M2M Guide atrify datapool

Machine to Machine Communication for the atrify datapool

Author:	**Selcuk Övüc**
Department:	**Product Management**
Version:	**3.2.6**
As of:	**February 2022**

Terms of use, disclaimer

This document (hereinafter "Material") is the sole property of atrify. For the purposes of these Terms of Use, atrify grants its customers, partners or other interested third parties (hereinafter "Users") the royalty-free, non-exclusive, non-transferable right, unlimited in time and space, to use the Material for their own purposes to the extent set out below. The User is not entitled to modify, reproduce, translate or use the Material for the purpose of resale, subletting, publication, demonstration or lecture. The Material may contain proprietary trademarks or logos which the User may not reproduce without the permission of the copyright owner. Any use of the Material within the limits described above must include the credit "© atrify GmbH".
Insofar as atrify is obliged to provide the Material on the basis of a paid contract with the User, the liability regulations laid down in the respective contract shall also apply to the use of the Material. In all other cases, atrify provides the Material, despite the greatest possible care in its creation, as a mere courtesy and is only liable for its content in accordance with § 516 et seq. BGB. Insofar as atrify is not obliged to provide the Material on the basis of a paid contract with the User, no rights can therefore be derived against atrify from the Material. In particular, atrify assumes no liability for the correctness, completeness and up-to-dateness of the Material. The User is solely responsible for the selection and use of the content and is aware that the Material is subject to constant further development. The above provisions also apply in favour of atrify's legal representatives and agents.

Imprint:

atrify GmbH
Maarweg 165, 50825 Cologne
T +49 221 93373 0 **F** +49 221 93373 199 info@atrify.com
Represented by Jochen Moll

Responsible for the content: Lars Schickner, atrify GmbH, Maarweg 165, 50825 Köln
Commercial register: Cologne Local Court HRB 45457
Sales tax ID: Sales tax identification number according to §27a sales tax law: DE22 4602462
Regulatory Affairs: Local Court Cologne

Table of contents

Overview of figures

Overview of tables

1. Introduction

1.1. About atrify

atrify is the platform for product content that enables more than 20,000 satisfied users in over 50 countries to share accurate and reliable content with their partners and consumers around the globe. With complementary expert services, atrify provides a holistic solution portfolio for transparency, compliance and cross-channel trading. atrify is 100% owned by GS1 Germany.

1.2. General note

This document contains two different signs, which point the reader to important issues:

 Check	The check sign sections in this document contain important information for Data Senders (DS) and Data Recipients (DR) on special impacts which both users have to take into account by using this document.
 Exclamation Mark	The exclamation point sections in this document contain special issues which must be considered while reading this document. Sometimes background information to special services, which atrify will provide for their customers, will be part of this section as well.

This document describes the processes and technical specifications concerning the message handling between atrify and the customers as well as atrify and the GDSN. It offers an overview of prerequisites, including technical connectivity, message types and communication channels. The document also describes a number of customer specific processes implemented by atrify.

Please note this document only concerns data transmission to the atrify datapool (atrify datapool).

1.3. atrify datapool

The atrify Data Pool atrify datapool is the central data management system, which connects all customer UI systems (Publishing and Approval) and messaging (M2M) users with each other and additionally provides a bridge to the GDS network.

atrify datapool is a GDSN certified Data Pool on version BMS 3.1 connected to 44 GDSN pools worldwide.[1]

[1] See "http://www.gs1.org/docs/gdsn/gdsn_certified_data_pools.pdf" for the list of GDSN certified Data Pools

The existence of different communities and different customer demands, results in a complex landscape of customer specific applications. To ensure interoperability between all applications, a central data management system is required, which provides a connection between all applications using GDSN standard messages (e.g. CIN, CIP, CIC etc.).

Furthermore this system provides a link to the GDS network and a single interface to the Global Registry and all GDSN certified Data Pools. For example, rather than connecting a Publishing for the German community with all existing retailer applications and the GDS Network – which would mean multiple connections and logic to know where the receiver of data is located – a single connection to atrify datapool is required. atrify datapool is in charge of delivering the data to the final recipient.

The following paragraph describes the specifics of the atrify Data Pool which is based on the GDSN standard.

