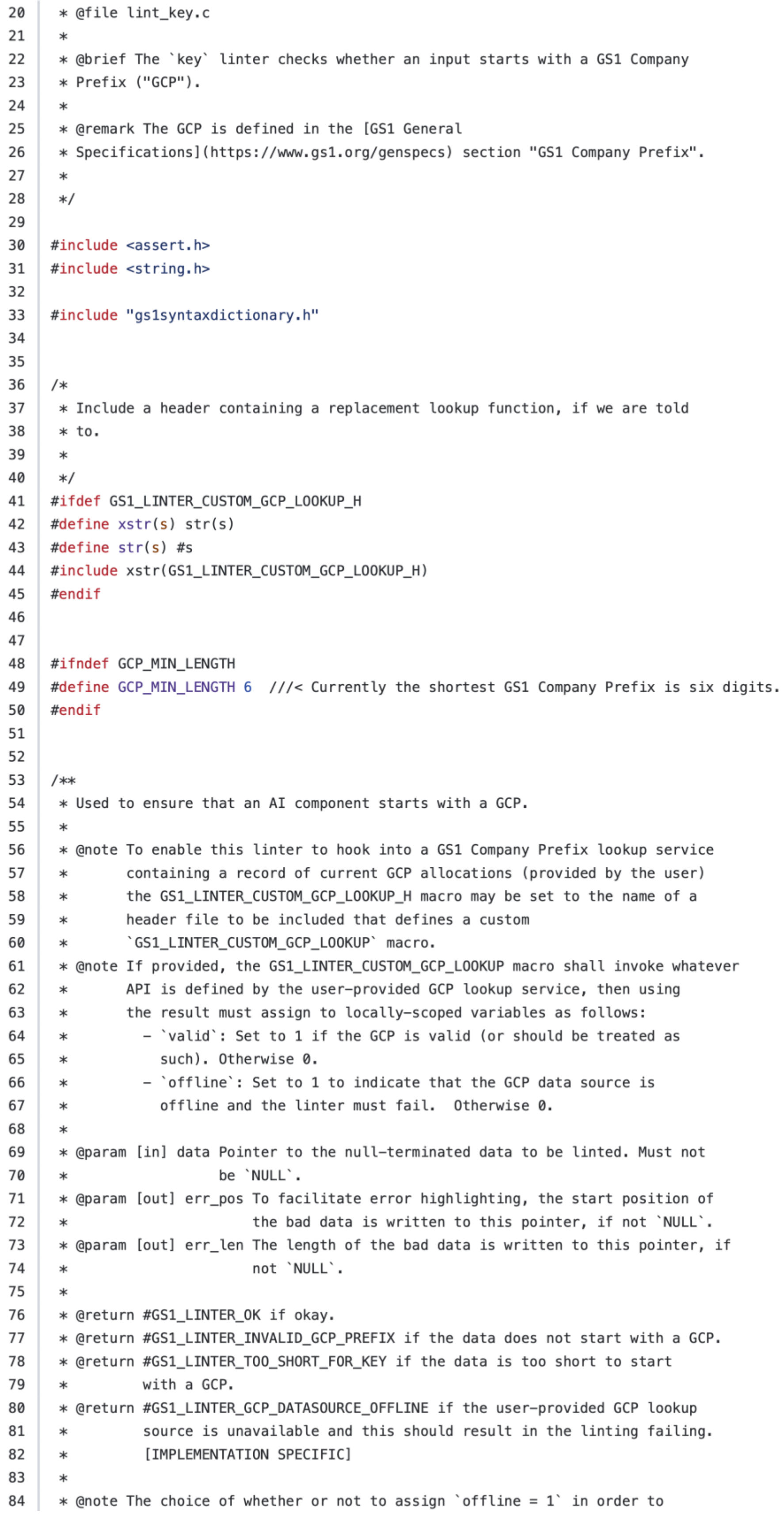
Some GS1 Barcode Syntax Tests define macros where additional context may be required for the purpose of further defining the data field.

