The Global Language of Business

GDSN Operations Manual Release 3.1

GDSN Version 3.1

Release GDSN Release 3.1, Issue 7, Apr 2017

Document Summary

Document Item	Current Value
Document Name	GDSN Operations Manual Release 3.1
Document Date	August 2017
Document Version	GDSN Release 3.1
Document Issue	7
Document Status	Issue
Document Description	GDSN Version 3.1

Contributors

Name	Organisation
GDSN Operations and Technology Advisory Group members	GS1

Log of Changes

Lists only changes between the latest Operations Manual 2.8 from August 2014 and the Major Release 3.1.

© 2018 GS1 AISBL

Release	Date of Change	Changed By	Summary of Change
3.1	August, 2015	Ewa Iwicka	• Section added: Document Conventions • Section 1: Introduction has been updated to refer to the new Major Release of the GDSN standard: 3.1 • Section 2.1: Updated list of GDSN messages and Trade Item Modules • Section 2.2: Simplified for improved readability • Section 2.2.2: Updated Party Dump document formats • Section 2.3.1.1: Added reference to the new set of 3.1 Validation Rules • Section added: 2.3.1.2 Context and the Population of Attributes • Section 2.3.5: Updated list of Catalogue Item states • Section 4.1: Simplified for improved readability • Section added: 4.2 GS1 XML Architecture for GDSN Major Release 3 • Section 4.3 (previously 4.2): Updated URL to hosted standards • Section 4.4 (previously 4.3): Updated URL to GS1 Registry • Section 4.5 (previously 4.4): Updated information about namespaces and referencing SBDH in 3.1 instance documents • Section 6.4: Updated way of SBDH use in release 3.1, added new rules previously covered only in Validation Rules • Section 6.5: Updated response message names • Section 6.6.1: Updated Entity Identification structure • Section 6.7: Updated Transaction, Commands and Document structure • Section 6.8: Updated Content Owner structure • Section 6.9: Updated Language Code example • Section 6.10: Corrected example of incorrect DateTime • Section 6.11: Updated message and attribute names • Section added: 6.12: Deleting Published Catalogue Item • Section added: 6.13.1 Module ordering • Section added: 6.14: Custom extensions • Section added: 6.17: Code list versioning • Section added: 6.18: Populating code data for code lists not defined by GS1 • All sections: Conformant statements updated according to the GS1 Document Conventions
3.1 i3	August 2016	Ewa Iwicka	• Section 2.3.2 Added in new modules for release 3.1.2 • Section 6.6.1 Updated entity identification examples. • Section 6.10 Removed reference to use of language code together with country code. Only ISO 639-1 codes are allowed. Examples updated accordingly. • Section 6.13.2 Eliminated the reference to aspectRatioInformation. • Section 6.18 Changed the way of versioning externally managed code lists. • Section added: 6.21: Populating AVP component
3.1 i4	March 2017		• Section added: 4.7 XML Schema Validation in GDSN • Section 6.11.1: Corrected time offset expression examples • Section added: 6.11.2 Time Precision • Section added: 6.12 Country and Country Subdivision format • Section 6.15: Specified the only allowed method of deleting published Catalogue Item, in alignment with new Validation Rules.
3.1 i5	May 2017 June 2017		• Section 6.11.1: Specified attributes which the time offset should be used for. Specified single way of expressing time zone offset. • Added an explicit statement that the time zone offset is optional in ALL GDSN attributes. • Example of business attributes made neutral (concrete example removed).
3.1 i6	April 2018		• Added to Section 4.3 links to both the Beta and Production Environments.

© 2018 GS1 AISBL

Disclaimer

GS1 ®, under its IP Policy, seeks to avoid uncertainty regarding intellectual property claims by requiring the participants in the Work Group that developed this **GDSN Operations Manual Release 3.1** to agree to grant to GS1 members a royalty-free licence or a RAND licence to Necessary Claims, as that term is defined in the GS1 IP Policy. Furthermore, attention is drawn to the possibility that an implementation of one or more features of this Specification may be the subject of a patent or other intellectual property right that does not involve a Necessary Claim. Any such patent or other intellectual property right is not subject to the licencing obligations of GS1. Moreover, the agreement to grant licences provided under the GS1 IP Policy does not include IP rights and any claims of third parties who were not participants in the Work Group.

Accordingly, GS1 recommends that any organization developing an implementation designed to be in conformance with this Specification should determine whether there are any patents that may encompass a specific implementation that the organisation is developing in compliance with the Specification and whether a licence under a patent or other intellectual property right is needed. Such a determination of a need for licencing should be made in view of the details of the specific system designed by the organisation in consultation with their own patent counsel.

THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGMENT, 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 Standard, 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

© 2018 GS1 AISBL

Document conventions

The keywords, SHALL, SHALL NOT, and MAY, when they appear in this document written in CAPITAL LETTERS, are to be interpreted as described in in Annex G of the ISO/IEC Directives, Part 2, 2001, 4th edition, as defined here:

- **SHALL** means that all conforming implementations must do what the statement says, otherwise the implementation is not conforming. No deviation is permitted.
- **SHALL NOT** means that all conforming implementations must not do what the statement prohibits, otherwise the implementation is not conforming. No deviation is permitted.
- **MAY** (or CAN) means that a conformation implementation is allowed to do what the statement says, but it is not required to for conformance.
- **MAY NOT** means that an implementation is not required to do what the statement says and still remain conformant.

© 2018 GS1 AISBL

1 Introduction

This GS1 GDSN (Global Data Synchronisation Network) Operations Manual is a guide for implementers of the Network. It is a 'living document' that is periodically updated on as needed basis. This implementer's guide should be referenced along with the Business Message Standards (BMS) defined within the GS1 Global Standards Management Process (GSMP). The BMS's are the normative documents – they specify the standards; how to implement the standards is described in this Operations Manual. The authors and editors of the Operations Manual are responsible for ensuring that there are no discrepancies between the normative documents and this manual. Because the standards for GDSN are dynamic and implementation learnings continually result in modifications to the standards, gaps will arise between the documented standards and the learnings forged by implementations. The applicable implementation learnings are being captured in the Operations Manual, being the reference point collecting these decisions reached by the GDSN group before these make their way as standards in the BMS.

1.1 Purpose of this Document

The purpose of this document is to explain how to use the GS1 Standards in the Global Data Synchronisation Network (GDSN). It answers the questions – *"what do I need to do to build a standards-compliant implementation?"* Other GDSN documents provide an overview of GDSN or the business requirements behind GDSN and this document serves as an Operations Manual to help the GDSN user and implementers' community. Its content will be expanded and will change as the needs of the users change. This document and updated versions of this document will be located and can be downloaded from: *http://www.gs1.org/gdsn-operations-manual-31*.

1.2 Audience

This document should be of use to implementers of the Global Data Synchronisation Network (GS1 Global Registry, GDSN Data Pools, and GDSN Trading Partners). This document is intended for business persons and developers who will implement GDSN from the operations guide and the standards as published in the BMS. In addition, the document is intended for the participants in the standards development process for GDSN.

1.3 GDSN Standards and documentation

The following link gives access to documents providing additional background and relevant information: *http://www.gs1.org/gdsn*.

2 GDSN Standards

GDSN Trading Partners, Data Pools and the GS1 Global Registry interact with each other through standard interface messages. These standard messages, their UML class diagrams and use cases are located in the Business Message Standard (BMS) documents. Class diagrams are transformed into W3C XML Schema Definition Language schemas. The business intent is preserved from the BMS through the schema realization, although demands of XML syntax, cross- business process consistency and component reuse can cause minor deviations between the BMS document and schema.

A trade item is any trade item (product or service) upon which there is a need to retrieve pre-defined information and that may be priced, or ordered, or invoiced at any point in any supply chain. Trade Items are identified with a GTIN - one of the keys of the GS1 System. The term "Trade Item" can represent any level of product containment. There is a difference between a Trade Item and a "Catalogue Item". The Trade Item embodies the representation of the definition of a trade item and may or may not include hierarchy structure. The Trade Item key is a GTIN. The Trade Item is registered by the GTIN Owner and allows define specific non-changeable attributes.

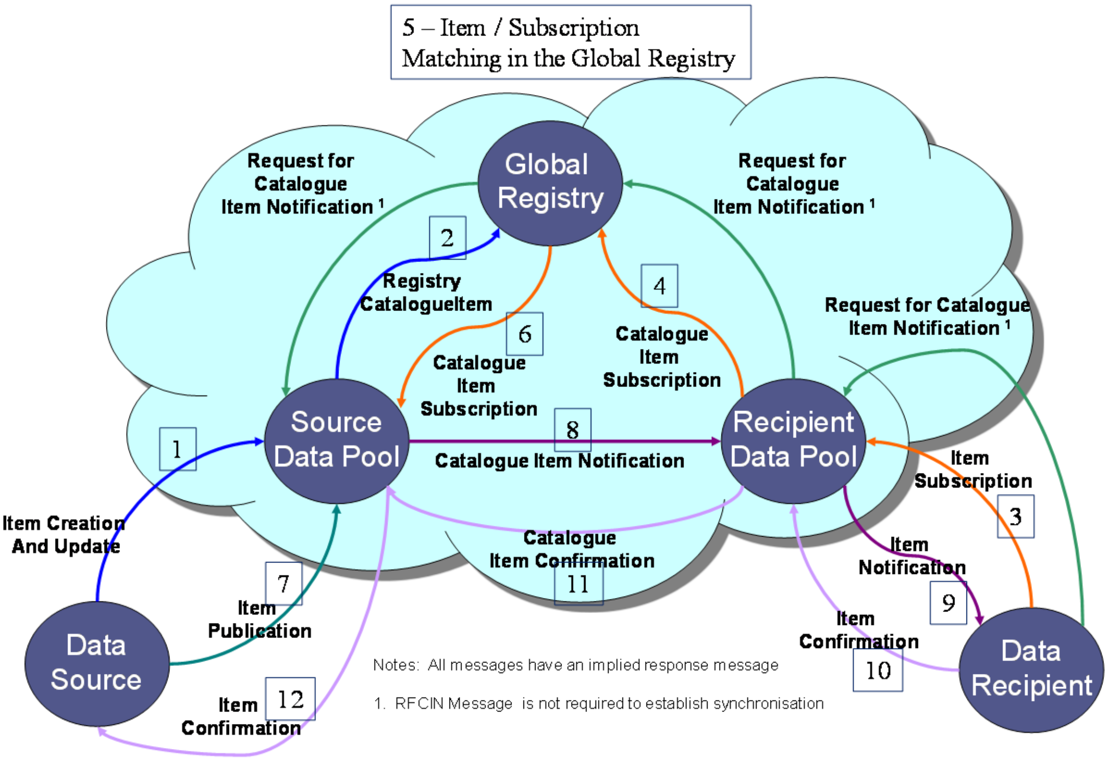
The Catalogue Item embodies the representation of the usage of an item, or in other words the content of an item. A Catalogue Item is uniquely defined within the GS1 Global Registry by the combination of the item's Global Trade Identification Number (GTIN), Information Provider Global Location Number (GLN), and Target Market. It is registered by the Information Provider who may or may not be equivalent to the GTIN Brand Owner and allows for the flexibility for Information Providers to allow for varying data across GLN's and/or Target Markets. The Target Market (TM) is a 3-digit numerical code that indicates the country level or higher geographical definition in which the information provider will make the item available to buyers. This indicator does not govern where a buyer may re-sell the item to consumers. The Target Market Country Code is taken from the ISO 3166-1 list. Target Market could also include a subdivision code. The Target Market Subdivision Code is taken from the ISO 3166-2 list. Each Trade Item is treated as a separate and independent item within the GS1 Global Registry.

(For more details on the functionality please see the BMS.)

2.1 Global Data Synchronisation (GDSN) Data Flow

Figure 2-1 depicts the overall flow of all the different messages in the GDSN among the entities. These messages may vary depending upon the sender and receiver of the messages. Some messages are designed as multi-hop messages and can possibly flow through different parts of the network.

Figure 2-1 Full GDSN Choreography


5 – Item / Subscription Matching in the Global Registry
Global Registry
Request for Catalogue Item Notification 1
Request for Catalogue Item Notification 1
Request for Catalogue Item Notification 1
2
Registry CatalogueItem
4
6
Catalogue Item Subscription
Catalogue Item Subscription
Source Data Pool
Recipient Data Pool
1
8
Catalogue Item Notification
Item Subscription
3
Item Creation And Update
Catalogue Item Confirmation
11
7
Item Publication
Item Notification
9
Item Confirmation
10
Data Source
Data Recipient
Item Confirmation
12

Notes: All messages have an implied response message

1. RFCIN Message is not required to establish synchronisation

© 2018 GS1 AISBL

2.2 Basic Party Synchronisation Choreography

The Basic Party Synchronisation process is a required prerequisite to the Catalogue Item Synchronisation process and is used by all Data Pools to register Parties (GLN's) in the Global Registry. In the Global Registry, Parties are identified by individual GLN's registered by a Data Pool. GDSN Trading Partners require Party data information to achieve data synchronization. A Party must be registered in the Global Registry in order to participate in any GDSN functionality, including the registration of items or subscriptions. All Data Pools (either Source or Recipient) must register Party information on behalf of its Trading Partners. The process for this registration is:

- The Data Pool's Trading Partner communicates its Party information to the Data Pool in a mutually agreed format.
- The Data Pool uses this information and creates the standard Party Registration message as defined in the Basic Party Registration process. This message SHALL pass all validations in place for the registration of Parties.
- The Global Registry returns the "Party Registration Response" message with an applicable responseStatusCode, which value depends on whether the registration was successful or not. If the Party was successfully registered, the GS1 Global Registry will store the Party information, including the Data Pool information identifying responsible Data Pool for that Party.

2.2.1 Party Dump

There is a batch process in Global Registry running on a daily basis that generates a file containing all the information for all of the Parties in the Global Registry, including such information as GLN, Party Name, Address, Contact Name, Country, Country Code, etc. These data of all Parties in the GDSN is sent in the format of text file to all Certified Data Pools in the GDSN for use in synchronising the Party information in the GDSN. The XML version of this file can be obtained under the following URL: *http://www.gs1globalregistry.net/eanucc/dumps/partydump/*

2.2.2 GDSN Actors

Below are the functions of the actors in the GDSN - Global Registry, Data Source, Source Data Pool, Recipient Data Pool, and Data Recipient:

GS1 Global Registry

- Enables the registration and distribution of Party information, identifying the actors and roles
- Enables the registration all the Item information through a small set of core information
  - GTIN, GLN of the information Provider, Target Market, and the GPC
- Provides Validation Services to ensure uniqueness
- Enables the registration all the Item subscriptions with a small set of criteria
  - GTIN, GLN of the information Provider, Target Market, and the GPC
- Performs the Item / Subscription matching process at the core of the GDSN choreography

Data Source

- Typically a Manufacturer/Distributor
- Maintains trade item information that it wants entered into the GDSN
- Registers trade item information in a Source Data Pool to be registered with the Global Registry and sent to a Recipient Data Pool
- Sends trade item information in any format agreed by the Data Source and the Source Data Pool

Data Recipient

- Typically a Retailer, Hospital, Group Purchasing Organization, Distributor or any other User of Data
- Subscribes to trade item information by the any of the following combinations of criteria:
  - Item (GTIN)
  - Party (GLN)
  - Target Market
  - GPC Brick
- Receives trade item information in any agreed-to format with Recipient Data Pool

© 2018 GS1 AISBL

Source Data Pool

- SHALL validate Item Information against the GDSN Validation Rules (Mandatory)
- Receives trade item information from Data Sources to be registered
- Uses standard XML Messages to register the item information in the Global Registry
- Uses GS1 standard XML Messages to exchange trade item information with the Recipient Data Pool (& Data Recipient)

Recipient Data Pool

- MAY validate Item Information against the GDSN Validation Rules (Optional)
- Receives subscriptions from its Data Recipients using criteria
- Registers subscriptions in the Global Registry
- Receives item information from the Source Data Pool, including new and updated
- Provides the trade item information from the Source Data Pool to the Data Recipient

The significance of the cloud is that inside the cloud represents the “in-network” portion of the GDSN. Inside the cloud, all of the exchange of the messages is strictly defined by the GDSN standards. All Data Pools must use the same messages in the same exact way. Out of the network (outside the cloud), the way that the Data Pools communicate with their trading partners (Data Sources and Data Recipients) is up to the Data Pools. This flexibility allows for the Data Pools to create value-add offerings for their customers. For example, item information can be communicated to the Data Pools in additional formats such as excel files, text files, existing Electronic Data Interchange (EDI) messages. Data Pools can translate these formats into the standards-based XML messages for use in the network and can translate when messages are received through the network.