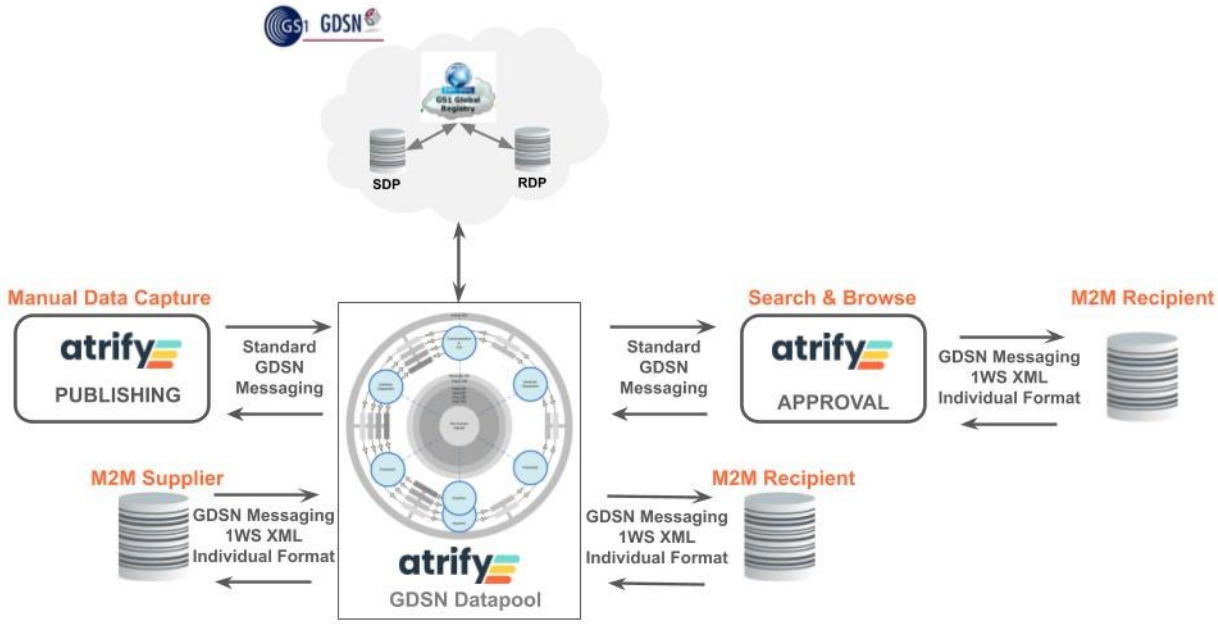


Figure 1: Pool architecture and message choreography for an item published to target market

atrify datapool uses standard GDSN messages to interact with the Publishing and Approval applications. The communication type used can be either AS2 or a simple file sharing connection depending on the location of each application.

Applications connected to atrify datapool acts as messaging (M2M) suppliers and retailers in terms of maintaining and retrieving data.

Messaging (M2M) customers are connected directly to atrify datapool and bypass the applications.

1.4. Contacts and support

For all questions regarding message exchange please contact the atrify Global support team.

General URL:
www.atrify.com

atrify datapool URL:
Production: https://datasync-prod.atrify.com
Pre-Production: https://datasync-test.atrify.com

Download center:
https://www.atrify.com/en/kundenbereich/download-center/

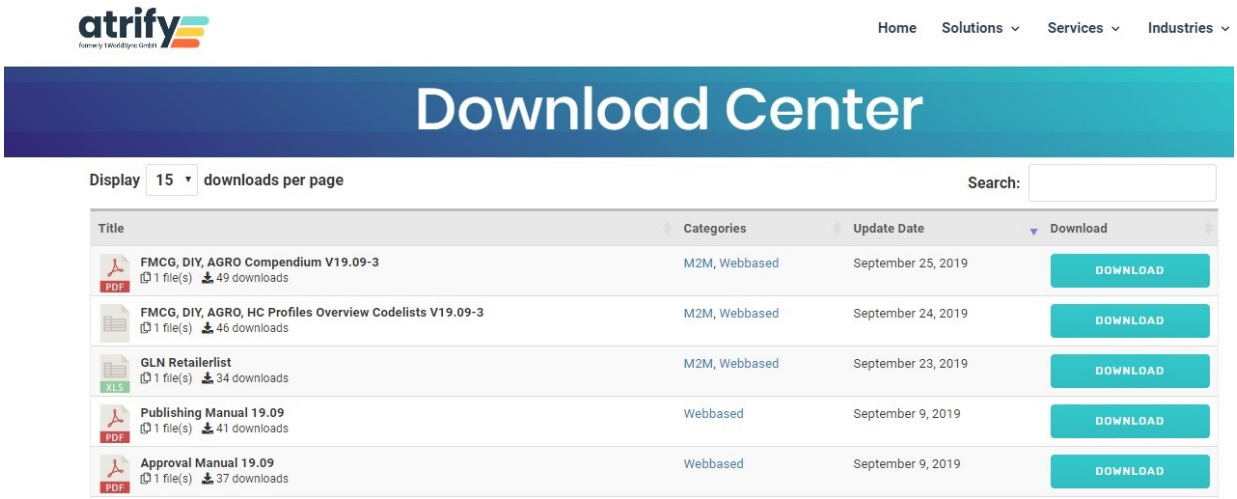


Figure 2: Download Center

Access to the download centre does not require any user name and password.