Figure 3-4 Extract of the GS1 Barcode Syntax Test "lint_csum" which assesses if a valid check digit is present, as referenced for AI (01) in Figure 3-3



```
 * @file lint_csum.c
 *
 * @brief The `csum` linter ensures that the data has a valid numeric check
 * digit.
 *
 * @remark The process for validating the standard GS1 numeric check digit is
 * described in the [GS1 General Specifications](https://www.gs1.org/genspecs)
 * section "Standard check digit calculations for GS1 data structures".
 *
 */


#include <assert.h>
#include <string.h>

#include "gs1syntaxdictionary.h"


/**
 * Use to ensure that the AI component has a valid numeric check digit.
 *
 * @param [in] data Pointer to the null-terminated data to be linted. Must not
 *                  be `NULL`.
 * @param [out] err_pos To facilitate error highlighting, the start position of
 *                      the bad data is written to this pointer, if not `NULL`.
 * @param [out] err_len The length of the bad data is written to this pointer, if
 *                      not `NULL`.
 *
 * @return #GS1_LINTER_OK if okay.
 * @return #GS1_LINTER_INCORRECT_CHECK_DIGIT if the check digit is incorrect.
 * @return #GS1_LINTER_TOO_SHORT_FOR_CHECK_DIGIT if the data is too short.
 * @return #GS1_LINTER_NON_DIGIT_CHARACTER if the data contains a non-digit
 *         character.
 *
 */
GS1_SYNTAX_DICTIONARY_API gs1_lint_err_t gs1_lint_csum(const char *data, size_t *err_pos, size_t *err_len)
{

        int weight;
        int parity = 0;
        const char *p;
        size_t len, pos;

        assert(data);

        len = strlen(data);

        /*
         * Data must include at least the check digit.
         *
         */
        if (*data == '\0') {
                if (err_pos) *err_pos = 0;
                if (err_len) *err_len = 0;
                return GS1_LINTER_TOO_SHORT_FOR_CHECK_DIGIT;
        }

        /*
         * Data must consist of all digits.
         *
         */
        if ((pos = strspn(data, "0123456789")) != len) {
                if (err_pos) *err_pos = pos;
                if (err_len) *err_len = 1;
                return GS1_LINTER_NON_DIGIT_CHARACTER;
```

Figure 3-5 An extract of the GS1 Barcode Syntax Test “lint_key” which assesses if a GCP is present to form a valid GS1 identification key, as referenced for AI (01) in Figure 3-3



```c
 * @file lint_key.c
 *
 * @brief The `key` linter checks whether an input starts with a GS1 Company
 * Prefix ("GCP").
 *
 * @remark The GCP is defined in the [GS1 General
 * Specifications](https://www.gs1.org/genspecs) section "GS1 Company Prefix".
 *
 */

#include <assert.h>
#include <string.h>

#include "gs1syntaxdictionary.h"


/*
 * Include a header containing a replacement lookup function, if we are told
 * to.
 *
 */
#ifdef GS1_LINTER_CUSTOM_GCP_LOOKUP_H
#define xstr(s) str(s)
#define str(s) #s
#include xstr(GS1_LINTER_CUSTOM_GCP_LOOKUP_H)
#endif


#ifndef GCP_MIN_LENGTH
#define GCP_MIN_LENGTH 6  ///< Currently the shortest GS1 Company Prefix is six digits.
#endif


/**
 * Used to ensure that an AI component starts with a GCP.
 *
 * @note To enable this linter to hook into a GS1 Company Prefix lookup service
 *       containing a record of current GCP allocations (provided by the user)
 *       the GS1_LINTER_CUSTOM_GCP_LOOKUP_H macro may be set to the name of a
 *       header file to be included that defines a custom
 *       `GS1_LINTER_CUSTOM_GCP_LOOKUP` macro.
 * @note If provided, the GS1_LINTER_CUSTOM_GCP_LOOKUP macro shall invoke whatever
 *       API is defined by the user-provided GCP lookup service, then using
 *       the result must assign to locally-scoped variables as follows:
 *         - `valid`: Set to 1 if the GCP is valid (or should be treated as
 *           such). Otherwise 0.
 *         - `offline`: Set to 1 to indicate that the GCP data source is
 *           offline and the linter must fail.  Otherwise 0.
 *
 * @param [in] data Pointer to the null-terminated data to be linted. Must not
 *                  be `NULL`.
 * @param [out] err_pos To facilitate error highlighting, the start position of
 *                      the bad data is written to this pointer, if not `NULL`.
 * @param [out] err_len The length of the bad data is written to this pointer, if
 *                      not `NULL`.
 *
 * @return #GS1_LINTER_OK if okay.
 * @return #GS1_LINTER_INVALID_GCP_PREFIX if the data does not start with a GCP.
 * @return #GS1_LINTER_TOO_SHORT_FOR_KEY if the data is too short to start
 *         with a GCP.
 * @return #GS1_LINTER_GCP_DATASOURCE_OFFLINE if the user-provided GCP lookup
 *         source is unavailable and this should result in the linting failing.
 *         [IMPLEMENTATION SPECIFIC]
 *
 * @note The choice of whether or not to assign `offline = 1` in order to
```