2.2.3 Use of XSD-based Schema XML Messages

All in-network transactions between a data pools and from the Data Pools to the Global Registry SHALL use the GDSN standard XML messages. The GDSN uses a set of standardised XSD-based, XML messages. Figure 2-2 shows the existing set of GDSN-related standards

Figure 2-2 GDSN Standards

Catalogue Item Synchronisation
Basic Party Synchronisation
Item Authorisation
GDSN Common Library
GDSN Price Synchronisation
GDSN Trade Item
GDSN Trade Item Modules

- Alcohol Information Module
- Allergen Information Module
- Animal Feeding Module
- Apparel Information Module
- Audience or Player Information Module
- Audio Visual Media Content Information Module
- Audio Visual Media Product Description Module
- Audio Visual Media Production Information Module
- Award Prize Module
- Battery Information Module
- Certification Information Module
- Chemical Regulation Information Module
- Onyx Publication File Information Module
- Organism Classification Module
- Packaging Information Module
- Packaging Marking Module
- Packaging Sustainability Module
- Pharmaceutical Item Information Module
- Physical Resource Usage Information Module
- Place of Item Activity Module
- Plumbing HVAC Pipe Information Module
- Product Characteristics Module
- Promotional Item Information Module
- Propellant Information Module
- Publication Title Rating Module
- Referenced File Detail Information Module

- BMS Consumer Instructions Module
- Controlled Substance Information Module
- Copywrite Information Module
- Dairy Fish Meat Poultry Item Module
- Dangerous Substance Information Module
- Delivery Purchasing Information Module
- Diet Information Module
- Durable Goods Characteristics Module
- Duty Fee Tax Information Module
- Electronic Device Characteristics Information Module
- Farming and Processing Information Module
- Food and Beverage Ingredient Module
- Food and Beverage Preparation Serving Module
- Food and Beverage Properties Information Module
- Health Related Information Module
- Health Wellness Packaging Marking Module
- Healthcare Item Information Module
- Marketing Information Module
- Medical Device Trade Item Module
- Movie Revenue Information Module
- Nonfood Ingredient Module
- NonGTIN Logistics Unit Information Module
- Nutritional Information Module
- Regulated Trade Item Module
- Safety Data Sheet Module
- Sales Information Module
- Security Tag Information Module
- Software System Requirements Module
- Sustainability Module
- Textile Material Module
- Trade Item Data Carrier and Identification Module
- Trade Item Description Module
- Trade Item Disposal Information Module
- Trade Item Handling Module
- Trade Item Hierarchy Module
- Trade Item Humidity Information Module
- Trade Item Licensing Module
- Trade Item Lifespan Module
- Trade Item Measurements Module
- Trade Item Size Module
- Trade Item Temperature Information Module
- Transportation Hazardous Classification Module
- Variable Trade Item Information Module
- Video Display Device Information Module
- Warranty Information Module

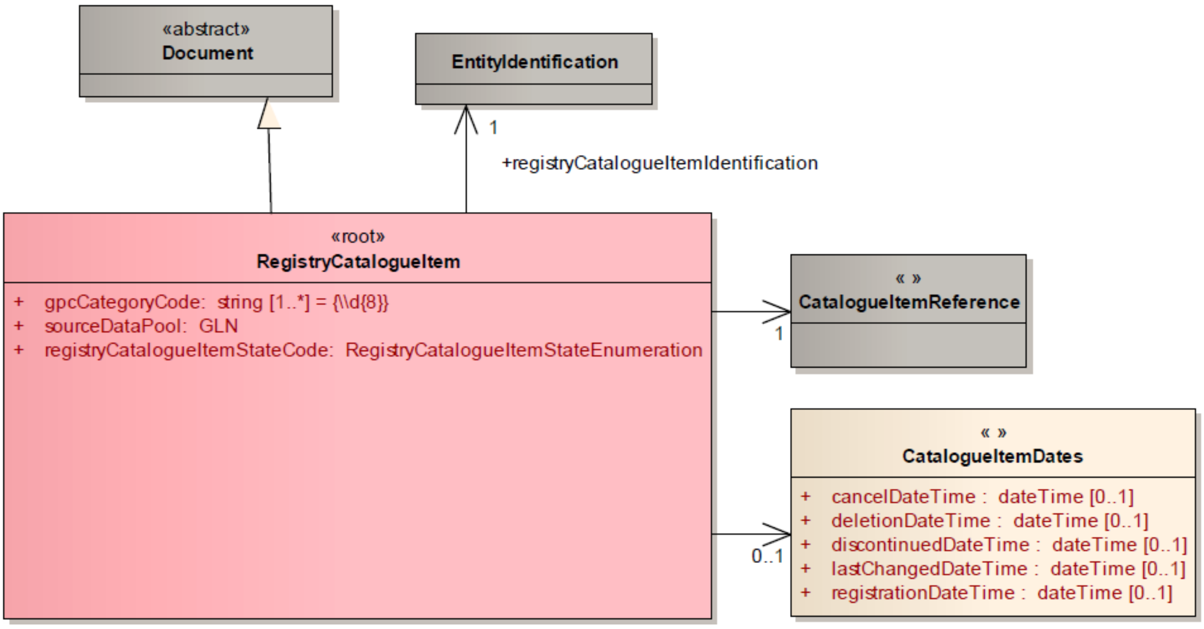
The most significant standard for the GSDN is the “Catalogue Item Synchronisation”. It is this standard that is the most implemented one in the GDSN. The GDSN Trade Item Modules are special extensions to the Trade Item that add additional information specific to a particular industry or business need – the business context.

Within the Catalogue Item Synchronisation Process, there are a number of standard messages:

- Catalogue Item Confirmation
- Catalogue Item Hierarchical Withdrawal
- Catalogue Item Notification
- Catalogue Item Publication
- Catalogue Item Registration Response
- Catalogue Item Subscription
- Registry Catalogue Item
- Request for Catalogue Item Notification
- GS1 Response

All of the messages are defined in the standard and are created using Universal Modelling Language (UML) models. These models are then used in the creation of the XML messages. An example of a model for the Registry Catalogue Item message is shown in Figure 2-3.

Figure 2-3 UML Model of the Registry Catalogue Item Message


«abstract»
Document
EntityIdentification
1
+registryCatalogueItemIdentification
«root»
RegistryCatalogueItem
+ gpcCategoryCode: string [1..*] = {\\d{8}}
+ sourceDataPool: GLN
+ registryCatalogueItemStateCode: RegistryCatalogueItemStateEnumeration
« »
CatalogueItemReference
1
« »
CatalogueItemDates
+ cancelDateTime : dateTime [0..1]
+ deletionDateTime : dateTime [0..1]
+ discontinuedDateTime : dateTime [0..1]
+ lastChangedDateTime : dateTime [0..1]
+ registrationDateTime : dateTime [0..1]
0..1

2.2.4 Communication Transfer Protocol

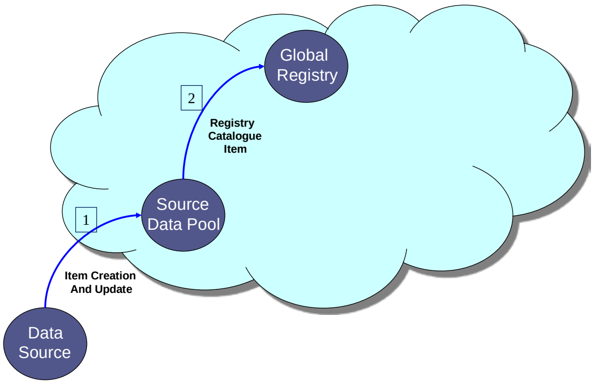
All messages communicated ‘in network’ between the Data Pools themselves or between the Data Pool and the Global Registry are sent over the internet using the EDIINT AS2 communication protocol. This protocol is used for the exchange of information in a decentralised, distributed environment and was designed and codified by the Internet Engineering Task Force (IETF). AS2 uses the Hyper Text Transfer Protocol (HTTP) to build a tunnel to the recipient address, establish the connection, and then send the information in a secured environment assuring the sender of the receipt. The use of digital certificates is also implemented in the GDSN, so that the only entities that can communicate with the Global Registry are the GDSN-Certified Data Pools.

2.3 Catalogue Item Synchronisation and Global Registry Functionality

2.3.1 Registry Catalogue Item (RCI) Registration

The Registry Catalogue Item is the business message used to register trade item information from a data pool to the Global Registry within the GDSN Network. The process for this is shown in Figure 2-4.

Figure 2-4 Item Registration Process



1. First a Data Source loads the full set of item information (which may include values for as many attributes about the Trade Item as needed by their customers and the business processes which will eventually use the data).
2. From this information the Source Data Pool forms a Registry Catalogue Item message using the primary information of the:
   a. GTIN
   b. GLN of the Information Provider
   c. Target Market of where the GTIN is intended to be sold
   d. Global Product Classification (GPC) Brick Code

      - and -

   e. A Status and relevant dates (creation date, last update date)
   f. The Data Pool that is responsible for the Trading Partner and its items
3. This information is registered in the Global Registry if it passes validation and uniqueness checks.

The Registry Catalogue Item message is defined very strictly by the standards and all Data Pools SHALL use the same messages and validations. The method by which the Data Source creates the items in the Source Data Pool depends upon their choice of data pools and their mutual agreement. For example, it is possible to use a graphical user interface to create and register information. This can also be done using existing EDI message sets that are not related to GDSN.

One of the important aspects of the GDSN processes is that at the Source Data Pool, the Items that are to be communicated are usually arranged in product hierarchies. This is a method how items are physically packaged to contain each successive level of the hierarchy. An example would be that a can of soda exists by itself. However, in the supply chain, the can is usually packaged in packs of 6 cans (a six-pack), which are in turn packaged in 1 case of 4 six-packs. These cases may then be packaged together on a pallet of 60 cases. This would be an example of an item hierarchy. It is also possible to have the same can of soda packaged as a 12-pack, with two 12-packs per case, and 40 cases per pallet. So it is possible that the same can of soda can be contained within multiple hierarchies. It is important to note, only the items themselves are registered in the Global Registry. In the above examples, the can of soda, the six-packs of soda, the 12-packs of soda, the cases of soda and the pallets of soda would all be registered individually in the Global Registry, each identified by its own unique GTIN. However, the Global Registry would not know how these levels are pieced together to create the individual hierarchies. That function is performed at the Source Data Pool.

2.3.1.1 Validation Rules

Validation Rules are used in the GDSN to enforce business rules associated with Parties, Items, and Prices. They allow some validations that are likely to change occur outside of the schemas. Thanks to that, validations can be modified without having to implement a whole new set of schemas for each change.

The latest version of Validation Rules can be accessed here: *http://www.gs1.org/gdsn/gdsn-validation-rules/3-1*.

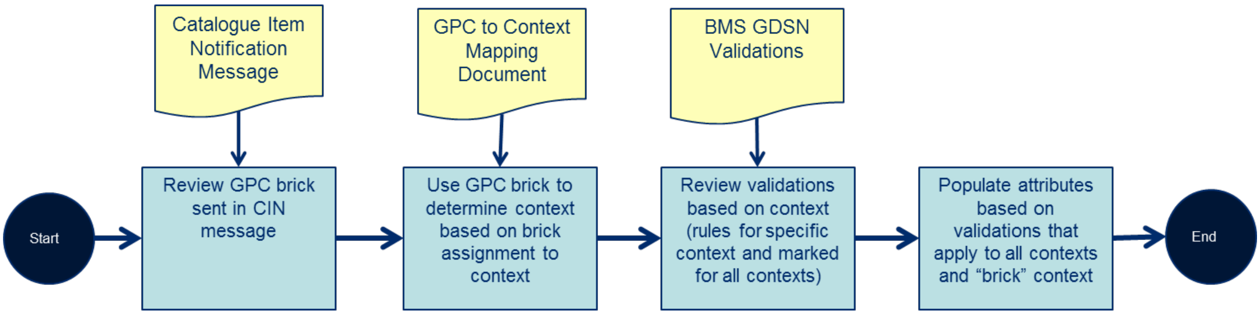
The validation rules performed on a CIN are driven by the product context of an item. Please refer to contexts in the GDSN Validation Rules BMS to see the context associated with your item and the relevant validation rules. Please note that validations for an item include both the list for All Contexts and the specific context of the trade item (based on GPC Brick).

2.3.1.2 Context and the Population of Attributes

Release 3.1 introduces the concept of context which allows for specific validations to be run based on the Process Context currently limited to Distributing Product Information (DPI) and Product Context (a grouping of similar products, for example Food, Beverage, Tobacco and Pet Food). While the CIN has attributes used to populate context for a trade item (contextIdentification) this attribute is not populated by the Data Source. Instead, the context for a trade item will be populated by the Source Data Pool based on the brick sent in the gpcCategoryCode attribute within the CIN. The GPC to Context Mapping document is used to map the GPC brick to the correct context. Population of the contextIdentification attribute for this release is informational only.

The context is then used to determine the correct validations to be run for a trade item. Some validations relate to all contexts while other validations relate to specific contexts only. The validations run for a CIN will be the ones that apply to all contexts and the ones that apply to the specific context related to the GPC brick for the trade item.

GS1 provides a mapping of modules per context (Modules by Context) but this document is not normative. Instead, the modules that need to be populated are driven by the validation rules that apply to the trade item. The process for determining the attributes that need to be sent for an item and thus the modules that need to be sent is summarized in the following figure:

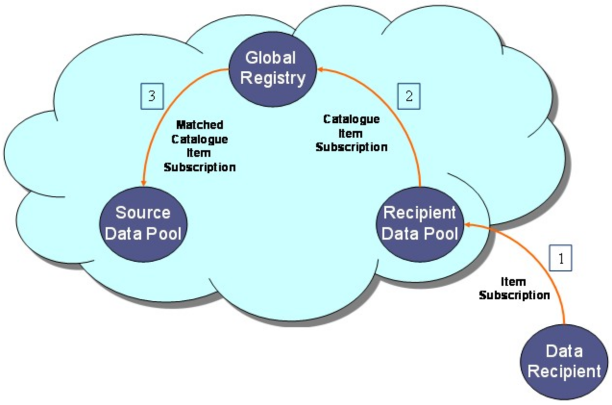

Catalogue Item Notification Message
GPC to Context Mapping Document
BMS GDSN Validations
Start
Review GPC brick sent in CIN message
Use GPC brick to determine context based on brick assignment to context
Review validations based on context (rules for specific context and marked for all contexts)
Populate attributes based on validations that apply to all contexts and "brick" context
End

Please note that contextIdentification also exists at the component level. It is currently assumed that context will not be populated at the component level (ComponentInformation/contextIdentification).

2.3.2 Catalogue Item Subscription (CIS) Registration

The Catalogue Item Subscription is the business message used to communicate a request made by a Data Recipient for Trade Item information, with the intention of synchronising this data with the Data Source. The process for this is shown in Figure 2-5.

Figure 2-5 Catalogue Item Subscription Registration Process



1. First a Data Recipient loads the criteria for the subscription to the item information.
2. From this information the Recipient Data Pool forms a Catalogue Item Subscription message using the criteria of:
   a. GTIN
   b. GLN of the Information Provider
   c. Target Market of where the GTIN is intended to be sold
   d. Global Product Classification (GPC) Brick Code

   and -

   e. A Status and a few relevant dates (creation date, last maintained date)
   f. The Data Pool that is responsible for the Trading Partner and its subscriptions
3. This information is registered in the Global Registry if it passes validations
4. The Global Registry Matching Process is initiated

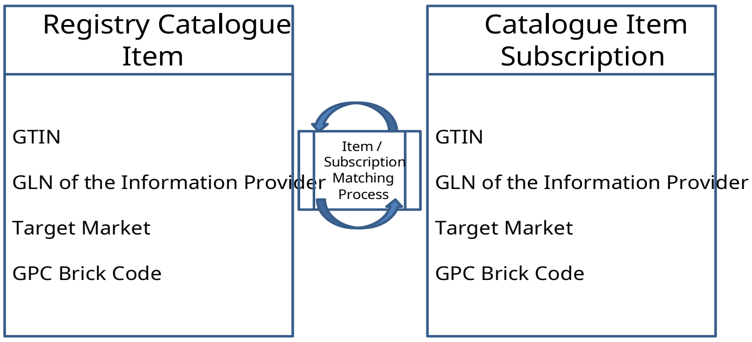
The Catalogue Item Subscription message is defined by the standards and all Data Pools must use the same messages and validations. The method by which the Data Recipient creates subscriptions or the Data Source receives subscriptions depends on their choice of data pools and their mutual agreement. For example, it is possible to use a graphical user interface to create subscriptions and receive information as well.