Contact details for support:
Please follow https://www.atrify.com/en/contact/ to get the contact details to reach out to the atrify support team globally. For M2M customers requiring technical support the following contact eMail should be used: technical-support@atrify.com

1.5. Further reading

Following is a list of necessary documentation (see table below), depending on the selected way of communication with the data pool.

#	Document & Link	Description	Published by	Helpful for…
1	GDSN XML Business Message Standards (BMS Version 3.1.x) http://www.gs1.org/gsmp/kc/gdsn/xml/xml_v_3	All relevant schemas to implement the GDSN BMS Version 3.1 can be found on the GS1 Org web side. All XML schemas are available and sorted by Business Message Standards, Module Schemas and Context Reports.	GDSN Inc.	M2M customers who want to implement BMS Version 3.1.x
2	GDSN Validation Rules (BMS Version 3.1) http://www.gs1.org/gdsn/gdsn-validation-rules/3-1	This Excel file outlines all GDSN validations which are provided by the atrify datapool. They enforce the minimum data quality needed in the network.	GDSN Inc.	M2M customers who want to implement and check their data quality in there in-house systems in accordance to the GDSN rules.
3	Fast Track Attributes (Approved Attributes) http://www.gs1.org/access-gdsn-standards (Scroll down to “Fast Track Attributes”)	This Excel file contains all new attributes which can be synchronized via AV-Pair extension (BMS 3.1) and were implemented by GDSN on a regular basis.	GDSN Inc.	M2M customers who want to check if new (needed) attributes are available to synchronize in the network.
4	Global Product Classification (GPC) http://www.gs1.org/gpc	Here the current and the future version of the GPC can be found. Depending on the implementation date the user has to check which version is the right one. Note: The GPC files are changed in the network twice a year: end of spring (May/June) and end of autumn (December).	GDSN Inc.	M2M customers who want to implement the GPC in their internal system or to create a mapping table from internal key to GPC.
5	Local Code Lists http://www.gs1.org/access-gdsn-standards (Scroll down to “Local Code Lists”)	In some countries (target markets) local code lists exist. The code lists which are affected by local rules can be found here. The lists are sorted by target market/ country.	GDSN Inc.	M2M customers who want to implement local codes in their systems or want to build mapping tables.
6	Global Data Dictionary (GDD) http://apps.gs1.org/gdd/SitePages/Home.aspx	The GDD contains an overview of all attributes which can be synchronized via GDSN with detailed information about the format, repeatability etc.	GDSN Inc.	M2M customers who want to implement BMS Version 3.1 and want to get detailed information on the attributes.
~~7~~	~~atrify Data Dictionary~~ ~~GPC_UPDATE_SPRING_2017_(EN)_(GDSN,_FMCG,_DIY,_AGRO)~~	~~The documents which can be downloaded contains files in Excel format:~~ ~~Overview of all attributes, codes and validations, GPC codes which can be synchronized via GDSN~~	~~atrify GmbH~~	~~M2M customers who want to implement BMS Version 3.1.x plus additional extensions and want to get detailed information on the attributes, their code lists and codes.~~
8	GDSN-XML-Subset-Schemas and Explanations FMCG-GDSN https://www.atrify.com/en/customer-section/download-center/ See here: FMCG Schema (V 19.06 or higher)	This file contains all needed schemas to implement the FMCG community profile of Germany and Austria. In addition a guide how to use the schema files and their comments.	atrify GmbH	M2M customers who want to fulfil the requirements for the FMCG community of Austria and Germany.
9	Compendium FMCG, DIY, AGRO	The document gives a detailed overview of all attributes and business rules which are	atrify GmbH	M2M customers who want to fulfil the requirements for the FMCG

#	Document & Link	Description	Published by	Helpful for…
	https://www.atrify.com/en/customer-section/download-center/ See here: Compendium FMCG & DIY 6.x	applicable for the GDSN user community FMCG in Austria and Germany		community of Austria and Germany.
10	Profiles Overview / Code lists FMCG, DIY, AGRO, HC https://www.atrify.com/en/customer-section/download-center/ See here: New GDSN System/ FMCG-GDSN Profile DE/AT/ Documentation Profiles Overview / Code lists FMCG, DIY, AGRO 6.x	The Excel file contains a detailed overview of all attributes, business rules and used code lists and their codes which are applicable for the GDSN user community FMCG in Austria and Germany.	atrify GmbH	M2M customers who want to fulfil the requirements for the FMCG community of Austria and Germany.

Table 1: Overview of helpful documents

In addition to the GS1 documentation recommended above we advise you to familiarize yourself with a number of documents, depending on the way you wish to transmit your data to the atrify Data Pool.

Pure GDSN customers should use the GDSN Implementation Package BMS 3.1.x (see # 1 - 7 in table above).