3.4 GS1 Barcode Syntax Engine

The GS1 Barcode Syntax Engine is one possible implementation of the GS1 Barcode Syntax Dictionary and GS1 Barcode Syntax Tests, providing an example of the harmonised framework required to “run” the resources. As all GS1 data syntaxes are structured AI-based data formats (with the exception of plain syntax), the GS1 Barcode Syntax Engine provides a mechanism to facilitate the detection and conversion of GS1 data strings, whether provided as raw or human-friendly format, or as barcode message scan data format as received from barcode readers. This enables data to be structured so that it can be interpreted and understood consistently, regardless of which GS1 data syntax has been input or required for output.

Solutions may reference the GS1 Barcode Syntax Engine to ensure a solid, and consistent foundation for the processing of GS1 data syntaxes and a means of ensuring alignment to any updated GS1 standards today, tomorrow and well into the future. The GS1 Barcode Syntax Engine is also available as a graphical user interface (GUI) Demonstration Application and as a GS1 web tool, allowing users to test GS1 data strings to check if data is valid or invalid, and see how the data can be transformed into various alternative GS1 data syntaxes.

Whilst the demonstration application and web tool offer tangible representations of how the GS1 Barcode Syntax Resource can be implemented by solutions or projects that are required to process and validate GS1 data syntaxes, it is not intended to be the complete resource on offer to the GS1 user community.

To access the GS1 Barcode Syntax Engine please visit https://ref.gs1.org/tools/gs1-barcode-syntax-resource/syntax-engine/

3.4.1 How does the GS1 Barcode Syntax Engine work?

The GS1 Barcode Syntax Engine takes data input provided as any GS1 data syntax approved for use in 1D or 2D barcodes within the GS1 system, and extracts each AI for validation, using the GS1 Barcode Syntax Tests referenced in the GS1 Barcode Syntax Dictionary. AI and data fields are parsed for transformation into alternate GS1 data syntaxes and rendered to HRI text, as well as non-HRI data titles if required. The transformed data can then be used for encoding to a barcode, or output for further processing and end use. This enables GS1 data strings to be backwards compatible, no matter the intended application or chosen solution.

Data can be input as:

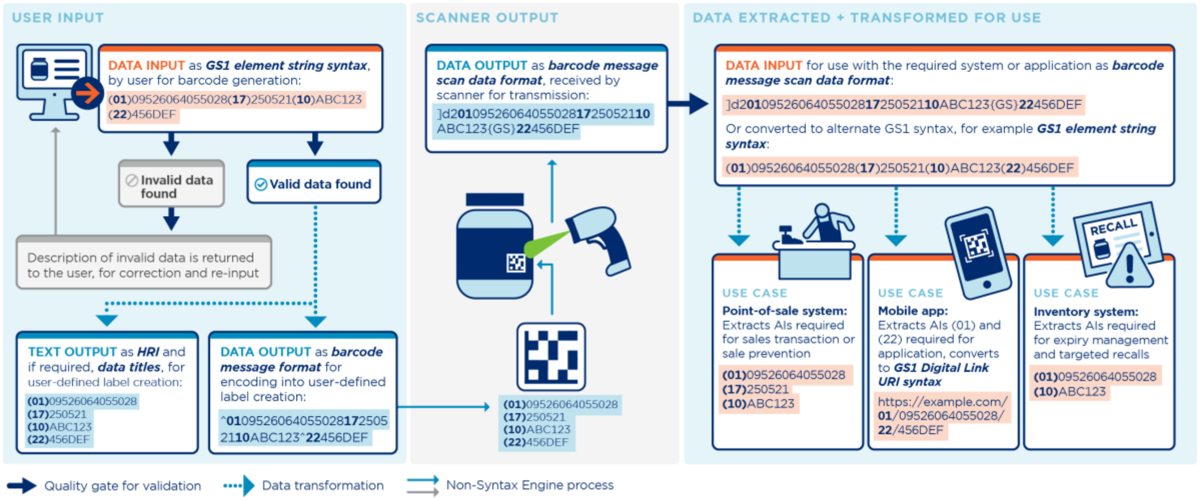
- Plain syntax
- Bracketed GS1 element string syntax
- Barcode message format / unbracketed GS1 element string syntax
- Scanned barcode message format
- GS1 Digital Link URI syntax

If the detected data syntax and all content is valid, according to the GS1 Barcode Syntax Dictionary and the referenced GS1 Barcode Syntax Tests, data is parsed, translated, and output into the following formats:

- Barcode message format / unbracketed GS1 element string syntax
- Bracketed GS1 element string syntax
- GS1 Digital Link URI (canonical form) syntax
- HRI text (with option to include non-HRI text as data titles)

Note: Plain syntax data is not directly translated, as it is not AI-based. However, as GS1 standards only allow a GTIN to be encoded as plain syntax, the AI (01) is added, for the purpose of conversion to alternative GS1 data syntaxes.

Figure 3-6 Example of how the GS1 Barcode Syntax Engine could be used to process GS1 data syntaxes


USER INPUT
DATA INPUT as *GS1 element string syntax*, by user for barcode generation:
(01)09526064055028(17)250521(10)ABC123(22)456DEF
Invalid data found
Valid data found
Description of invalid data is returned to the user, for correction and re-input
TEXT OUTPUT as *HRI* and if required, *data titles*, for user-defined label creation:
(01)09526064055028
(17)250521
(10)ABC123
(22)456DEF
DATA OUTPUT as *barcode message format* for encoding into user-defined label creation:
^0109526064055028172505 2110ABC123^22456DEF
SCANNER OUTPUT
DATA OUTPUT as *barcode message scan data format*, received by scanner for transmission:
]d2010952606405502817250521 10ABC123{GS}22456DEF
(01)09526064055028
(17)250521
(10)ABC123
(22)456DEF
DATA EXTRACTED + TRANSFORMED FOR USE
DATA INPUT for use with the required system or application as *barcode message scan data format*:
]d20109526064055028172505211 0ABC123{GS}22456DEF
Or converted to alternate GS1 syntax, for example *GS1 element string syntax*:
(01)09526064055028(17)250521(10)ABC123(22)456DEF
USE CASE
Point-of-sale system: Extracts AIs required for sales transaction or sale prevention
(01)09526064055028
(17)250521
(10)ABC123
USE CASE
Mobile app: Extracts AIs (01) and (22) required for application, converts to *GS1 Digital Link URI syntax*
https://example.com/01/09526064055028/22/456DEF
USE CASE
Inventory system: Extracts AIs required for expiry management and targeted recalls
(01)09526064055028
(10)ABC123
RECALL
Quality gate for validation
Data transformation
Non-Syntax Engine process

To demonstrate a real-world scenario, Figure 3-6 represents a several possible use cases where a product is marked with a GS1 DataMatrix and applied to product packaging for use at point-of-sale (POS), for inventory purposes (e.g., expiry management and targeted recalls), as well as for extended packaging or consumer engagement purposes.