2.3.3 Global Registry Matching Process

The Global Registry then performs a matching process. Whenever a Registry Catalogue Item registration or a Catalogue Item Subscription occurs, the Global Registry performs a matching process to compare items against subscriptions and vice-versa, using the common criteria for both registrations as shown in Figure 2-6.

© 2018 GS1 AISBL

Figure 2-6 Global Registry Item / Subscription Matching Process

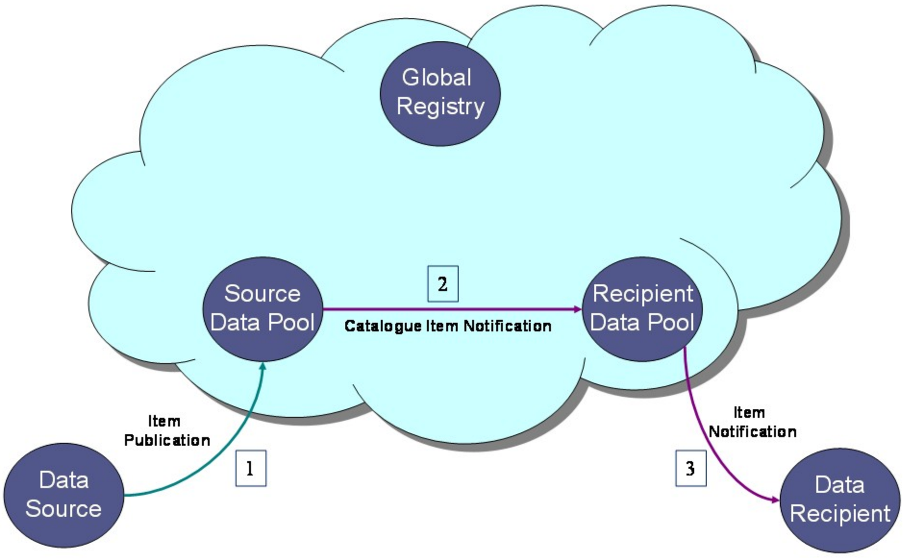

Registry Catalogue Item
GTIN
GLN of the Information Provider
Target Market
GPC Brick Code
Item / Subscription Matching Process
Catalogue Item Subscription
GTIN
GLN of the Information Provider
Target Market
GPC Brick Code

The result of a successful match is that the Global Registry sends this matched Catalogue Item Subscription to each and every Data Pool that has an Item registered that matches that specific subscription criteria.

2.3.4 Item Publication and Distribution

Once the Global Registry completes its matching process and sends the matched subscriptions out to the appropriate Source Data Pools, the Source Data Pools must match the Item Publications to the received subscriptions. When a Data Source publishes its item information, it creates a Catalogue Item Notification containing all the information about the item, including the whole hierarchy of products. It also adds this entry to its synchronisation list. This synchronisation list contains all items that have been published, the result of the synchronisation process, whether or not the Data Recipient is interested in receiving automatic updates and any future changes. This Catalogue Item Notification is sent directly from the Source Data Pool to the Recipient Data Pool. It does not go through the Global Registry, which has facilitated the connection but is not involved in the actual message exchange - as shown in Figure 2-7.

Figure 2-7 Item Data Distribution Process



As previously mentioned, the in-network message, the Catalogue Item Notification is defined by the standards and all Data Pools SHALL use the same messages and validations. The method by which the Data Source Publishes or the Data Recipient receives depends on their choice of data pools and their mutual agreement. For example, it is possible to use a graphical user interface to publish information and receive information as well. This MAY also be done using existing EDI message sets that are not related to GDSN.

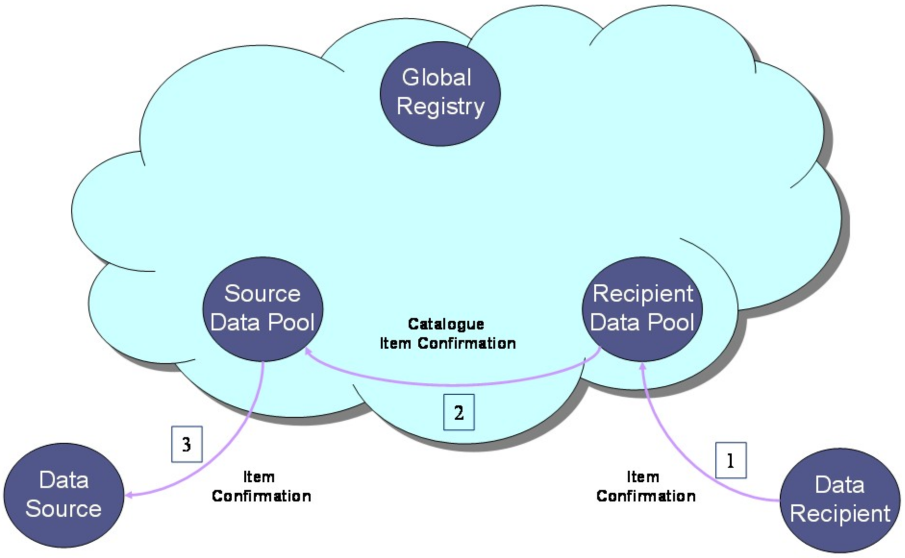
Once a Data Recipient receives the item information, they can decide whether or not they want to continue to synchronise data with this Data Source. This decision is based on a number of factors related to the business relationship between the Data Recipient and the Data Source. Once this decision is made, it SHALL be communicated through the GDSN back to the Data Source.

2.3.5 Catalogue Item Confirmation

When the Data Recipient makes the decision, the communication back to the Data Source is created and the Recipient Data Pool creates a message called a Catalogue Item Confirmation, as shown in Figure 2-8.

© 2018 GS1 AISBL

Figure 2-8 Catalogue Item Confirmation Process



This message contains the business decision on whether or not the Data Recipient will continue to synchronise the item information. The possible states that can be communicated in the message are:

- **Synchronised:** Data is integrated, synchronised and added to the synchronisation list
- **Received:** Data has been received by the Recipient, but no business decision has been made on the data.
- **Rejected:** Data will no longer be synchronised nor will updates be provided
- **Review:** A request to the data source to "review" their data because the data recipient has received discrepant data which they cannot synchronise.

If no confirmation is received, data updates will continue to be provided until the data recipient responds with a "rejected" state.

In addition to these Catalogue Item Confirmation States, if the data is put into a state of "reviewed" or "rejected", the Data Recipient MAY respond back with a very specific set of responses that would indicate what exactly is wrong with the data and what process should be used to fix the problem and resend the correct information. The Catalogue Item Confirmation Status Enumeration list provides a code value and a description of what is wrong. The Corrective Action Code provides the list of correction actions in a Catalogue Item Confirmation response.

When the Source Data Pool receives this Catalogue Item Confirmation, it can update the synchronisation list and pass on the information to the Source Data Pool. This closes the loop and completes the data synchronisation process for the initial synchronisation of data. With the information contained in the synchronisation list, the Source Data Pool knows which Data Pools (and Data Recipients) will be sent updates to the item data that has already been synchronised. This way, the item data always remains as up to date as possible.

2.3.6 Request for Catalogue Item Notification (RFCIN)

A Request for Catalogue Item Notification (RFCIN) functions very much like a normal Catalogue Item Subscription. However, this is a one-time request for the data to be (re)sent. The request for notification is not stored in the Global Registry after the matching process occurs. The RFCIN is only executed once and then discarded by the source data pool.

© 2018 GS1 AISBL

For Catalogue Items that were previously synchronised (e.g. in the synchronisation list) or rejected, the request for notification resets the confirmation status – it removes the catalogue item from synchronisation list. The primary use for an RFCIN is to recover from a catastrophic failure of an internal system, where the item information must be rebuilt in that system.

3 Joining GDSN

Trading partners wishing to participate in the GDSN should contact the GS1 Member Organisation (MO) in the country in which their company is headquartered. Data pools SHALL be certified to be a part of the GDSN. All interested parties should visit the GS1 GDSN website for instructions on how to become members at *http://www.gs1.org/gdsn*. The same page contains a link to the list of certified data pools.

3.1 Data Pool On-boarding

Upon making the decision to join the GDSN, a new data pool SHALL contact the GDSN to start the Registration process. A new data pool will need to establish a main contact for all GDSN communications. The data pool will also need to prepare information such as the URL for the data pool for all the regions (environments), the IP address, Digital Certificate information, and contact information. A ticket should be entered into the tracking system for GDSN for the data pool with all relevant information.

The Data Pools and the Global Registry are subject to adhere to the Terms of the GDSN Service Level Agreement [SLA] as well as the GDSN Acceptable Use Policy [AUL]. Both documents are available through GDSN.

3.2 GDSN Certification

All GDSN participating data pools SHALL be certified. Certification criteria and test plans are based upon the approved standard and GS1 GDSN released functionality. The certification agent develops certification tests based upon criteria developed and approved by the GDSN during the GDSN Release Materials Development Phase.

Upon successful completion of the certification event, the data pool will then be entered into the GR System(s) by the GDSN Technical Support team after GDSN Inc. approval.

The data pool would then be able to begin testing in the Test / Beta Environment.

3.3 Change Management

The scope and timeline of any new GDSN release is based upon the GDSN road map. The GDSN road map presents a high level overview of the development priorities of the GDSN User Group.

Based on the priorities outlined in the GDSN road map and the availability of resources, the GDSN User Group can determine the number and types (major or minor) of releases scheduled during the year. The release scope and plan also includes the Version of Global Product Classification which is adopted for the specific GDSN release as well as the adopted versions of externally managed code lists.

✓ **Note:** GDSN, GSMP and GPC release version numbers may not be in sync and can diverge over time.

Also, some of the XML Schema files might be at the different version than the overall GDSN release.

4 Use of XML in GS1 and GDSN

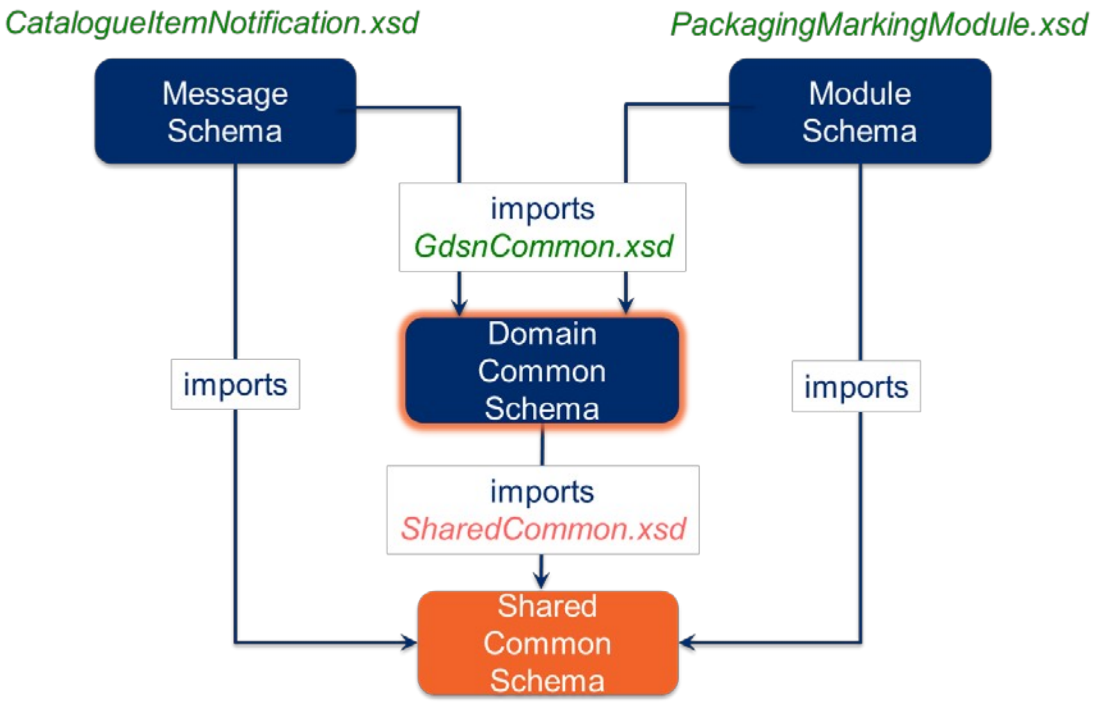
4.1 GS1 XML Standards

GS1 XML Standards are developed within the Global Standard Management Process (GSMP) by the GS1 user companies, GS1 Member Organisations and the GS1 Global Office.

The standards provide a flexible and extensible approach for transacting business-to-business communication. They have multi-sector and global applicability. GS1 XML Standards are based on the World Wide Web Consortium (W3C) XML specifications.

4.2 GS1 XML Architecture for GDSN Major Release 3

Figure 4-1 Schema relationships in Major Release 3

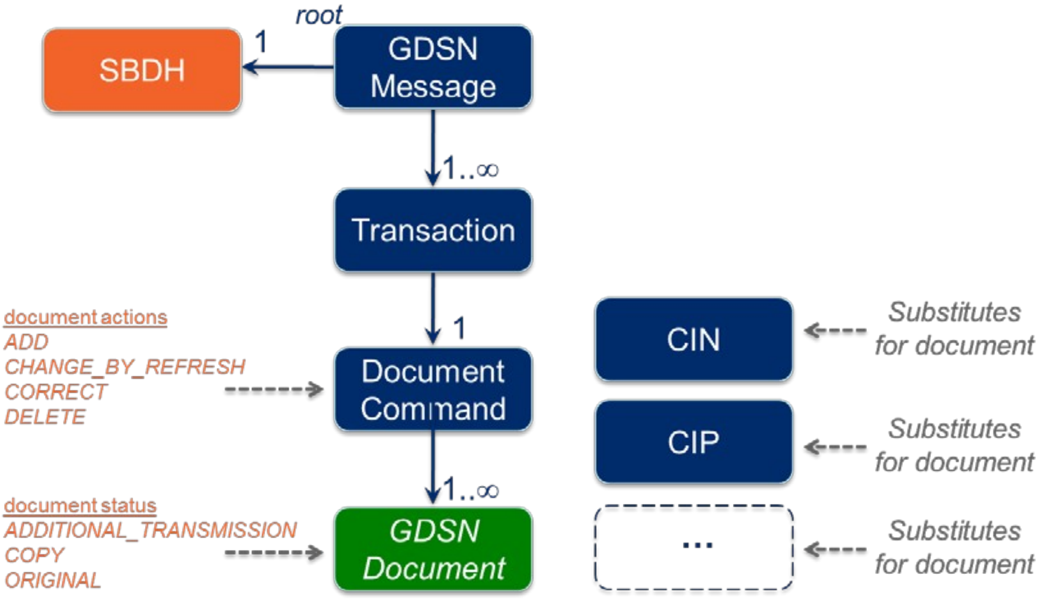


In the GS1 XML release 3 the business message components are defined in the XML schemas of three levels:

- Shared Common schema – contains all the components that can be re-used across domains, e.g. EDI and GDSN, such as GS1 Identification Keys, Party Identification, etc.
- Domain Common – contains all the components that can be re-used within one domain, e.g. GDSN, such as PartyInRole, Certification, etc. The domain specific components are defined using the shared components when available. Thus, the GdsnCommon.xsd needs to import the SharedCommon.xsd
- Message and Module schema – defines the structure of the actual message and the business document, as well as the content of the potential modules. The document and module specific components are defined using the shared and domain specific components when available. Thus, the BusinessMessage.xsd (e.g. CatalogueItemNotification.xsd, CatalogueItemSubscription.xsd) needs to import the SharedCommon.xsd and GdsnCommon.xsd

Message schema architecture includes layered structure introduced in release 2, with some changes.