Customers who want to fulfil the data requirements for the FMCG Community profile of the target market Austria and Germany should use documents # 8 - 10 (see in table above).

2. GDSN basics[2]

GDSN is the Global Data Synchronization Network® managed by GS1 and GDSN Inc., which connects trading partners to the GS1 Global Registry® via a world-wide network of interoperable GDSN-certified Data Pools like the atrify datapool.

The GDSN concept essentially is a set of strict rules defining the communication between the pools. Within the network a commonly agreed message format and message choreography is used.

The GS1 Data Quality Framework and the GS1 Global Product Classification (GPC) forms a powerful environment for secure and continuous synchronization of good quality item data on the GDSN platform. Changes made to one company's database can automatically be sent to all trading partners who do business with them, so all have the same information in their systems.

GDSN certified atrify datapool makes it easy to globally synchronize item data, since almost any data content in the atrify pool is automatically transformed to GDSN compliant data.
Also, the appropriate messaging choreography within the networks handled automatically, e.g. registration of master data at the global registry, publication of data to authorized data requesting recipients, as well as update and correction management are all part of the GDSN based Data Pool service provided by atrify.

The current GDSN release version is BMS 3.1.x is the fundament for the whole message choreography.

2.1. GDSN party roles and functions

There are three types of actors in the GDSN, i.e. three roles:

the Global Registry,

the Data Pools, split into two types: Source and Recipient Data Pools,

and the Trading Partners, who are the initiating actors of each transaction: Data Supplier and Data Recipient.

The **Global Registry** is the central repository, where all item key information and all item subscriptions come together:

Every catalogue item registered there is identified through its triple key of GTIN, GLN and TM (Target Market), supplemented by its GPC (Global Product Classification), and linked to its registering Source Data Pool. Thus the Global Registry knows in which GDSN pool the item master data is stored.

Every request for item information (data inquiry, subscription) is stored at the Global Registry, and all registered items are filtered by the search criteria defined in it (GTIN, GLN, TM and GPC). If there are items that match the search criteria, the subscription is forwarded to the Source Data Pool, where these items are stored.

[2] See GDSN https://www.gs1.org/how-gdsn-works.

A Source Data Pool

- registers the items of the Data Suppliers at the Global Registry,
- maintains, validates and stores the complete item master data,
- receives, stores and executes the data inquiries (subscriptions) received,
- executes the data filtering, and forwards the master data of published and subscribed items (Pub-Sub match) to the Recipient Data Pools of the subscribing retailers.

A Recipient Data Pool

- manages the reception and forwarding of the data inquiries (subscriptions) from retail side,
- manages the reception and the forwarding of the received item master data to the data inquiring recipients,
- manages the reception and the forwarding of the responses/confirmations of the retailers (= Data Recipients) to the Source Data Pools of the data suppliers.