With the GS1 Barcode Syntax Engine integrated to barcode generation software, data input by the user in the bracketed GS1 element string format is first validated against the GS1 conformance specifications and rules defined by GS1 Standards. If the input data string and any of the AI data is invalid, a description of the issue is returned to the user so that it can be corrected. If the data string and each AI is deemed valid, the AIs are transformed to the barcode message format (i.e., unbracketed GS1 element string format) and rendered with HRI, ready for encoding to the GS1 DataMatrix. This ensures that a compliant GS1 data syntax is present BEFORE it is encoded to a barcode.

Once the GS1 DataMatrix is applied to pack, the product makes its way out into the world, where it will need to be scanned at POS, for transactional purposes. The POS scanner is configured to read 2D barcodes and recognises the GS1 DataMatrix, enabling transmission of the encoded data in scan data format to be sent to the POS software. However, as linear barcodes encoded with only a GTIN in plain syntax are usually scanned at POS, the system is unable to understand the transmitted scan data. With the GS1 Barcode Syntax Engine integrated to the POS software though, the system is able to recognise the GS1 data syntax and extract the GTIN to perform the traditional price look up and complete the transaction.

To improve recall management, the retailer wants to enable a targeted recall process, as well as a hard stop at POS to prevent the sale of recalled or expired product. To enable the inventory management system to recognise batch/lot numbers, the GS1 Barcode Syntax Engine is adopted so that the system can parse the data read from the GS1 DataMatrix, including recognition of the FNC1 group separator to understand where the data ends. The POS system is also configured to parse the batch/lot and expiry date data when scanned at POS, so affected products are not sold.

For extended packaging or consumer engagement purposes, such as for product reviews, a consumer would likely scan the product's GS1 DataMatrix with their mobile device using a dedicated application. Using a fictitious example, a mobile application may exist for the purpose of providing product reviews and allowing users to provide reviews. The GS1 Barcode Syntax Engine could be integrated to the application, so that users could scan the barcode of any product and have the GS1 data syntax converted to a GS1 Digital Link URI so that the relevant product review webpage can be provided to the user. Reviews could possibly also utilise consumer product variant (CPV) data to display reviews to users based on current models or versions.

3.4.2 Format of the GS1 Barcode Syntax Engine

As with the GS1 Barcode Syntax Dictionary and GS1 Barcode Syntax Tests, the GS1 Barcode Syntax Engine is a collection of files that can be integrated into an application's open source or proprietary code base or simply redistributed with an application. Intended for use as an implementation framework by developers and programmers of solutions that process, validate, or generate GS1 data strings, the resources are provided in C and C# .NET, two programming language formats that can be easily adopted or referenced by solution provider community.

Below is a summary of the assets contained in the GS1 Barcode Syntax Engine repository:

Source files

- **Native C library**[3]: includes source code for the GS1 Barcode Syntax Engine that enables the detection and transformation of valid GS1 data syntaxes, including a brief description of any invalid data syntax issues detected. The C library also contains the GS1 Barcode Syntax Test files as explained in the previous section and the GS1 Barcode Syntax Dictionary text as well as a demonstration console application.
- **C# .NET**[4] **library**: contains the wrappers (also known as bindings) that provide an interface from .NET to execute the actions described by the native C-library.
- **C# .NET application**: contains a demonstration desktop application to provide an example of how to use the C# .NET library.
- **WebAssembly (WASM)**[5] **library**: contains a JavaScript (JS) Object Oriented wrapper that provides the sole interface to the WASM library, along with supporting JS and HTML files to provide a demonstration web browser application, compatible with all modern web browsers.

The demonstration applications are provided as:

- A console application to show how to use the native C library
- A desktop application to show how to use the C# .NET wrappers to access the native C library
- A web application to show how to use the WASM library for a browser-based application

Further information can be found in the section for Accessing the GS1 Barcode Syntax Resource.

Documents

Contains documentation for all public API of the GS1 Barcode Syntax Engine, including the API headers used for GS1 Barcode Syntax Tests.

4 How to utilise the GS1 Barcode Syntax Resource

The GS1 Barcode Syntax Resource can be utilised by directly integrating the assets within an application’s code base, or through transliteration into a solution’s build system. Solutions may choose to adopt the components in full or only the parts that relate to their service requirements. Regardless of the mode of integration chosen, the GS1 Barcode Syntax Resource can provide a solid foundation for creating and processing valid GS1 data syntaxes.