Figure 4-2 Message schema architecture in Major Release 3



The message schema comprises the following components:

- The **root element** specific for the type of message, e.g. catalogueItemNotificationMessage or gS1ResponseMessage.
- The ***SBDH – Standard Business Document Header*** component is embedded in the root element of the message schema. The SBDH is defined in its own set of schemas, while the structure of the header is defined in the StandardBusinessDocumentHeader.xsd that needs to be imported into the BusinessMessage.xsd.
- The **Transaction** element may contain multiple GDSN documents, it allows processing of these documents together: if one of them fails, all of them will be discarded. One message can contain multiple transactions. Although in the schema the number of Transaction documents is specified as unbounded, their maximum number is set in the Validation Rules. The advantage of such an approach is that if in the future this limit must be changed, no schema change is required.
- Each Transaction contains one **DocumentCommand** element that instructs the recipient of that command to perform a particular action related to the documents within the command. There are four instructions (commands) to choose from:

  ADD – the receiver of the document (or documents) is instructed to store the document or documents

  CHANGE_BY_REFRESH – the receiver is instructed to update the existing document or documents, by total replacement

  CORRECT – the receiver is instructed to update the existing document or documents, by total replacement, skipping certain business specific validation rules. The syntactical and content validation rules still apply. This command is used in cases where the specific validation rules would otherwise prevent the application from changing data. It can be used only if the correction does not impact the integrity of the corrected data. Otherwise, correction should be performed by sending two commands: DELETE (with the old document) and ADD (with the new document) in one transaction.

  DELETE – the receiver is instructed to delete the document (or documents)

  Note: All documents in one transaction SHALL follow the same command. Documents to which different commands apply, SHALL be placed in a separate transaction element.

© 2018 GS1 AISBL

The **Document** element is defined as ‘abstract’, it means that it cannot be directly used in the instance document, but needs to be substituted by a ‘concrete’ element, e.g. catalogueItemNotification or catalogueItemPublication. The substituting elements inherit all the properties of the abstract document element, e.g. creation data and time, document status, etc.

Note: The GS1Response does not follow this principle, because its document element does not inherit the properties of the abstract document.

4.3 Standards Hosting

The Business Message Standards (BMS) from previous, current and new releases is available on the GS1 website at: *http://www.gs1.org/access-gdsn-standards*

GS1 XML Schemas and the example instance files for GDSN are placed in Implementer's Packets and made available for download.

4.4 GDSN XML Schemas hosting

GDSN schemas are based on the conventions of the GS1 XML Standard. As of the publication of this document, GDSN is supported by version 3.1 of GS1 XML Standards. The URL for directory for the latest version of the GDSN 3.1 standards is:

Beta: *http://www.gdsregistry.org/3.1/*

Production: *http://www.gs1globalregistry.net/3.1/*

4.5 GDSN XML Instance Documents

It is a responsibility of the Sender of GDSN XML messages to ensure that GDSN XML messages being sent are GDSN XML Schema-valid (conform to corresponding GDSN XML Schemas).

In addition to being valid GDSN XML Schema documents, all GDSN XML instance documents also SHALL be compliant with the following rules:

- All namespace declarations for GDSN instance documents SHALL be placed at the root element. The only namespace declarations that SHALL be placed inside the document, are the Trade Item Module namespaces. These SHALL be declared inside the ‘extension’ element.
  Below is a snippet from the GDSN **FoodAndBeverageIngredientModule** that illustrates the usage of the namespace declaration and schema location attributes:

```
<tradeItemInformation>
<extension>
<food_and_beverage_ingredient:foodAndBeverageIngredientModule
xmlns:food_and_beverage_ingredient="urn:gs1:gdsn:food_and_beverage_ingredient:xsd
:3" xsi:schemaLocation="urn:gs1:gdsn:food_and_beverage_ingredient:xsd:3
http://www.gdsregistry.org/3.1/schemas/gs1/gdsn//FoodAndBeverageIngredientModu
le.xsd">
.......... /*Module content*/  .........
</food_and_beverage_ingredient:foodAndBeverageIngredientModule>
</extension>
</tradeItemInformation>
```

Note: If there more than one module of the same kind is placed in the message, the validating schema and namespace SHALL be declared for all of them:

```
<tradeItemInformation>
<tradeItemComponents>
<componentInformation>
<componentNumber>1</componentNumber>
<extension>
<food_and_beverage_ingredient:foodAndBeverageIngredientModule
```

© 2018 GS1 AISBL

```
xmlns:food_and_beverage_ingredient="urn:gs1:gdsn:food_and_beverage_ingredient:xsd
:3" xsi:schemaLocation="urn:gs1:gdsn:food_and_beverage_ingredient:xsd:3
http://www.gdsregistry.org/3.1/schemas/gs1/gdsn//FoodAndBeverageIngredientModu
le.xsd">
………. /*Module content*/  ………
</food_and_beverage_ingredient:foodAndBeverageIngredientModule>
</extension>
<componentNumber>2</componentNumber>
<extension>
<food_and_beverage_ingredient:foodAndBeverageIngredientModule
xmlns:food_and_beverage_ingredient="urn:gs1:gdsn:food_and_beverage_ingredient:xsd
:3" xsi:schemaLocation="urn:gs1:gdsn:food_and_beverage_ingredient:xsd:3
http://www.gdsregistry.org/3.1/schemas/gs1/gdsn//FoodAndBeverageIngredientModu
le.xsd">
………. /*Module content*/  ………
</food_and_beverage_ingredient:foodAndBeverageIngredientModule>
</tradeItemInformation>
```

- GDSN XML instance documents SHALL include at the root element namespace declarations for the following namespace names:
  - http://www.w3.org/2001/XMLSchema-instance
  - http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader
  - urn:gs1:gdsn:[message_or_module_name]:xsd:3
- The following GS1 XML Schema namespace prefixes SHALL be used for the above namespace names as in:
  - xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
  - xmlns:[message_name]:="urn:gs1:gdsn:[message_name]:xsd:3"
- GDSN XML instance documents SHALL NOT include namespace declarations for extensions (modules) that are not present in the message.
- A recipient of the GDSN message SHALL NOT fail a GDSN message due solely to the existence of arbitrarily namespace prefixes. This does not supersede the data pool's responsibility to declare namespaces for extensions used and not declare namespaces for extensions that are not used.
- All GDSN XML instance documents SHALL include the fully qualified URLs that point to production schema locations as the value for the **schemaLocation** attribute.

Below is a snippet from the GDSN **CatalogItemNotification** message that illustrates the usage of the **schemaLocation** attribute:

```
<catalogue_item_notification:catalogueItemNotificationMessage
xmlns:catalogue_item_notification="urn:gs1:gdsn:catalogue_item_notification:xsd:3"
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:gs1:gdsn:catalogue_item_notification:xsd:3
http://www.gdsregistry.org/3.1/schemas/gs1/gdsn/CatalogueItemNotification.xsd">
```

4.6 Character Set in GDSN XML Instance Documents

All text characters, including special characters in every written language are allowed in GDSN instance documents. Parsing these characters is the responsibility of business partners and the Data Pools.

The XML instance documents, however, SHALL NOT contain any formatting characters such as HTML line breaks.

4.7 XML Schema Validation in GDSN

Since the GDSN standards are published as W3C XML schemas (XSD files), most of the attribute definitions and message restrictions that can be represented using XML syntax are encoded in the schemas. Thus, it is natural that these schemas will be used for validating the outgoing and incoming instance messages.

Schema validation of outgoing messages is critical for the quality of messages within the GDS network. If all the outgoing messages are valid against the respective schema, no incorrect instance messages will enter the network.

In case of incoming messages, the receiving Data Pool must provide specific and meaningful error messages, which would allow the sender to identify the problem and correct the message.

5 Message Transport

5.1 Communication Protocol

The EDI over the Internet Applicability Statement 2 (EDIINT AS2) protocol is the sole approved communications protocol for GDSN. The reader is referred to the following document for information on the use of this protocol: “EDIINT AS1 and AS2 Transport Communication Guidelines”. This document is available on the GS1 website at:

http://www.gs1.org/docs/xml/EDIINT_AS1_AS2_Transport_Comm_Guide_i1.pdf

6 GDSN Specifics

6.1 Validation Rules

The Standard GDSN Validation Rules are enumerated in the document which is available on the GS1 website at: *http://www.gs1.org/gdsn/gdsn-validation-rules/3-1*

All standard Error Message ID’s and Error Massages SHALL be used as defined in that document.

6.2 Receipt of Valid, Well-Formed Messages

Recipient Data Pools SHALL NOT fail any message that is valid and well-formed due to non-standard (i.e. local) validation rules.

Data Recipients MAY use the Catalogue Item Confirmation message, including the available CIC Response codes (including the free form text option) to communicate any issues with the related data.

6.3 Global attributes validation

The GDSN attributes may be either:

- Validations approved by GSMP, also referred to as global validations – the attribute is globally relevant and its value is always the same for every target market and for every trading partner
- Validations not approved by GSMP, introduced locally by the Data Pools, also referred to as local validations – their value may vary on different target markets and or the attribute may or may not be globally applicable.
- Global/Local the field is relevant for business cases around the world but while its definition is the same around the world it may have a different value depending on the geography.

The full list of all attributes and their Global / Local properties is published in the BMS Trade Item Modules, section 7.1.

Regardless of whether the attributes are global, global/local or local, they should always be validated using ALL three GDSN keys: GTIN, GLN and Target Market. The only exceptions are:

© 2018 GS1 AISBL

- GTIN
- Brand Name
- quantityOfNextLowerLevelTradeItem
- QuantityOfChildren

These attributes may be validated excluding the Target Market in support of GTIN Allocation Rules.

6.4 Target Market Specific Validations (TMSV)

Since there are instances where validations that are specific to a target market need to be enforced, it is important to define how those validation rules coexist with already established GDSN Validation Rules (Section 6.1) that have a global scope. Here are the rules that SHALL be followed in that case:

- A TMSV SHALL NOT be used to relax the characteristics of any attribute (e.g. from mandatory to be optional);
- A TMSV SHALL NOT be used to relax an existing validation with a global scope;
- A TMSV SHALL be supported by the GS1 Member Organization responsible for the target market in question.
- The Target Market of the Registry Catalogue Item (RCI) determines which Target Market-specific validations that are performed against that RCI.
- Target Market-specific validations SHALL be at the Target Market Country Code level (SHALL NOT be at the Target Market Subdivision level).
- The Catalogue Item Notification (CIN) sent by the Source Data Pool SHALL comply with all Target Market-specific validations.

6.5 Validation error message

GDSN standard provides two ways the validation errors may be reported to the sender:

- GS1Response
- Catalogue Item Confirmation

The message and the error code to be used, depends on the type of validation performed and the problem found.

The possible errors may result from a validation:

1. Errors resulting from XML Schema validation, report technical issues; the message sender needs to take corrective action
2. Errors resulting from GSMP approved Validation Rules, including the target market specific rules; the message sender needs to investigate the problem and decide what corrective action needs to be taken
3. Errors resulting from local Validation Rules these are rules not GSMP approved; the message sender needs to report the problem to the affected Trading Partners

Issue 1 (schema validation error) should be reported using the **GS1 Response** message and the responseStatusCode value '**REJECTED**'.

Issue 2 (GSMP approved Validation Rules) may be reported in two ways, depending which party needs to take the corrective action:

a. **GS1 Response** message and the responseStatusCode value '**REJECTED**' should be sent when the error is to be solved by the sender Data Pool or the com
b. **GS1 Response** message and the responseStatusCode value '**ACCEPTED**', followed by the **Catalogue Item Confirmation** and catalogueItemConfirmationStateCode value '**REVIEW**'

Issue 3 (local Validation Rules) should be reported using the **GS1 Response** message and the responseStatusCode value '**ACCEPTED'**, followed by the **Catalogue Item Confirmation** and catalogueItemConfirmationStateCode value '**REVIEW**'

Other, non GDSN ways of reporting the error, such as email notification, phone call, etc. should be avoided and used only exceptionally.

6.6 UN/CEFACT Standard Business Document Header use in GDSN

The UN/CEFACT Standard Business Document Header (SBDH) provides information about the routing and processing of the XML instance document. It can also optionally provide business scope and business service information. The SBDH is designed to be independent of the specific transport protocol used. Information contained in the SBDH can be used by communication applications to determine routing for any transport protocol.

Detailed UN/CEFACT Technical Specification for the SBDH and corresponding XML Schemas are available at the following location: *http://www.gs1.org/sbdh-technical-specifications*.

SBDH schemas used in GDSN are hosted with other GDSN schemas, as explained Section *4.4. GDSN XML Schemas*.

While GDSN SBDH usage is compliant with the previously mentioned SBDH specifications, GDSN standards impose additional semantics on its usage within the GDSN network. The following sections provide an overview of those constrains.

6.6.1 SBDH Usage Rules

The majority of usage rules of SBDH in GDSN are defined in the *GDSN Validation Rules*. Below are the rules for SBDH tags both covered by the Validation Rules and the additional ones.

- The Standard Business Document Header is OPTIONAL under the UN/CEFACT SBDH standard however, as published for use in GDSN it is MANDATORY**.** As of release 3, the SBDH is a mandatory part of the GS1 XML message, thus the StandardBusinessDocumentHeader element SHALL be used.

Table 6-1 Table 6-2 SBDH tags use in GDSN

Tag name	Schema defined occurrence	Use in GDSN	Value in GDSN
HeaderVersion	1..1	1..1	1.0
Sender	1..*	1..1	-
Identifier	1..1	1..1	Valid GLN of the message sender
Authority	0..1	1..1	GS1
ContactInformation	0..*	0	-
Receiver	1..*	1..1	-
Identifier	1..1	1..1	Valid GLN of the message receiver
Authority	0..1	1..1	GS1
ContactInformation	0..*	0	-
DocumentIdentification	1..1	1..1	-
Standard	1..1	1..1	GS1
TypeVersion	1..1	1..1	3.1 (current schema version)

© 2018 GS1 AISBL

Tag name	Schema defined occurrence	Use in GDSN	Value in GDSN
Type	1..1	1..1	Equal to GS1 document sent with this SBDH, written in lower camel case (message name concatenated, starting with small letter, each following word starting with capital letter), e.g. catalogueItemNotification, catalogueItemPublication, etc.
MultipleType	0..1	0	-
CreationDateAndTime	1..1	1..1	The date and time when the 'document originating application' created the document, which typically is trading partner. This date and time usually differs from the time stamping of the message by the EDIINT AS2 software.
Manifest	0..1	0	-
BusinessScope	0..1	0 in requesting messages 1..1 in all responses	-
Scope	0..*	1..1	-
Type	1..1	1..1	GDSN
InstanceIdentifier	1..1	1..1	The value of InstanceIdentifier element that is a child of the DocumentIdentification element
Identifier	0..1	0	-
CorrelationInformation	0..1	1..1 in all responses	
RequestingDocumentCreationDateTime	0..1	0..1	If used, its value SHALL match the value of the CreationDateAndTime element of the requesting message
RequestingDocumentInstanceIdentifier	0..1	1..1	SHALL equal original message instance identifier.
ExpectedResponseDateTime	0..1	0..1	