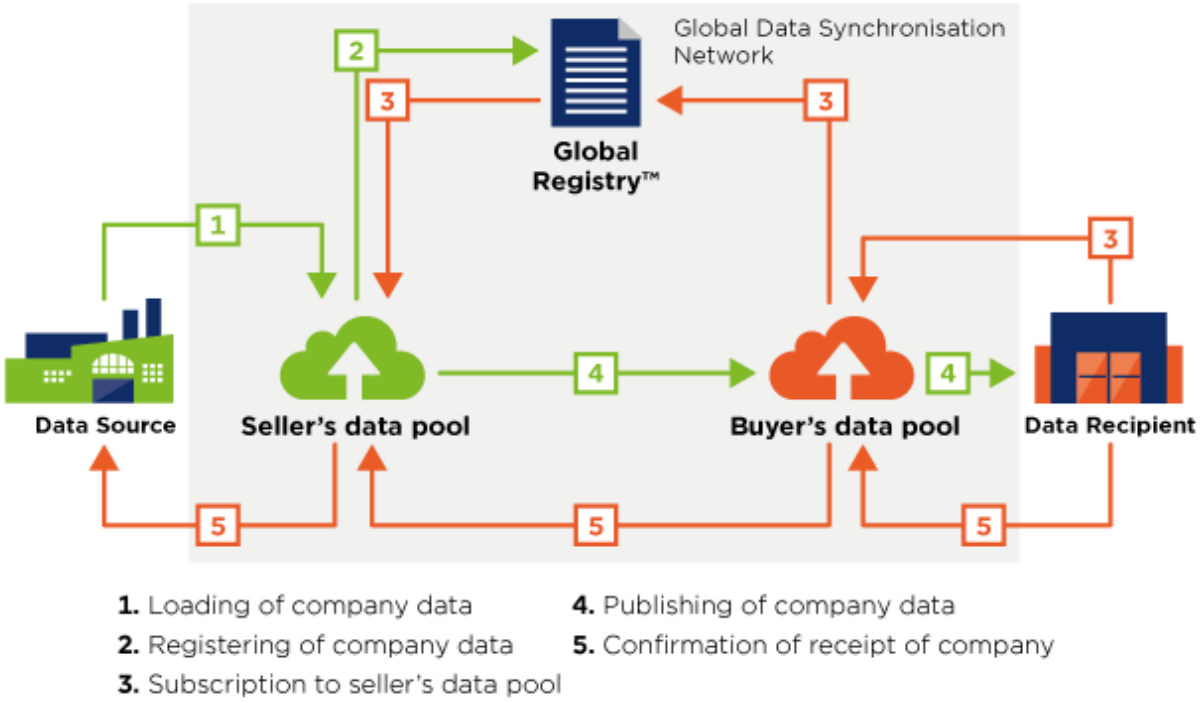


Figure 3: GDSN party roles and functions

Basically there are five steps to synchronize item data successfully via GDSN:

1 Load data: the seller (Data Supplier) registers and loads product and company data for data maintenance in his Data Pool (Source Data Pool).

2 Register data: key data is sent from there to the GS1 Global Registry.

3 Request data: the buyer (Data Recipient), through his Recipient Data Pool and the Global Registry, subscribes to receive a seller's product data (initial load and ongoing updates).

4 Publish data: the seller (Data Supplier) publishes his data in his Source Data Pool to certain buyers. When published, the Source Data Pool sends the data to the Recipient Data Pool of the buyer (Data Recipient), who has requested the data.

5 Confirm & inform: the buyer (Data Recipient) can return a confirmation to the seller (Data Supplier) - via both data pools - to inform him about the action he has taken, i.e. whether he has received, synchronized or rejected the data, or is demanding a data review. In the latter case the seller may adjust the contents for him.

2.2. GDSN message choreography

Within the GDSN the message choreography described below is used. All these messages have the defined GDSN XML message format.

The atrify pool as a GDSN pool uses these messages, as well. Therefore the **abbreviations** given below (e.g. CIN, CIS, CIP, CIC etc.) are used in further graphics and also in the document text.

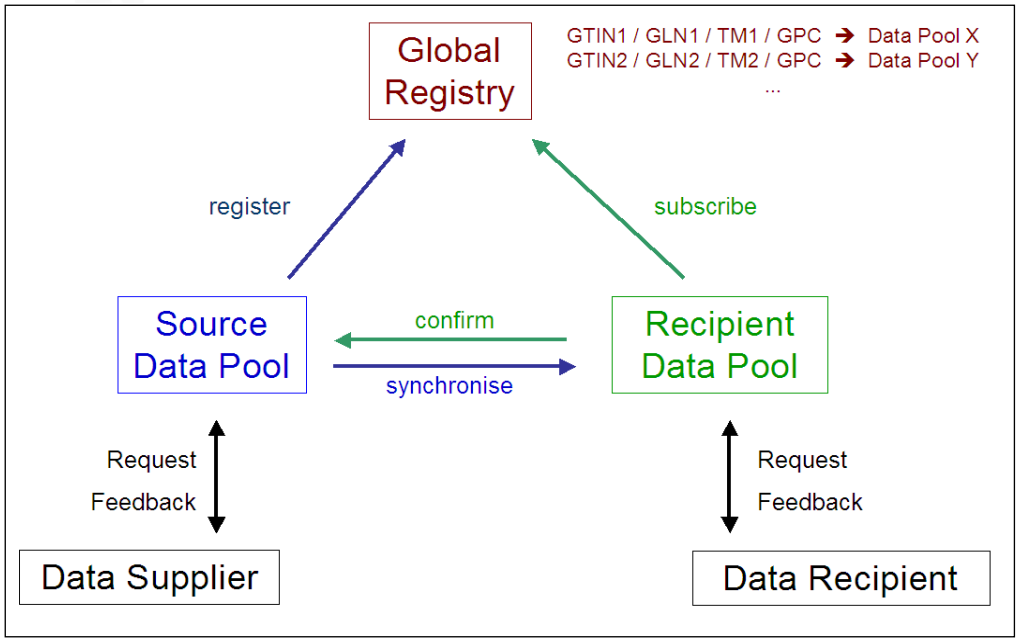


Figure 4: GDSN message triangle

The **registration message** (**RCI, Register Catalogue Item**) is used to register new items in the Source Data Pool at the Global Registry by sending their items keys (GTIN, GLN and TM), the GPC and the GLN of the Source Data Pool.

The **publication message (CIP, Catalogue Item Publication**) initiates the release of the item data for Data Recipients, who have subscribed for these data.