This section provides information on how the resources can be accessed, possible approaches for integration with a chosen solution, including guidance on where solution providers can reference further technical information, and lastly, how to stay updated.

4.1 Accessing the GS1 Barcode Syntax Resource

The GS1 Barcode Syntax Resource can be accessed from three different locations, depending on who requires access and for what purpose.

[3] A native C library provides macros, type definitions, and functions for tasks such as string validation, mathematical computations and processing or input/output data.

[4] C# .NET bindings are a programming process that establish a connection between an object and its function.

[5] WebAssembly is a type of code that can be run in modern web browsers.

For general, non-technical information, such as for introductory purposes or quick reference by the broader GS1 user community, the GS1 Barcode Syntax Resource can be accessed through the public GS1 website www.gs1.org, accompanied by a user-friendly online web tool. The online web tool allows users to test GS1 data strings and see any potential pieces of invalid data, providing a demonstration of one possible implementation of the resources.

- https://www.gs1.org/ (landing page currently under construction)

For permanent access to the assets, such as for linking on a public website (e.g., www.gs1.org), or for reference in a published document (e.g., GS1 General Specifications), the GS1 Barcode Syntax Resource should be accessed from GS1's persistent repository ref.gs1.org. To ensure the most recent version of the assets are referenced every time, the ref.gs1.org site offers a persistent link to the current version of the resources, as well as a stable record of all previous iterations.

- https://ref.gs1.org/tools/gs1-barcode-syntax-resource/

For technical access to the resources, such as for adoption purposes and to access the pre-built demo applications for implementation reference, the resources should be accessed directly from the GitHub repositories. As a platform widely used by the software development community, the assets are developed and maintained within GitHub, enabling technical users to reference and utilise the source code directly, for both testing and deployment purposes.

- https://github.com/gs1/gs1-syntax-dictionary
- https://github.com/gs1/gs1-syntax-engine

4.2 Integration with a chosen solution

Integration can be approached in several different ways, as the resource is intended to be non-prescriptive, enabling flexibility for pervasive adoption based on application specific needs. Below are some suggestions on possible integration approaches for the three different components.

For the GS1 Barcode Syntax Dictionary, integration could involve:

- Embedding the text file into the solution directly, to enable parsing when the application is run
- Using the solution's build system to extract the AI entries for inclusion in the application's static code

For the GS1 Barcode Syntax Tests, integration could involve:

- Building the framework and links to assimilate the source files directly into the application
- Using the files as a reference for re-implementation into the native language of the solution (including the unit tests)

For the GS1 Barcode Syntax Engine, integration could involve:

- Adopting the framework's source code directly into the application's source code
- Using the files as a reference to be rebuilt into the solution's native language

4.3 Staying updated and aligned

GS1 standards are regularly updated and developed to meet the evolving needs of the GS1 community. As GSCNs are approved through the GSMP and published publicly, the GS1 Barcode Syntax Resource will be updated to ensure alignment to the conformance specifications and rules defined by GS1 standards.

Solution providers and technical users can easily stay updated as all changes will be released through GitHub and tagged with version information for complete visibility of what's happened with each release, alongside supporting notes to provide a summary of the modifications. GitHub allows users to "watch" repositories, so they can be notified of the changes, enabling developers to review the code and initiate their solution's own quality assurance processes, before deciding to accept and merge the updated code for deployment. Solutions can also use "webhooks" to enable notifications to be delivered to an external server instead.

Updates will predominantly be made to the GS1 Barcode Syntax Dictionary, to include new AIs as they are approved for use in the GS1 system, or revised conformance specifications or rules, for existing AIs. Changes to GS1 Barcode Syntax Tests and the GS1 Barcode Syntax Engine are less likely but may still occur from time to time when necessary.

All iterations of the GS1 Barcode Syntax Resource provided on GS1 reference ref.gs1.org and surfaced on the public GS1 website www.gs1.org, will be pushed from the source files hosted on GitHub, ensuring consistency and ease of maintenance for all users, regardless of how the resources are accessed.