6.6.2 Annotated SBDH usage rules

The following is an annotated requesting message snippet that illustrates the usage rules of the SBDH in the GDSN. The example used is the Catalogue Item Notification message.

```
<catalogue_item_notification:catalogueItemNotificationMessage
xmlns:catalogue_item_notification="urn:gs1:gdsn:catalogue_item_notification:xsd:3"
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:gs1:gdsn:catalogue_item_notification:xsd:3
../Schemas/gs1/gdsn/CatalogueItemNotification.xsd">

    <sh:StandardBusinessDocumentHeader>                              /* MANDATORY */

        <sh:HeaderVersion>1.0</sh:HeaderVersion>                     /* MANDATORY */

        <sh:Sender<sh:Sender>

            <sh:Identifier Authority="GS1">6903148000007</sh:Identifier>

            /* PRIMARY MANDATORY MUST NOT USE CONTACT INFORMATION */

        </sh:Sender>

        <sh:Receiver>

            <sh:Identifier Authority="GS1">2203148000007</sh:Identifier>
            /* PRIMARY MANDATORY MUST NOT USE CONTACT INFORMATION */

        </sh:Receiver>

        <sh:DocumentIdentification>                                  /*  SHALL   NOT USE
        MULTIPLE TYPE */

            <sh:Standard>GS1</sh:Standard>

            <sh:TypeVersion>3.1</sh:TypeVersion>

            <sh:InstanceIdentifier>100002</sh:InstanceIdentifier>

            <sh:Type>catalogueItemNotification</sh:Type>

            <sh:CreationDateAndTime>2011-03-11T11:00:00.000-05:00</sh:CreationDateAndTime>

        </sh:DocumentIdentification>

          /* MUST NOT USE MANIFEST */

          /* MUST NOT USE BUSINESS SCOPE */

    </sh:StandardBusinessDocumentHeader>

    <transaction>                                                /*START OF THE GS1
    TRANSACTION LAYER*/

      <transactionIdentification>

        <entityIdentification>20051101</entityIdentification>

      <contentOwner>

       <gln>8712345678913</gln>                       /* MAY BE THE SAME AS SBDH SENDER /
    IDENTIFIER */

      </contentOwner>

      </transactionIdentification>

     <documentCommand>

      <documentCommandHeader type="ADD">

       <documentCommandIdentification>

        <entityIdentification>20051101</entityIdentification>

        <contentOwner>

         <gln>8712345678913</gln>                     /* MAY BE THE SAME AS SBDH SENDER /
    IDENTIFIER */

         </contentOwner>

        </documentCommandIdentification>

          </documentCommandHeader>
          <catalogue_item_notification:catalogueItemNotification>                 /*START OF THE GS1
          DOCUMENT*/
              <creationDateTime>2011-03-11T11:00:00.000-05:00</creationDateTime>
              <documentStatusCode>ORIGINAL</documentStatusCode>
              <catalogueItemNotificationIdentification>
                  <entityIdentification>100002</entityIdentification> /* MAY BE THE SAME AS SBDH
                                                                       INSTANCE IDENTIFIER */
                  <contentOwner>
                      <gln>8712345678913</gln>                      /* MAY BE THE SAME AS SBDH SENDER /
                      IDENTIFIER */
                  </contentOwner>
              </catalogueItemNotificationIdentification>
              <isReload>false</isReload>
              <catalogueItem>
                ....
          </catalogue_item_notification:catalogueItemNotification>
      </documentCommand>
    </transaction>
</catalogue_item_notification:catalogueItemNotificationMessage>
```

```
            THE FOLLOWING ELEMENT (BUSINESS SCOPE) AND ITS SUBELEMENTS MUST BE USED
            FOR ALL RESPONDING MESSAGES AND ONLY FOR RESPONDING MESSAGES
            <sh:BusinessScope>
                <sh:Scope>
                    <sh:Type>GDSN</sh:Type>
                    <sh:InstanceIdentifier>300001</sh:InstanceIdentifier>
                                        /* MAY BE THE SAME AS SBDH INSTANCE IDENTIFIER */
                                        /*   SHALL    NOT USE IDENTIFIER */
                     <sh:CorrelationInformation>
                        <sh:RequestingDocumentCreationDateTime>2004-01-10T10:15:01Z
                                        /*SAME AS THE CREATION DATE AND TIME IN THE REQUEST*/
                        </sh:RequestingDocumentCreationDateTime>
                        <sh:RequestingDocumentInstanceIdentifier>100002
                                        /*SAME AS THE INSTANCE IDENTIFIER IN THE SBDH REQUEST*/
                        </sh:RequestingDocumentInstanceIdentifier>
                    </sh:CorrelationInformation>                        /*   SHALL    NOT USE BUSINESS
                    SERVICE*/
                </sh:Scope>
            </sh:BusinessScope>
        </sh:StandardBusinessDocumentHeader>
        <gS1Response>
        ....
        </gS1Response>
```

6.7 Responses in GDSN

There are three types of response messages in GDSN:

- GS1Response
- CatalogueItemRegistrationResponse
- PartyRegistrationResponse

GS1Response indicates the processing success of transactional unit of work. It is not a transaction or a command or a document, but an indicator of the acceptance of a processed transaction.

CatalogItemRegistrationResponse and **PartyRegistrationResponse** serve the same purpose of accepting requesting message, but they are exclusively sent by the Global Registry upon successful registration of a Registry Catalogue Item (RCI) or a Registry Party (RP).

✓ **Note**: These differ slightly by how the responses reference the Identification of the Requesting Document (as opposed to the Requesting Transaction or Message).

GS1 Response with the responseStatusCode value 'REJECTED' is used to indicate back to the sender of the original (requesting) message various errors that might occur while processing the message at the recipient side.

In addition to specifically referring to a single response type, for the rest of this paragraph we will also use the following classification in order to refer to a proper subset of GDSN response messages:

- **GDSN Response** – include all four response types
- **Positive GDSN Response** – includes **CatalogItemRegistrationResponse, PartyRegistrationResponse**, and **GS1Response**.

Below are the rules that are enforced for GDSN response messages:

1. The GDSN Response SHALL correspond to one and only one requesting message. This means that it is not possible to respond to more than one requesting message at the same time.

1. The recipient of the GDSN requesting message that includes multiple transactions MAY package and send GDSN Responses related to original transactional requests either as a single GDSN response message or as multiple GDSN response messages.

2. Multiple GS1Response elements MAY be contained within a single GDSN response message.

3. The Positive GDSN Response SHALL have responseStatus attribute set to '**ACCEPTED**'.

4. The Positive GDSN Response identification SHALL correspond to the transaction identification of the requesting message.

   - The **uniqueCreatorIdentification** element and the contentOwner element that are children of the documentReceived / responseIdentification element in a Positive GDSN Response SHALL be copied from the uniqueCreatorIdentification element and the contentOwner element that are children of the entityIdentification element that is a child of the transaction element of the requesting message.

4. **CatalogItemRegistrationResponse:** Response identification SHALL correspond to the document identification of the requesting message.

   - The uniqueCreatorIdentification element and the contentOwner element that are children of the documentReceived / responseIdentification element in a Positive Catalogue Item Registration Response SHALL be copied from the uniqueCreatorIdentification element and the contentOwner element that are children of the entityIdentification element that is a child of the document element of the requesting message.

5. **PartyRegistrationResponse:** Response identification SHALL correspond to the document identification of the requesting message.

   - The uniqueCreatorIdentification element and the contentOwner element that are children of the documentReceived / responseIdentification element in a Positive Party Registration Response SHALL be copied from the uniqueCreatorIdentification element and the

© 2018 GS1 AISBL

contentOwner element that are children of the entityIdentification element that is a child of the document element of the requesting message.

6. The Sender and the Receiver of the **GS1Response** SHALL match the Sender and the Receiver that were present in the SBDH of the requesting message.
7. The recipient of the requesting message SHALL return as many **GS1 Response messages with the responseStatusCode value 'REJECTED'** as discovered during processing the message regardless of these errors being at different levels (message, transaction, command …).
8. The recipient of the requesting message SHALL return **GS1 Response messages with the responseStatusCode value 'REJECTED'** at the lowest level at which errors occurred.

6.8 Referencing original message in GDSN Response

When a GS1 Response message is being sent, it needs to reference the original message and / or transaction within the message it replies to.

6.8.1 Referencing originating message

The response message contains a dedicated field to reference to the originating message, with the following XPath: gs1_response/gS1Response/originatingMessageIdentifier/entityIdentification.

If the requesting (original) message has the following identification:

```
<catalogueItemNotificationIdentification>
    <entityIdentification>51101</entityIdentification>
    <contentOwner>
              <gln>8712345678913</gln>
    </contentOwner>
</catalogueItemNotificationIdentification>
```

The following should be contained in the response message:

```
<originatingMessageIdentifier>
    <entityIdentification>51101</entityIdentification>
    <contentOwner>
              <gln>8712345678913</gln>
    </contentOwner>
</originatingMessageIdentifier>
```

Please note that the GS1 message consists of two major parts – the header and the actual business documents wrapped in the transaction element (see section 4.2). The purpose of the header (SBDH, see section 6.4) is to provide document ID, sender, receiver and routing information without accessing the business document. Hence certain intentional repetitions between SBDH and document content.

Thus, it is also necessary to provide the reference to the originating document in the header part. This reference field in SBDH has the following XPath: gs1_response/StandardBusinessDocumentHeader/BusinessScope/Scope/CorrelationInformation/RequestingDocumentInstanceIdentifier.

```
sh:BusinessScope>
  <sh:Scope>
    <sh:Type>GDSN</sh:Type>
    <sh:InstanceIdentifier>12345</sh:InstanceIdentifier>
    <sh:CorrelationInformation>
      <sh:RequestingDocumentInstanceIdentifier>51101</sh:RequestingDocumentInstanceIdentifier>
    </sh:CorrelationInformation>
  </sh:Scope>
</sh:BusinessScope>
```

6.8.2 Referencing transaction in the originating message

The purpose of the transaction element is to group a number of documents to handle them together (see section 4.2). Response message references the transaction of the originating message. Depending on the scenario which the response is used in, it can be done in one of two places:

- The TransationResponse for positive responses: gs1_response:gS1ResponseMessage/gS1Response/transactionResponse/**transactionIdentifier/entityIdentification**
- The GS1 TransactionException for responses reporting errors:gs1_response:gS1ResponseMessage/gS1Response/transactionResponse/**transactionIdentifier/entityIdentification**

In both cases, the transaction identifier from the origination message needs to be referenced.

The transaction in every message has a unique identifier with the following XPath: Message/transaction/transactionIdentification/entityIdentification

Example of transaction identification:

```
<transactionIdentification>
        <entityIdentification>999876</entityIdentification>
        <contentOwner>
                <gln>8712345678913</gln>
        </contentOwner>
</transactionIdentification>
```

The reference to the originating transaction should look like that:

gS1Response/transactionResponse/transactionIdentifier/entityIdentification

Thus, the transaction ID must equal to the one in original message:

```
<transactionResponse>
        <entityIdentification>999876</entityIdentification>
        <contentOwner>
                <gln>8712345678913</gln>
        </contentOwner>

        <responseStatusCode>ACCEPTED</responseStatusCode>
</transactionResponse>

or

<transactionException>
        <entityIdentification>999876</entityIdentification>
        <contentOwner>
                <gln>8712345678913</gln>
        </contentOwner>
</ transactionException >
```

6.9 Identifiers in GDSN

6.9.1 Entity Identification

The GLN and GTIN are used to globally identify parties and trade Items in GDSN. A similar concept is used to identify messages, transactions, commands, etc. The structure used for this purpose is called **EntityIdentification**; here is the current XML Schema complex type definition for it:

```
<xsd:complexType name="EntityIdentificationType">
    <xsd:sequence>
        <xsd:element name="entityIdentification">
```

```
              <xsd:simpleType>
                      <xsd:restriction base="xsd:string">
                                <xsd:maxLength value="80"/>
                                <xsd:minLength value="1"/>
                      </xsd:restriction>
              </xsd:simpleType>
            </xsd:element>
            <xsd:element name="contentOwner" type="shared_common:PartyIdentificationType"
            minOccurs="0"/>
        </xsd:sequence>
</xsd:complexType>
```

As seen from this definition the **EntityIdentification** is composed of the mandatory entityIdentification element (the actual identifier) and an optional content owner's GLN (of the party who assigns the entityIdentification). The content owner itself is represented as a party, and as previously mentioned in GDSN party identification SHALL be ensured by leveraging the usage of GLN. The content owner SHALL ensure that the generated identifier is unique among entities within the owner's domain. The combination of GLN and the unique identifier makes the **EntityIdentification** globally unique.

6.9.2 Rules for GDSN Identifiers

Here are rules that need to be followed when creating identifiers in GDSN:

- Message identity (that is a combination of the **SBDH.Sender.Identifier** and the **SBDH.DocumentIdentification.InstanceIdentifier**) SHALL be globally unique and SHALL NOT be reused across different messages.

> **Note**: It is important to further explain the meaning of "reuse" in the above rule. "Reuse" denotes the case when two different messages (with a different purpose, content, etc.) happened to have the same identification. This is not allowed. On the other hand, there are various scenarios that might lead the sender to "resend" the original message; some of these are explained in Section *5.1 Communication Protocol*. In this case, the content of the whole message SHALL be identical to the original one, which implies that the message identification will be the same.

- The following attributes, when taken together, SHALL be unique in order to prevent duplicate documents and increase traceability within the GDSN:

Sender (Data Pool or Global Registry) GLN / SBDH.DocumentIdentification.InstanceIdentifier/ Transaction ID / Command ID / Document ID.

For example:

- If the same Document ID is used within the same Command the transaction MAY be failed by the receiving data pool.
- If the same Command ID is used within the same transaction the transaction MAY be failed by the receiving data pool.
- If the same Transaction ID is used within the same message the message MAY be failed by the receiving data pool.
- A receiving data pool MAY fail any messages from a sending data pool if the sending data pool has previously sent the receiving data pool a message using the same **SBDH.DocumentIdentification.InstanceIdentifier**.

6.10 Standard on Transaction, Commands and Documents

When transmitting transactions, commands and documents in GDSN, a consistent convention in the usage of these 3 elements SHALL be followed. There is a distinction between what is technically valid within a well-formed and validated GS1 XML instance document and the desired standard from the business point of view. This section describes the standard for usage of transactions, commands and documents in the GDSN.

The following restrictions apply when sending messages within the GDSN:

1. There is a limit of 1 Document type within 1 Message
2. There is a limit of 1000 Transactions within 1 Message
3. There is a limit of 1 Command type within 1 Transaction
4. There is a limit of 100 Documents within 1 Transaction
5. Implementation of commands and documents could take the form of:
    - Wrapping all documents within one command
    - Including command/document pairs multiple times

If a data pool receives a message that does not follow these restrictions, the receiving data pool MAY reject this message and send a GS1 Response message with the responseStatusCode value 'REJECTED'.

Note: As there were some misunderstandings regarding the Party Dump functionality between the Global Registry and data pools, it is worth mentioning that Party Dump is a file and not a GDSN message. As such, it is not restricted to the above limits.

6.11 Content Owner

Within the GDSN data pools need to be consistent in populating the *content owner* of a message at the transaction, command and document levels. The following rule is enforced for all GDSN messages:

- The **contentOwner** GLN value at the message level is the GLN of the Data Pool / Global Registry. The **contentOwner** GLN value at the transaction, command and document levels is the GLN of the Data Source / Data Recipient.

6.12 Language Code

The language code that is used in Description types within the GDSN release 3.1 is expressed as the **languageCode** of type xsd:string. Although the actual code values are not enumerated in the schema, and the code set to be used is not specified, the standard requires that the code values from ISO 639 list SHALL be used.

Below is an example of one of the Description types used in GDSN 3.1:

Description Type example:

```
<xsd:complexType name="Description200Type">
    <xsd:simpleContent>
        <xsd:extension base="shared_common:String200Type">
            <xsd:attribute name="languageCode" use="required">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:maxLength value="80"/>
                        <xsd:minLength value="1"/>
```

© 2018 GS1 AISBL

```
                                        </xsd:restriction>
                              </xsd:simpleType>
                    </xsd:attribute>
                    <xsd:attribute name="codeListVersion">
                    ....
                    </xsd:attribute>
            </xsd:extension>
      </xsd:simpleContent>
</xsd:complexType>
```

The two positions of **language code** SHALL be a valid language code from ISO 639-1 code, (i.e. "en", "ru", "zh") and these SHALL be lower case.

Examples of correct language codes

- “en”
- “ru”
- “zh”

Examples of incorrect language codes

- “EN”
- “tw”
- “ZH”
- “en840”
- “en826”
- “826”
-

6.13 DateTime Format

Currently there are various places in GDSN XML schemas where data elements are of the “**xsd:dateTime**” data type. The population of data elements that are **xsd:dateTime** data type SHALL be compliant to constrain described in the following subsections.