The **subscription message** (**CIS, Catalogue Item Subscription**) contains the filter criteria of a 'permanent' data inquiry, and is sent by the Data Recipient to the Global Registry via the Recipient Data Pool. The Global Registry forwards the subscription (including the requesting recipient's link) to the Source Data Pool of the items that match the filter criteria. There, the item stock is filtered accordingly and, if items are published to him, the master data (see CIN message below) (initial load and ongoing updates) is sent to the subscribing Data Recipient via the respective Recipient Data Pool.

A **Request for Catalogue Item Notification** (**RFCIN)** message is a one-time inquiry of item data (specified by filter criteria contained) that is already in the synchronization list of the data pool, thus has been synchronized with the Data Recipient already before, but is now needed again. An RFCIN can also be used to request data that was already rejected before (CIC-REJECT).
Other than the CIS message, this data inquiry is sent to the Source Data Pool via the Global Registry, but is not stored there, but is executed only once and then discarded. This means, that only a "reload" (without data rejected before) or an "initial load" (including data rejected before) is delivered, but no ongoing data updates.

In case of a PUB-SUB match (= publication matches subscription) at the Source Data Pool the complete catalogue item master data is directly sent to the Recipient Data Pool as a notification message (**CIN, Catalogue Item Notification**). The Recipient Data Pool forwards it to the subscribing Data Recipient.

As a result of data inquiry the Data Recipient via the Recipient Data Pool may reply with a confirmation message (**CIC, Catalogue Item Confirmation**), that is forwarded to the Data Supplier via his Source Data Pool. With this response, the recipient can give feedback about the acceptance status, including specific error messages, i.e. whether he has accepted (CIC-RECEIVED), synchronized (CIC-SYNCHRONIZED) or rejected (CIC-REJECT) the data, or is demanding a data review (CIC-REVIEW), i.e. adjustment or correction.

GS1 Response messages are sent out as an acknowledgement of receipt for transactions (message contents) that have been received and processed successfully. In this case the GS1 Response message will contain a <gS1Response> tag with <responseStatusCode>ACCEPTED</responseStatusCode> will be sent to the Data Supplier.

GDSN Response messages are also sent out, when GDSN errors have occurred while transactions have been processed (GDSN validation report). In this case the GS1 Response message will contain a <gS1Exception> tag having sub tags <gS1Error> <errorCode> <errorDateTime> <errorDescription> that will be sent to the Data Supplier.

2.3. Types of messages within the GDSN

Below is an overview of the most relevant[3] types of messages within the triangle (see figure above):

Type of message	Purpose	Sent by	Sent to
BPR, Basic Party Registration	Registration of party data	Source Data Pool	Global Registry
RCI, Register Catalogue Item	Registration of item data	Source Data Pool	Global Registry
CIP, Catalogue Item Publication	Publication of items to target market or retailer(s)	Data Source	Source Data Pool
CIHW, Catalogue Item Hierarchy Withdrawal	Delete GLN Publication to items retailers. And also used for hierarchy link corrections. **Note:** Deleting target market Publication is not possible with CIHW (bad design in GDSN!) as CIHW XML schema does not support target market deletions. atrify datapool still allows to use the legacy CIP Delete for GLN and Target Market publications.	Data Source	Source Data Pool Recipient Data Pool / Data Recipient
CIS, Catalogue Item Subscription	Subscription of item data	Data Recipient Recipient Data Pool	Global Registry Source Data Pool
CIN, Catalogue Item Notification	Notification of item modification or new items	Data Source Source Data Pool	Source Data Pool Recipient Data Pool / Data Recipient
CIC, Catalogue Item Confirmation	Confirmation of items received	Data Recipient Recipient Data Pool	Recipient Data Pool Source Data Pool / Data Source
RFCIN, Request for Catalogue Item Notification	Request for data after synchronization has taken place	Data Recipient Recipient Data Pool Global Registry	Recipient Data Pool Global Registry Source Data Pool

Table 2: List of important messages

[3] For a complete list please refer to the GDSN documentation that can be found at www.gs1.org/gdsn.

2.4. GDSN responses

There are four types of responses within the GDSN:

Type of response	Purpose	Sent by	In response to
GS1 Response (Type=Response)	Indicates that a transaction was performed successfully. Not a transaction itself, simply an indicator.	Data pool	CIN, CIP, CIHW, CIC, CIS
GS1 Response (Type=GS1 Error)	Indicates errors that occurred while transaction was being processed	Data pool	CIN, CIP, CIHW, CIC, CIS
Catalogue Item Registration Response	Indicates accepted of registration request	Global Registry	RCI
Party Registration Response	Indicates accepted of registration request	Global Registry	Basic Party Registration

Table 3: List of response messages

3. Technical connectivity

3.1. Prerequisites

atrify uses open Internet standards to exchange data with Trading Partners. In most cases, all you require is a certified Internet browser to the atrify web applications (see chapter Browsers for details of supported browsers).

The trading partners are responsible for securing the following prior to exchanging data with atrify..

Obtain appropriate hardware and messaging software than can:

- Send and receive messages over the Internet using AS2, https or sFTP
- Translate and convert messages from/into the GDSN XML message types
- Send and receive messages to and from your back-end systems
- Implement support for the required business data types in the messaging software
- Connect to atrify using AS2 protocols/methods, https or sFTP
- Incorporate electronic messaging into your business processes and back-end systems
- Test connectivity with atrify

3.1.1. Browsers

More advanced applications require the exchange of necessary data between atrify and your back-end ERP systems.

atrify supports the most common browsers Google Chrome, Mozilla Firefox and Windows Internet Explorer 11. In principle the user should make sure to use the recent browser versions.

3.1.2. Firewall requirements

atrify Integration Hub can connect to any IP address provided it is either:

- AS2 over port 80/443 – requires digital certificate authentication.
- sFTP over port 5000

AS2 over port 80/443 can be arranged on a case by case basis. This does not require implementation of additional firewall rules for each trading partner on atrify side.

Especially, in order to send/receive transactions from atrify, trading partners must configure their firewall(s) to allow/accept connections from/to the following atrify IP addresses:

atrify datapool Instance	URL (http:// or https://)	IP Address
atrify datapool TEST	as2-datasync-test.atrify.com/as2	89.202.37.4
atrify datapool PRODUCTION	as2-datasync-prod.atrify.com/as2	89.202.37.4

atrify accepts files in GDSN XML format. Please refer to your atrify implementation documentation for details on the actual payload formats to assist in your back-end integration mapping.

Some of the response codes are listed below:

200 "OK" – Transaction completed successfully

400 "Bad request" – Error in request header or unrecognized content; this may mean that the format of the URL or request header information is not correct

401 "Unauthorized" – Authentication failure; this means that the User ID and/or password are invalid and, in rare cases, may indicate a problem with the HTTP/S server at the Exchange

404 "Not found" – Cannot find the specified URL

500 "Server error" – Indicates that the server is inoperative or is currently offline

3.2. Using EDIINT AS2

EDIINT stands for “Electronic Data Interchange-Internet Integration” or “EDI over the Internet”. EDIINT is a standard defined by the Internet Engineering Task Force (IETF) that defines a protocol for using the Internet to securely exchange business data (EDI, XML, or other). Please refer to http://www.ietf.org/html.charters/ediint-charter.html for general information on EDIINT.

There are two versions of the EDIINT standard: AS1 (SMTP based protocol) and AS2 (HTTP or HTTP/S based protocol). atrify only supports the EDIINT AS2 version in production at this time and uses technology that has successfully completed the Secure B2B INT AS2 Conformance Validation Test. The UCC (Uniform Code Council)-sponsored test was conducted by the Drummond Group to validate the ability of software vendors to interoperate with one another to communicate EDI data via the EDIINT AS2 standard. For more information on the interoperability test administered by the Drummond Group, please refer to http://www.drummondgroup.com. The following list outlines the primary feature specifications of the current production atrify EDIINT AS2 implementation:

- Exclusively uses the S/MIME (Secure/Multipurpose Internet Mail Extensions) version 2 cryptographic format to package, encrypt, and provide a digital signature to outbound data and to unpack, decrypt, and verify the authenticity of inbound data.
- Uses SHA-1 hash algorithm to sign outbound messages and verifies inbound messages that were signed with either SHA-1 or MD5.
- Supports custom encryption type and key length settings per trading partner. The choices are: Triple DES, DES, RC2(40), RC2(68), and RC2 (128).
- Delivers outbound messages to trading partners as signed, encrypted, signed/encrypted, or plain text.

- atrify AS2 sends receipt MDN messages back to the sending party if they are requested.
- atrify AS2 requests MDN messages from trading partners upon successful receipt of the EDI-INT AS2 message. Per default atrify expects synchronous signed MDN receipts. But asynchronous MDNs are also supported.
- The payload Content-Type of the document depends on the transaction being sent or received. For most transactions, the Content-Type will be application/xml.
- Set number of retry to 1.

3.2.1. Sending GDSN XML to atrify Using EDIINT AS2

For inbound EDIINT AS2 transactions, atrify supports either:

- HTTP/S over port 443, with Basic Authentication (User ID/ Password)
  OR
- HTTP with S/MIME encryption over port 4080, without authentication, but controlled firewall-to-firewall access and digital certificate being exchanged.