6.13.1 Time Offset Indication

The GDSN messages that are sent in small time intervals may arrive in different order. The RDP may need to determine what the intended order was, thus it is a good practice to specify time zone offset in attributes with DateTime data type at the header and document level.

These are:

1. **Standard Business Document Header** (SBDH) level
   - For ALL GDSN messages - **CreationDateAndTime** in Standard Business Document Header at the following location (example XPath): /catalogue_item_notification:catalogueItemNotificationMessage/sh:StandardBusinessDocumentHeader/sh:DocumentIdentification/sh:CreationDateAndTime
   - For Response messages – **RequestingDocumentCreationDateTime** at the following location (example XPath):

/gs1_response:gS1ResponseMessage/sh:StandardBusinessDocumentHeader/sh:BusinessScope/sh:Scope/sh:CorrelationInformation/sh:RequestingDocumentCreationDateTime

2. **Document** level:

   - For ALL GDSN messages – **creationDateTime** at the following location: /catalogue_item_notification:catalogueItemNotificationMessage/transaction/documentCommand/catalogue_item_notification (example XPath): catalogueItemNotification/creationDateTime

   - When applicable – **lastUpdateDateTime** at the following location: /catalogue_item_notification:catalogueItemNotificationMessage/transaction/documentCommand/catalogue_item_notification (example XPath): catalogueItemNotification/lastUpdateDateTime

For the other Trade Item the time offset should only be used if there is a clear business reason, such as business attributes that may be subject to legally binding commercial contract. In that case, a GSMP Work Request needs to be submitted, to use the time offset.

Use of the time zone indicator in ALL GDSN attributes of **xsd:dateTime** data type is **optional**, however, when it needs to be explicitly specified, it SHALL be expressed in local time with a time zone offset in hours and minutes ("**+hh:mm**" for times that are ahead of UTC and "**-hh:mm**" for times that are behind UTC.

Example of correct dateTime at Header and Document level:

- "2006-05-15T05:37:39-05:00" – corresponds to May 15, 2006, 5:37:39 am, US Eastern Standard Time

Example of an incorrect dateTime at Header and Document level:

- "2006-05-15T05:37:39" – no time offset indicator

6.13.2 Time Precision

In some cases it is important to determine the precise order of the incoming messages by the RDP. An example may be multiple Catalogue Item Confirmation (CIC) messages with different status, such as 'SYNCHRONISED' and 'REVIEW'. Although they may be sent in longer time intervals, they may arrive to the RDP almost simultaneously. The RDP must know which message had been sent the latest, as this status will overwrite all the previous ones.

The order is determined based on the value of creationDateTime attribute. To enable this, it is recommended to provide sufficient time precision.

The dateTime data type allow additional digits to be used to increase the precision of fractional seconds if desired i.e. the format ss.s... with any number of digits after the decimal point is supported.

The 'ss.s' denotes 2 digits of second (00 through 59) followed by one or more digits representing a decimal fraction of a second (milliseconds). The fractional second part is separated from the 2 digits of second by the use of a 'dot' as a separator. For creationDateTime attribute, it is a good practice to use three precision digits to denote the milliseconds.

Examples

1. creationDateTime="2017-03-22T09:30:47-05:00"
   Represents 47 seconds and 0 milliseconds.

2. creationDateTime="2017-03-22T09:30:47.0-05:00"
   Represents 47 seconds and 0 milliseconds and is equivalent to example 1 from above.

3. creationDateTime="2017-03-22T09:30:47.233-05:00"
   Represents 47 seconds and 233 milliseconds.

6.14 Country and Country Subdivision format

GDSN uses ISO standard code lists to refer to country and country subdivision.

Countries are encoded using ISO 3166-1 code list. This code list provides three different formats of encoding country name, in GDSN the three digit numeric code is used.

Example

In GDSN, country code is used e.g. in targetMarketCountryCode attribute.

<targetMarketCountryCode>360</targetMarketCountryCode> (360 is an ISO code for Indonesia)

<targetMarketCountryCode>840</targetMarketCountryCode> (840 is an ISO code for United States of America)

Country sub-division is encoded using ISO 3166-2 code list. This code list contains the name of principal subdivision (e.g. province or state) of countries encoded in ISO 3166-1. This code is based on the two-letter code element from ISO 3166-1 followed by a hyphen separator and up to three alphanumeric characters.

Example

In GDSN, country code is used e.g. in targetMarketSubdivisionCode attribute:

<targetMarketSubdivisionCode>ID-RI</ targetMarketSubdivisionCode> (ID-RI is an ISO code for Riau province of Indonesia)

<targetMarketSubdivisionCode>US-CA</ targetMarketSubdivisionCode> (US-CA is an ISO code for USA state of California)

6.15 Optional Messages, Attributes and Extensions

Within the GDSN, data pools might have a choice when responding to optional messages and messages that contain optional attributes or extensions. The following rule is enforced for those GDSN messages:

1. Data pools and the Global Registry SHALL NOT fail (GS1 Response message with the responseStatusCode value 'REJECTED') a GDSN message due solely to the existence of an optional attribute or well-formed extension.

    □ If the contents of the Attribute Value Pair Extension (AVP), and/or Extended Attributes, are not supported, it is the Recipient and/or Recipient Data Pool's responsibility to ignore those contents. It should be understood, that the Data Source is not required to alter the message to satisfy any review comment as the same item may be sent to another recipient who can process these contents.

1. The receiving data pool MAY fail (GS1 Response message with the responseStatusCode value 'REJECTED') optional messages within the network that are not supported by the receiving data pool.

2. If there is a functional way to use the production version of the schema to send standardised data, the data should not be sent in an additional, duplicated way in an Attribute Value Pair (AVP) extension.

6.16 Handling of optional XML elements with no data

GDSN standards defines 2 types of structures that potentially may be sent as empty elements:

- UML classes with all attributes and associations optional
- UML attributes of unrestricted string data type

Both are mapped to the schema as XML elements – see the two following sub-sections. From purely technical point of view, in the instance file the element tags can be sent without any data. This will be correctly validated against the schema, but may be confusing for the receiving party, who cannot be sure:

- is the element intentionally empty?
- is the data / child elements forgotten?
- is the file corrupted?

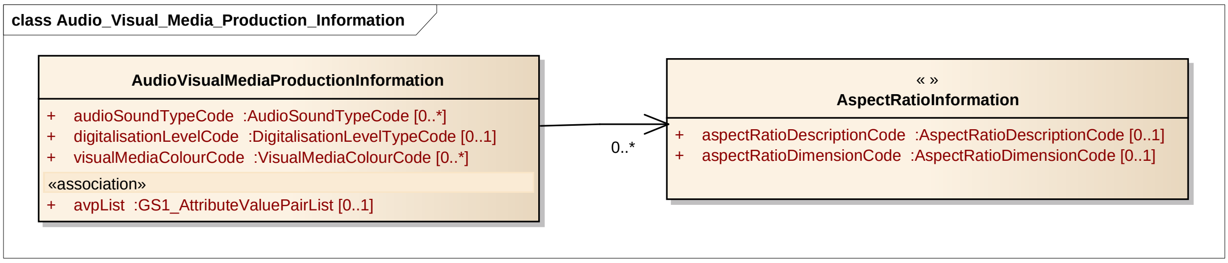
Therefore, empty elements should be avoided. If there is no data for optional element, it needs to be omitted.

6.16.1 Classes with only optional content

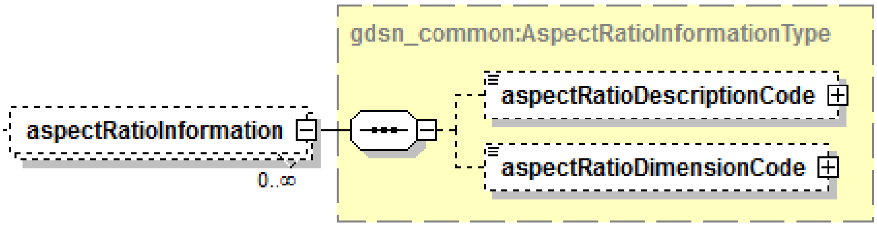
Some UML classes in GDSN standards have all the attributes and associations (if applicable) with optional cardinality. They are mapped to XSD as optional elements.

Example

AspectRationInformation class, associated toAaudioVisualMediaProductionInformation class in UML:


class Audio_Visual_Media_Production_Information
AudioVisualMediaProductionInformation
+ audioSoundTypeCode :AudioSoundTypeCode [0..*]
+ digitalisationLevelCode :DigitalisationLevelTypeCode [0..1]
+ visualMediaColourCode :VisualMediaColourCode [0..*]
«association»
+ avpList :GS1_AttributeValuePairList [0..1]
0..*
« »
AspectRatioInformation
+ aspectRatioDescriptionCode :AspectRatioDescriptionCode [0..1]
+ aspectRatioDimensionCode :AspectRatioDimensionCode [0..1]

AspectRationInformation mapped to XSD:


gdsn_common:AspectRatioInformationType
aspectRatioInformation
0..∞
aspectRatioDescriptionCode
aspectRatioDimensionCode

When there is no data to be sent for aspectRatioDescriptionCodea nd AspectRationDimensionCode, the whole structure – including parent element – should be omitted in the XML instance document.

An example of **correct** handling of no-data optional elements ('aspectRatioInformation' is omitted):

```
<audioVisualMediaProductionInformation>
   <!--<other elements>-->
   <!--<other elements>-->
</audioVisualMediaProductionInformation>
```

Sending just the empty parent element is not correct in GDSN.

An example of **incorrect** handling of no-data optional elements (aspectRatioInformation is omitted):

```
<audioVisualMediaProductionInformation>
   <!--<other elements>-->
    <aspectRatioInformation/>
   <!--<other elements>-->
</audioVisualMediaProductionInformation>
```

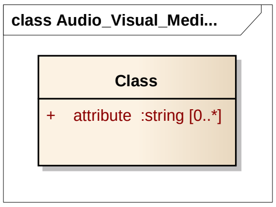
The second option is valid against the XML schema, but is not allowed in GDSN.

6.16.2 UML attribute with unrestricted string data type

Some attributes of UML classes in GDSN standards contain attributes that have data type string, with no further constraints. When mapped to XML, it means that the element may contain 0 characters of data – it other words, can be validly empty.

Example

Unrestricted string data type attribute in UML:



The call mapped to XSD:

```
<xs:element name=“class">
   <xs:complexType>
      <xs:sequence>
         <xs:element name=“attribute" minOccurs="0“ type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

When there is no data to be sent for the attribute, it should be omitted from the instance document.

An example of **correct** handling of no-data unrestricted string attribute (‘attribute’ is omitted):

```
<class>
 <!--<other elements>-->
</class>
```

Sending just the empty element is not correct in GDSN.

An example of **incorrect** handling of no-data optional elements (attribute is omitted):

```
<class>
 <!--<other elements>-->
   <attribute/>
</class>
```

The second option is valid against the XML schema, but is not allowed in GDSN.

6.17 Deleting published Catalogue Item

GDSN release 3.1 introduced the CatalogueItemHierarchicalWithdrawal message which can be used to withdraw a published hierarchy to correct an issue with the hierarchical links but also can be used by the Data Source or Data Pool to inform their respective Data Recipient that a given Catalogue Item is being withdrawn from publication to a trading partner.

In the previous releases, the same function was performed by sending the Catalogue Item Notification message, with command DELETE. However, this way required re-sending the full trade item details. Besides, having two different ways of achieving the same goal causes confusion in the network.

Therefore, two Validation Rules have been introduced. One forbidding the use of Catalogue Item Notification message, with command DELETE to withdraw a published hierarchy (VR 1458). The second one, requiring the use of command DELETE in Catalogue Item Hierarchical Withdrawal and forbidding use of any other commands in this message (VR 1457).

It is critical that all the Data Pools follow the two rules and use just one method of deleting the published Catalogue Item. The Validation Rules can be accessed on *GS1 website*.

✓ **Note**: These Validation Rules are binding as of release 3.1.3.

© 2018 GS1 AISBL

6.18 GDSN Modules

6.18.1 Module ordering

Trade Item Modules SHALL be sent in alphabetical order of the XML tag names. Since the module names in XML start with the namespace prefix that is composed of separate words divides by underscore, the sequence of letters in the prefix decides about the alphabetical order.

✓ **Note**: Please note that if for two concatenated prefixes the same letter appears the last of the preceding word and as the first one after the underscore, the one after the underscore takes precedence.

Examples of CORRECT alphabetical order (namespace prefixes included):

- audio_visual_media_content_information:audioVisualMediaContentInformationModule
- audio_visual_media_product_description_information:audioVisualMediaProductDescriptionInformationModule
- audio_visual_media_production_information:audioVisualMediaProductionInformationModule
- health_wellness_packaging_marking:healthWellnessPackagingMarkingModule
- healthcare_item_information:healthcareItemInformationModule

Examples of INCORRECT alphabetical order (namespace prefixes excluded):

- audioVisualMediaContentInformationModule
- audioVisualMediaProductDescriptionModule
- audioVisualMediaProductionInformationModule
- healthcareItemInformationModule
- healthWellnessPackagingMarkingModule

The rationale for this rule is that some users work with legacy mapping tools that require certain order to be specified. If this order is not followed, some of the modules may not be processed. Thus, to keep the reliability of data transfer, the alphabetical order needs to be followed.

6.18.2 Single instance / same module

An item defined as a single pallet, case, each (or other unit descriptor) SHALL have only a single instance of any single module per extension point. The same module MAY be passed at the item and item component level (ComponentInformation) for a single item. Note the same module MAY be passed at each component extension point.

The following rules are enforced for those GDSN messages:

1. Data pools SHALL NOT fail (GS1 response message with the responseStatusCode value ‘REJECTED’) a GDSN message due solely to the existence of one instance of a well formed module.
2. The receiving data pool MAY fail (GS1 response message with the responseStatusCode value ‘REJECTED’) a GDSN message within the network where there are more than one instance of the same modules.

6.19 Custom extensions

While every effort is being made to standardise the trade item attributes exchanged within GDSN, sometimes Data Pools have business need to pass attributes that are not defined in the standard, but are required by trading partners. These attributes should preferably be sent using the AVP structure but in cases of more complex data structures a custom extension may be sent.

In release 3.1, extension points are defined both at the trade item information level and the component level. Custom extensions should be placed at the trade item level. It is not recommended to send custom extensions at the component level until this functionality is better defined in the GDSN standards.

Custom extensions SHALL be placed after the standard modules. If there are multiple custom extensions, they SHALL be sent in alphabetical order of the XML tag names.

If a trade item has multiple variants, each variant should be treated as if it was the only variant being received by a data recipient. For example, if the same module and its data is required by two variants, this module should be repeated for each variant with the same values. If a custom extension is sent for only a single variant, it is assumed that this information is not relevant to the other variants.

6.20 Deprecation of Attributes

During the course of standards development, attributes can become obsolete or merged with other attributes to create more flexible implementations to support a broader community of product.

Therefore, within a GDSN release an attribute can be deprecated from service, or more commonly, an attribute may be moved into a more open, standardized implementation.

A common example of an attribute moving into a more open and standardized implementation would be during the migration of an “industry-specific” set of attributes into an “industry neutral” implementation. When this occurs, both the old and new attributes can coexist physically within the schemas. In such a scenario, while the older attribute would be considered obsolete, it could still be physically populated without error.

The paragraphs below outline an agreement between GDSN-certified data pools on how to handle certain attribute support scenarios when involving the deprecation of attributes.

6.20.1 Deprecation Due to Obsolescence

This classification defines an attribute that will no longer functionally exist after a release. The network has no other requirements to support this attribute, and it is considered no longer relevant.

Since a physical schema element cannot be removed until a major release, the schema will still show the attribute for a given amount of time. In this scenario, data pools SHALL NOT send this attribute into the network anymore.

6.20.2 Deprecation Due to Standardization or Enhancement

This classification defines either an attribute where the initial implementation is to be standardized now (e.g., an Extended Attribute moving into the Standard), or an attribute that is being moved from an industry-specific extension to a common industry-neutral set of core attributes.

Under this scenario, the original implementation of the attribute, regardless of location in the GDSN document, will be functionally changing and/or moving physically to another more accessible area of the GDSN document.

Since physical schema elements cannot be removed until a major release, the schema will still show the attribute in both its old location as well as its new location. This dual implementation often leads to confusion, unnecessary implementation costs, and data quality issues.

For any deprecation of an attribute, only the newest implementation SHALL be communicated between data pools to reduce costs and confusion inside the network.

6.21 Code list Validation

The recipient has an obligation to accept all the code lists that are published as a part of the GDSN standards in the release currently used by the network. Note that the use of some code lists is subject to restrictions, which needs to be observed.

Example: the unit of measure code list: GDSN_MeasurementUnitCode contains both metric and imperial units of measure, however, the standard specifies the following limitation: “Data Sources should be permitted to send either metric or imperial measurements, depending on the Target Market. The Data Source should provide the measurement system that is required in a specific target market."

The following rule allows optional rejection of codes:

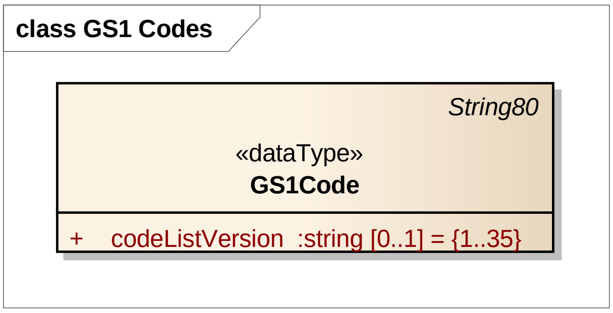
The recipient has the right (but not the obligation) to reject values for all attributes for which a standard code list applies, if the value in question is not a member of the code list that applies to that attribute.

6.22 Code list versioning

GDSN code lists can be updated with various frequencies. Version number for each code list is specified in the BMS for each code list for example:

GS1 Code List	**AllergenTypeCode**
GS1 Code List Version	**1**

Every code list in GS1 XML standards is defined following the common structure:



Thus, every code list element will have a 'codeListVersion' attribute. While this attribute is optional, it is advisable to pass the version number associated with the code list specified in the standard.

Versions of GS1 code lists are always specified using sequential number, starting from 1, the next one will be 2, 3 and so on. This number should be used to specify the code list version.

Below is an example of one of the GS1 codes used in GDSN 3.1:

GS1 code element example:

<partyRoleCode **codeListVersion="2"**>INFORMATION_PROVIDER</partyRoleCode>

Versions of external code lists are determined for ALL the GDSN users and specified in the code list definition in the Global Data Dictionary. For example, CurrencyCode list is based on ISO 4217, fully adopted by GS1. The GDD definition contains the following statement: "GDSN users should implement ISO code list version 2015. Users from other domains may select bilaterally preferred version." Should the community decide to use another version, this statement will be amended. The GS1 specified version for CurrencyCode is currently "1" and this value should be used to specify the code list version.

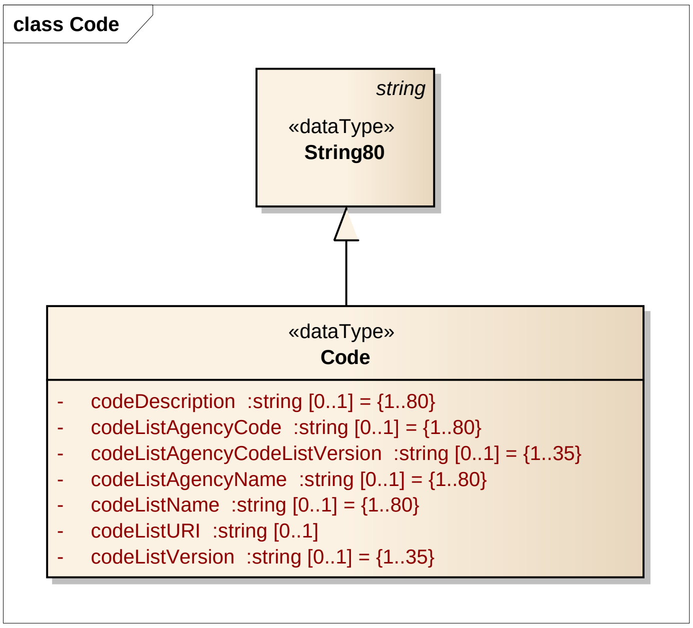
Below is an example of one of the codes from the CurrencyCode list used in GDSN 3.1:

GS1 code element example:

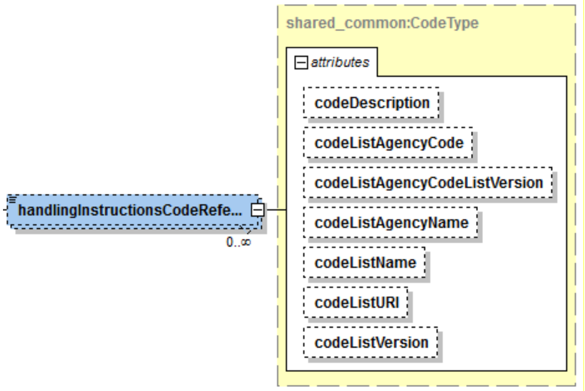
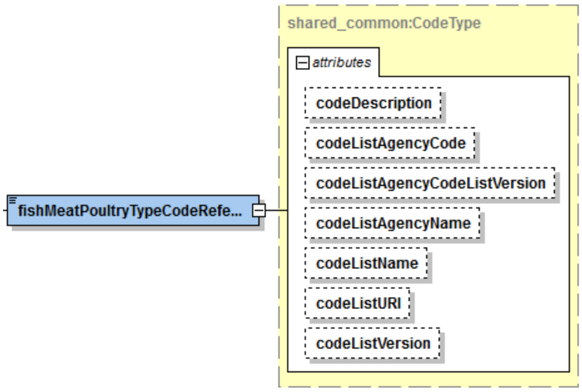
<propertyAmount currencyCode="USD" codeListVersion="1">3.14</propertyAmount>

6.23 Populating code data for code lists not defined by GS1

GS1 does not always stipulate specific code list to be used in the standard. It is done when different code lists are used for the same attribute in multiple target markets. These code lists can optionally be populated with metadata about the code used and the code list where it is defined, using the following common structure:


class Code
string
«dataType»
String80
«dataType»
Code
- codeDescription :string [0..1] = {1..80}
- codeListAgencyCode :string [0..1] = {1..80}
- codeListAgencyCodeListVersion :string [0..1] = {1..35}
- codeListAgencyName :string [0..1] = {1..80}
- codeListName :string [0..1] = {1..80}
- codeListURI :string [0..1]
- codeListVersion :string [0..1] = {1..35}

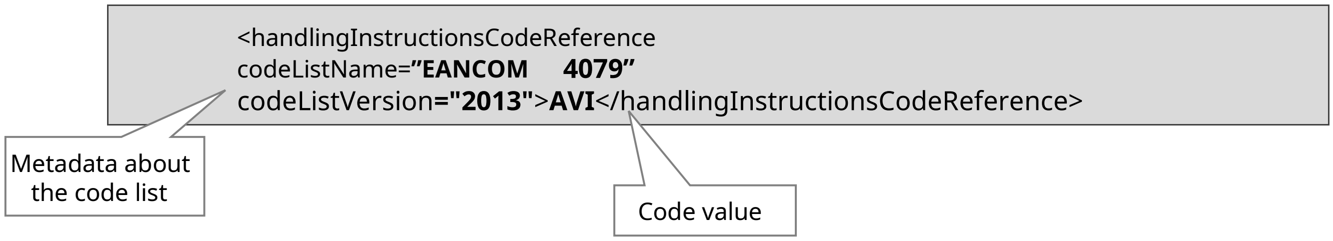
This structure would be mapped to the XML schema as following:

For handlingInstructionsCodeReference:	For fishMeatPoultryTypeCodeReference:
	

© 2018 GS1 AISBL

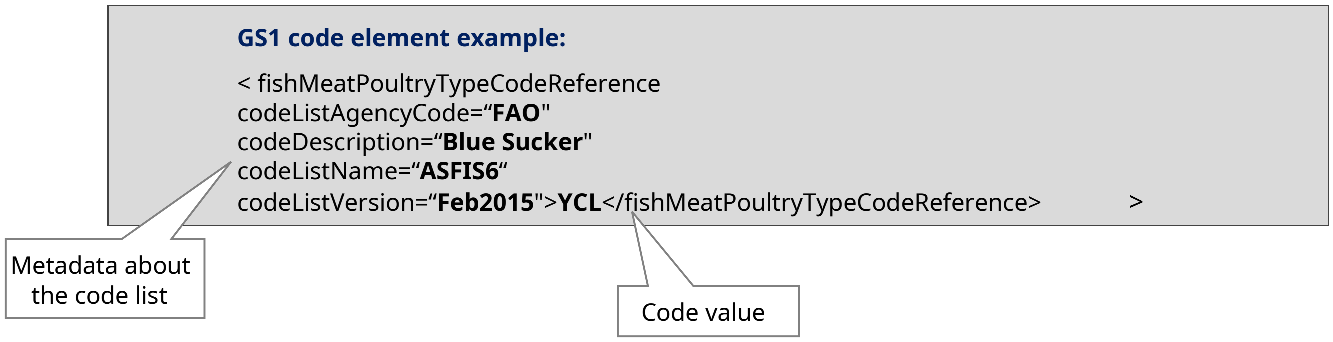
Below are two examples of how to populate the Code data type:

Example of handlingInstructionsCodeReference code:



```
<handlingInstructionsCodeReference
codeListName="EANCOM     4079"
codeListVersion="2013">AVI</handlingInstructionsCodeReference>
```

Example of fishMeatPoultryTypeCodeReference code:



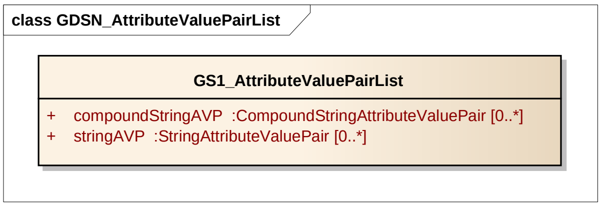
GS1 code element example:

```
< fishMeatPoultryTypeCodeReference
codeListAgencyCode="FAO"
codeDescription="Blue Sucker"
codeListName="ASFIS6"
codeListVersion="Feb2015">YCL</fishMeatPoultryTypeCodeReference>           >
```

6.24 Populating Attribute Value Pair component

Every GDSN XML schema contains a number of placeholders that allow sending attributes that are approved by the GDSN Community, but are not yet part of the GDSN standard – the Fast Track Attributes or Custom Extensions (see section 6.14).

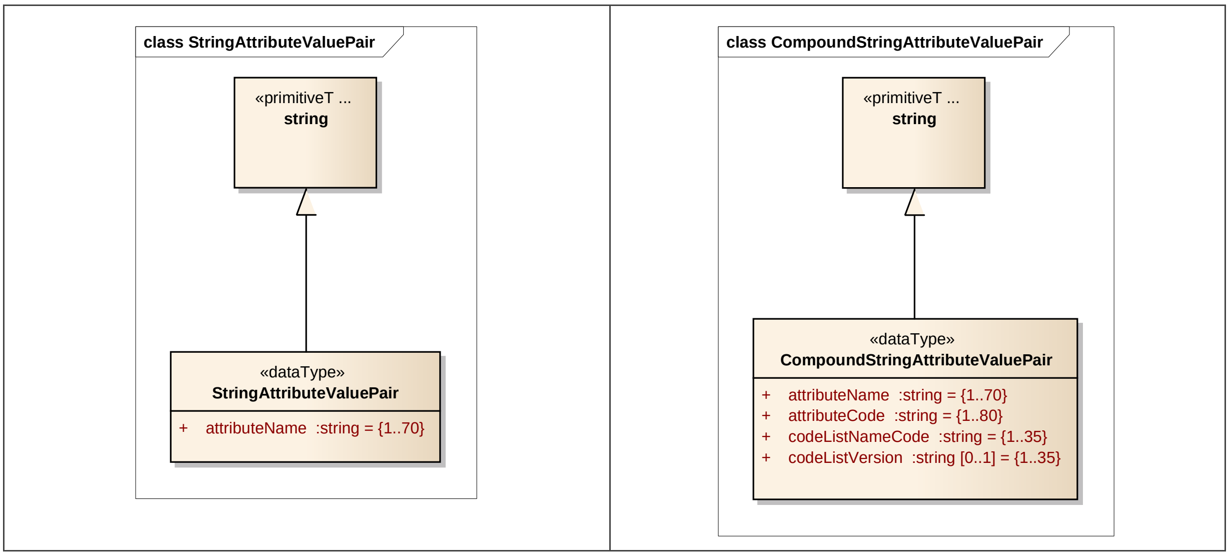
This placeholder is called the Attribute Value Pair (AVP). The AVPs are defined using the following common structure:



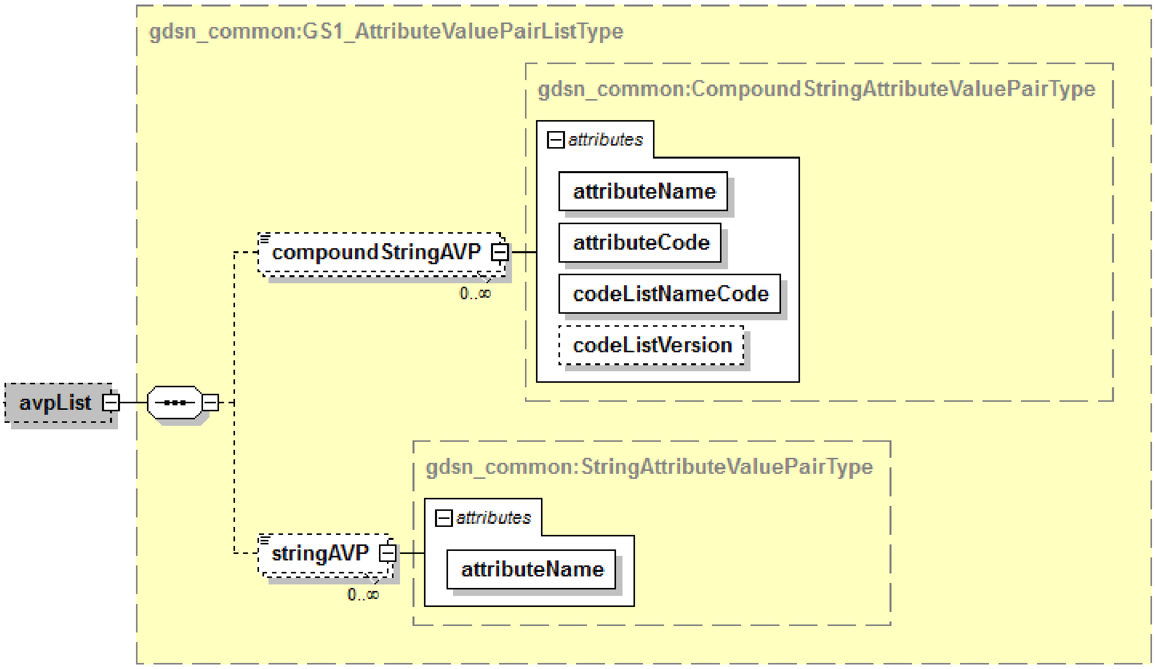
class GDSN_AttributeValuePairList
GS1_AttributeValuePairList
+ compoundStringAVP :CompoundStringAttributeValuePair [0..*]
+ stringAVP :StringAttributeValuePair [0..*]

The AVP may be of 2 types: Simple – with the StringAttributeValuePair structure or Complex – with the CompoundStringAttributeValuePair. The class diagrams for both structures are as follows:

StringAttributeValuePair – Simple AVP	CompoundStringAttributeValuePair – Complex AVP


class StringAttributeValuePair
«primitiveT ...
string
«dataType»
StringAttributeValuePair
+ attributeName :string = {1..70}
class CompoundStringAttributeValuePair
«primitiveT ...
string
«dataType»
CompoundStringAttributeValuePair
+ attributeName :string = {1..70}
+ attributeCode :string = {1..80}
+ codeListNameCode :string = {1..35}
+ codeListVersion :string [0..1] = {1..35}

These structures are mapped to the XML schema as following:



The approved Fast Track Attributes are published in a spreadsheet on the GDSN website http://www.gs1.org/access-gdsn-standards. The AVP Type column in the spreadsheet specifies which AVP structure should be used: Simple or Complex.

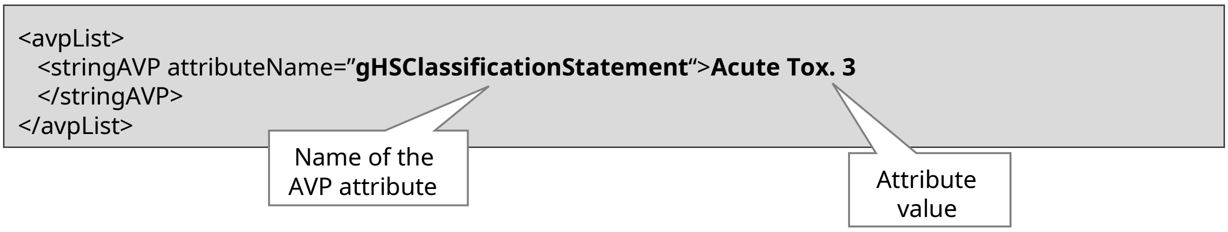
Below are two examples of how to populate the AVP.

Example of the Simple AVP

Selected columns from the Fast Track Attribute spreadsheet:

Attribute/Role	Module/Class to insert AVP into	AVP Type	Data Type/Secondary Class	Facets
gHSClassificationStatement	SafetyDataSheetModule /SafetyDataSheetInformation	Simple	string250	1..250

The above attribute mapped to XML:



```
<avpList>
  <stringAVP attributeName=”gHSClassificationStatement“>Acute Tox. 3
  </stringAVP>
</avpList>
```

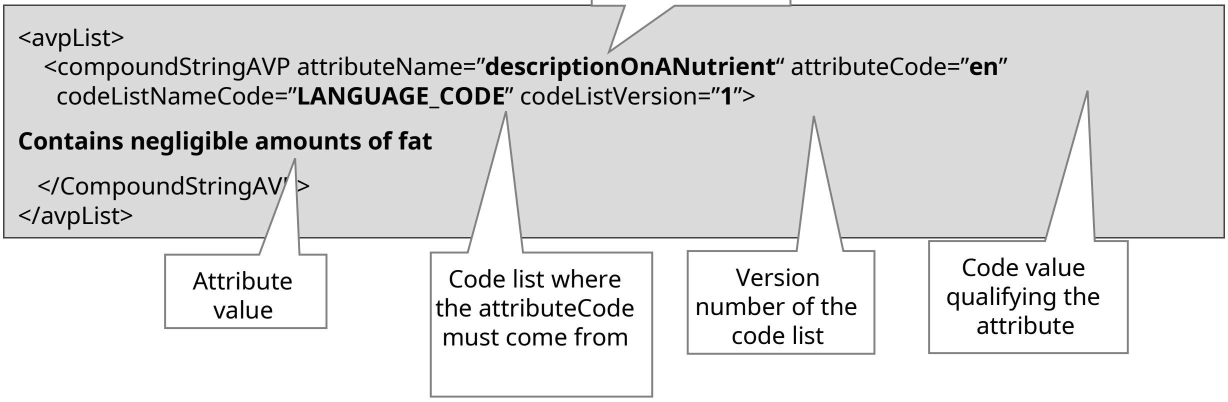
- The AVP Type structure to be used for this attribute is Simple - stringAVP
- Name of the attribute comes from the Attribute/Role column in spreadsheet
- Attribute value conform to the Data Type + Facets column constraints; here: the value must be of data type String. From 1 to 250 characters.
- The AVP placeholder location for this attribute is SafetyDataSheetModule, class SafetyDataSheetInformation

Example 1 of the Complex AVP

Selected columns from the Fast Track Attribute spreadsheet:

Attribute/Role	Module/Class to insert AVP into	AVP Type	Data Type/Secondary Class	Facets
descriptionOnANutrient	NutritionalInformationModule /NutrientHeader	Complex LANGUAGE_CODE	*Description1000*	1..1000

The above attribute mapped to XML:



```
<avpList>
   <compoundStringAVP attributeName=”descriptionOnANutrient“ attributeCode=”en”
     codeListNameCode=”LANGUAGE_CODE” codeListVersion=”1”>
Contains negligible amounts of fat
  </CompoundStringAVP>
</avpList>
```

Name of the AVP attribute
Attribute value
Code list where the attributeCode must come from
Version number of the code list
Code value qualifying the attribute

- The AVP Type structure to be used for this attribute is Complex - CompoundStringAVP
- Name of the attribute comes from the Attribute/Role column in spreadsheet

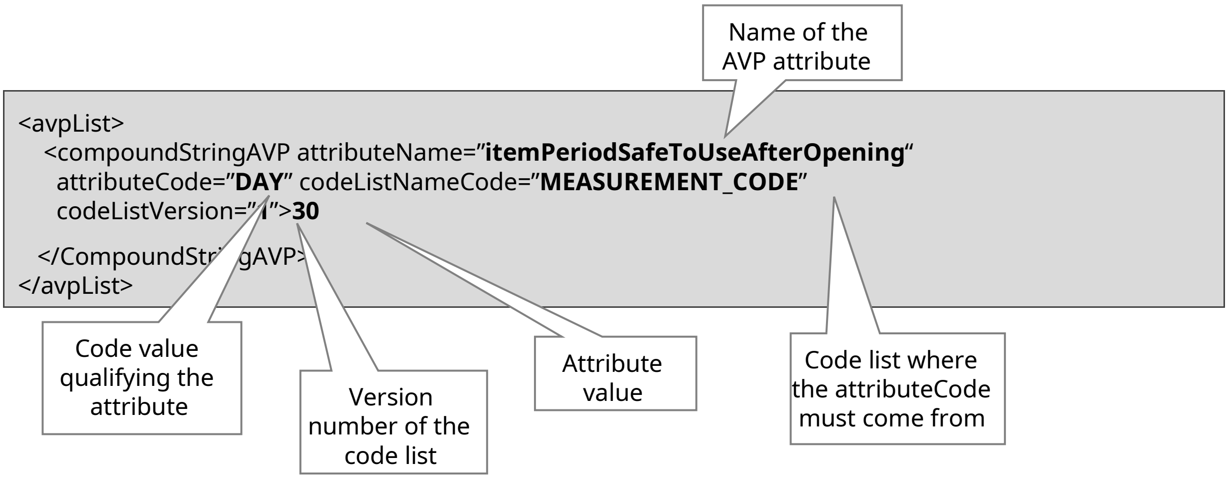
- Attribute value conform to the Data Type + Facets column constraints; here it must be between 1 and 1000 characters.
- The data type structure to be used for populating XML attributes should conform to TimeMeasurement (to be looked up in the hyperlinked GDD). TimeMeasurement consists of:
  - □ String1000 (base code)
  - □ languageCode, published in the GDD: *urn:gs1:gdd:cl:LanguageCode*; the attributeCode value must come from this list; please note – **no alternative code lists are allowed**.
  - □ codeListVersion (to be looked up in the GDD)
- The AVP placeholder location for this attribute is NutritionalInformationModule, class NutrientHeader

Example 2 of the Complex AVP

Selected columns from the Fast Track Attribute spreadsheet:

Attribute/Role	Module/Class to insert AVP into	AVP Type	Data Type/Secondary Class	Facets
itemPeriodSafeToUseAfterOpening	TradeItemLifespanModule	Complex MEASUREMENT_CODE	*TimeMeasurement* 1.	75

The above attribute mapped to XML:



```
<avpList>
   <compoundStringAVP attributeName="itemPeriodSafeToUseAfterOpening"
    attributeCode="DAY" codeListNameCode="MEASUREMENT_CODE"
    codeListVersion="1">30

  </CompoundStringAVP>
</avpList>
```

Name of the AVP attribute
Code value qualifying the attribute
Version number of the code list
Attribute value
Code list where the attributeCode must come from

- The AVP Type structure to be used for this attribute is Complex - CompoundStringAVP
- Name of the attribute comes from the Attribute/Role column in spreadsheet
- Attribute value conform to the Data Type + Facets column constraints; here it must be between 1 and 75 characters.
- The data type structure to be used for populating XML attributes should conform to TimeMeasurement (to be looked up in the hyperlinked GDD). TimeMeasurement consists of:
  - □ timeMeasurementUnitCode, published in the GDD: *urn:gs1:gdd:cl:TimeMeasurementUnitCode*; the attributeCode value must come from this list; please note – **no alternative code lists are allowed**.
  - □ codeListVersion (to be looked up in the GDD)
- The AVP placeholder location for this attribute is TradeItemLifespanModule

6.25 GPC Category Definition

The following rule allows setting of data pool’s own classification definitions:

The value for gpcCategoryDefinition MAY be the Brick definition or a data pool’s own version of the definition. The value SHALL be accurate but is not required. No validation is made on gpcCategoryDefinition apart from the schema imposed validation.

6.26 Operational Impact for Subscribing by GTIN

The Catalogue Item Synchronisation Business Message Standard states that for subscription, for every matching GTIN, independently from its level, all hierarchies will be returned.

For a given subscription:

- the matching process identifies Catalogue Items published to the GLN or TM of the subscription owner
- for each Catalogue Item, a notification is created including all dependent hierarchies.
- If the subscription is a synchronisation list, the hierarchy information included in the notification, will be limited to the GTINs maintained in the Synchronisation List.
- The notification is sent to the home data pool of the data recipient

If the subscription is done by GTIN:

1. Any subscription by GTIN registered at the Global Registry is stored. When a match occurs, that subscription is routed to all Source Data Pools that has that GTIN registered.
2. At the Data Pool level, subscriptions are received from the Global Registry and processed.
3. Recommendation: When subscribing using the GTIN as a criterion of the subscription, the Data Recipient SHALL use the GTIN of the highest level of the hierarchy that they would like to receive in the resultant Catalogue Item Notification. The Source Data Pool publication would result in the matching level of the hierarchy and all children items.

P (GTIN 3) [Pallet]

|

C (GTIN 2) [Case]

|

E (GTIN 1) [Each / Consumer Unit]

____

Hierarchy 1

4. A subscription for GTIN 1 would result in:
   the publication of only GTIN1 (assuming it passes validation)
5. A subscription for GTIN 2 would result in:
   the publication of GTIN 2 and its child GTIN1 (assuming they pass validation)
6. A subscription for GTIN 3 would result in:
   the publication of GTIN 3 and its children GTIN2 and GTIN 1 (assuming they pass validation)

7 Document Retention

There is a requirement for certified GDSN Data Pools to archive the actual messages sent and received (Minimum Archive Period [MAP]) in the following manner:

- 12 months for ALL file traffic into and leaving a GDSN-certified Data Pool.
    - Pass-through messages would not need to be stored by the receiving Data Pool (on the basis that the originating Data Pool is archiving to the 12 month requirement). However, the initiating subscription message from the Data Recipient would be stored by the home data-pool; as would any response messages sent by Data Recipient from the home Data Pool to the originating Data Pool. Definition of 12 months (MAP):

For Example: Beginning the 1st of the current month in the prior year. So if today is 17th February 2010, then the Data Pool archive is Feb 1st 2009 to Feb 17th 2010. In other words, the archive period will be at least 12 months (approaching 13 months at times).

By keeping this definition of the archive period, all Data Pools are aligned.

8 Disaster Recovery (DR) Capabilities

The GDSN has a Disaster Recovery capability for implementation upon a catastrophic failure of the GDSN Production systems. The decision for a DR situation will be communicated across as many channels as possible as quickly as possible.

There are a series of Internet Protocol addresses (IP's) that the Data Pools need to be aware of so that they can be incorporated in to their local firewalls. In the event of a GDSN Disaster Recovery situation, the network would cut over to the DR systems, which are currently in a different geographical location than the Production Servers. The DNS entries would change to these IP's. As long as these IP's are allowed in the firewalls at the Data Pools, there should not be any issues getting the service to run from the DR sites (other than a possible Data Pool restart). Below is the link to the public IP's we will be using for the GDSN DR servers.

http://www.gs1.org/docs/gdsn/support/GDSN_Disaster_Recovery_Site_2010April09.pdf

✓ **Note**: This information is presented for Certified Data Pools only and the information is contained within the Certified Data Pool Support Area, which requires authentication in order to access.

9 GPC Implementation and Integration in GDSN

9.1 GPC Implementation Process in GDSN

This section describes the process of GPC implementation in GDSN. This process also addresses the steps whereby a Trading Partner, or a Solution Partner on behalf of a Trading Partner, cannot find an appropriate GPC code.

A code SHALL be supplied as it is mandatory in the network for a Registry Catalogue Item to be registered:

- The network will validate codes against the list in Production in the GDSN.
- Valid codes include any deployed GPC code or “99999999” which is a temporary code for Segments that either have not been developed or for new products that do not fit with the current schema
- The network will not/cannot validate if the code used is valid in context (i.e. this Brick code is valid for this product)
- The type of code assigned depends on the relationship between the trading partners (TP) and solutions providers (SP)
- The TP will make every attempt to find the correct code

- If the correct code cannot be found, they must use "99999999" until an appropriate code is made available
- The next step would be to submit a Work Request (through the Global Standards Management Process (GSMP), monitor the progress, and once an appropriate code is made available, update their system

The GPC schema can only evolve when Work Requests are received. Every Work Request is processed individually and so categories can evolve separately at any time. This is the driving force behind GPC's publication release strategy which prohibits version numbers. Each publication is date stamped.

Updates to codes, once a product has been registered, are dependent on the Trading Partner and/or Solution Provider implementing changes as they are defined and published in the Delta reports. GDSN has a process for implementing GPC updates within the network. This means that GPC Brick Codes will not be implemented into GDSN as soon as they are released, but as per the publication cycle, usually within 6 months of publication.

9.2 GPC Integration Process into the GDSN

This section describes the process of GPC integrating GPC codes into the GDSN.

Upon completion of a GPC publication (twice per year) the GPC Service Provider sends the GDSN two files:

- XML Schema – A complete snapshot of all active nodes in all published standards in the GPC Schema at the point of publication. The purpose of this document is to provide a complete and correct view of what is contained in the GPC Schema at the point of publication.
- XML Delta – An XML document that contains all of the changes between the current and previous publications. The purpose of this document is to enable automatic changes/updates to GPC data contained in the GDSN.

GDSN will integrate/process the XML Delta document.

The XML Delta identifies changes to the GPC Schema by identifying the type of change using change codes and the level of change (Segment, Family, Class, etc.):

- Additions: The introduction of a new code. For every Addition the GDSN will add the new codes to their database.
- Modifications: When the code has NOT changed but the textual description HAS changed AND the definition has NOT changed. GPC Codes are NEVER modified and once deleted can never be re-used. If the Brick impacted has a significant definition change the normal process is to add new codes and delete old codes.
- For every Modification the GDSN will update the descriptions of the codes impacted in their database.
- Marked for Deletion: Codes that are marked for deletion remain as part of the GPC schema; however, the codes should not be used to code any items that are newly registered. This is step one of the two-step process to delete GPC Codes. At the next GPC publication these GPC codes that are 'marked for deletion' will become 'physical deletions'. One of the activities that is completed by the GS1 Global Registry as part of this process is to produce a data pool by data pool report of all items and subscriptions that currently use the GPC codes that have been identified as 'marked for deletion'. The intention is to use the period between releases to correct the items or delete the impacted subscriptions. The GPC codes that are marked for deletion become the list of codes that become 'physical deletions' in the next publication.
- Physical Deletions: Codes that have previously been identified as being 'marked for deletion' in the previous GPC publication move to 'physically deleted' from the GPC schemas. As the Publication is deployed into Production in the GDSN, the deletions occur in the GS1 Global Registry. As a result, any validation rules based on the GPC Codes would not be able to reference these deleted codes. As part of the deployment process in the GS1 Global Registry, any remaining items or subscriptions in the GS1 Global Registry are purged so that any references to these items and subscriptions are no longer present.

© 2018 GS1 AISBL

10 References

Source	Description
GS1 GDSN Standards	The latest version of all the GDSN-based GS1 Standards can be found here: *http://www.gs1.org/gdsn/3-1*
GDSN part of GS1 website	The information contained on the GS1 website provides additional content for the GDSN and its components: *http://www.gs1.org/gdsn*