The detailed information can be taken out of the following table:

Information for immediate assistance regarding machine to machine communication		
Technical Contact Email	connectivity@atrify.com	
Technical Contact Telephone Number	+49 221 93373 -363 or -111	
AS2 Connection Parameters of atrify (Production System)		
AS2 Protocol and Software		
AS2 Identifier (AS2-From)	4049111100007	
Used Software / Version	/n software inc. IP*Works! EDI AS2 Component 9.0.4729	
HTTP URL	http://as2-datasync-prod.atrify.com/as2	Use this URL in case you connect by HTTP
HTTP Port	80	Use this port in case you connect by HTTP
HTTPS URL	https://as2-datasync-prod.atrify.com/as2	Use this URL in case you connect by HTTPS
HTTPS Port	443	Use this port in case you connect by HTTPS
Security Issues		
Outbound IP	89.202.37.4	
Compression (Yes / No)	No	
Type of compression	./.	
Signature (Yes / No)	Yes	
MDN required (Yes / No)	Yes	
Type of Connection (Synchronous / Asynchronous)	Synchronous	
Document Encryption (Yes / No)	Yes	
Type of Encryption (Triple DES / RC2 / ARC4 / DES)	Triple DES	
Digital Communication Signature (SHA1 / MD5)	SHA1	

Name of Certificate	as2-datasync-prod_atrify_com_public.cer	
Certificate Expiration Date	September 11th, 2020	
AS2 Connection Parameters of atrify (Test System)		
AS2 Protocol and Software		
AS2 Identifier (AS2-From)	4049111170017	
Used Software / Version	/n software inc. IP*Works! EDI AS2 Component 9.0.4729	
HTTP URL	http://as2-datasync-test.atrify.com/as2	Use this URL in case you connect by HTTP
HTTP Port	80	Use this port in case you connect by HTTP
HTTPS URL	https://as2-datasync-test.atrify.com/as2	Use this URL in case you connect by HTTPS
HTTPS Port	443	Use this port in case you connect by HTTPS
Security Issues		
Outbound IP	89.202.37.4	
Compression (Yes / No)	No	
Type of compression	./.	
Signature (Yes / No)	Yes	
MDN required (Yes / No)	Yes	
Type of Connection (Synchronous / Asynchronous)	Synchronous	
Document Encryption (Yes / No)	Yes	
Type of Encryption (Triple DES / RC2 / ARC4 / DES)	Triple DES	
Digital Communication Signature (SHA1 / MD5)	SHA1	
Name of Certificate	as2-datasync-test_atrify_com_public.cer	
Certificate Expiration Date	September 11th, 2020	

Table 4: AS2 connectivity data

atrify will respond with a synchronous signed MDN message upon successful receipt of the EDIINT message. If the MDN is not received, the partner should consider the message delivery failed. atrify will provide the partner with its X.509 compliant public digital certificate to allow the partner to verify atrify's digital signature on the MDN message.

3.2.2. Receiving atrify (GDSN) XML from atrify Using EDIINT AS2

The payload stream will be signed and encrypted when delivered to the partner. atrify will provide the partner with its X.509 compliant public digital certificate to allow partner to verify atrify document signature. The partner must provide atrify with their X.509 compliant digital certificate to allow atrify to encrypt the message being sent to the partner.

The Content-Type setting should be:

Content-Type=”application/xml”

A synchronous signed MDN message is expected by atrify upon successful receipt of the EDIINT message. Otherwise, data delivery at the Exchange will be considered failed. The partner is required to provide an X.509 compliant public digital certificate to allow atrify to verify the partner’s digital signature.

The partner must provide atrify with a URL for sending messages to their interoperable EDIINT AS2 system.

3.2.3. AS2 authentication for outbound messages

AS2 Communication also provides basic authentication for outgoing messages on the HTTP/HTTPS-Layer. The software checks if the userid/password is specified as part of the AS2 URL, e.g. given the URL "http://james:secret@as2.customer.com:443/gateway", the userid is "james", and the password is "secret”.

Encryption and signature are options for AS2 Comchannel.

3.3. Encoding

UTF-8

UTF-8 is an encoding of Unicode (more precisely, one of several possible encodings), which is optimized for using mostly Latin characters. (Please note, that there is a big difference between a character set like Unicode and its actual encoding, e.g. "UTF-8" etc. Within UTF-8 all Latin characters are encoded with a single byte. But obviously, not all 63487 Unicode codes can be encoded with a single byte. All non-Latin characters must be encoded with two, three or even four bytes.

3.3.1. Restrictions for Unicode

The following limitations for Unicode codes apply to the atrify system:

Restrictions caused by the XML specification

The use of the following is prohibited:

All so called "C0 Controls", i.e. ASCII characters with code below 32, except the TAB (#x9), the LINE-FEED (#xA), and the
CARRIAGE-RETURN (#xD):

- #0 - #x8 (decimal: 0 - 8)
- #xB (decimal: 11)

- #xC (decimal: 12)
- #xE - #x1F (decimal: 14 - 31)

The "Surrogate Area"

The Unicode standard does not define any characters in this area and it never will, since these codes are reserved for use with UTF-16 encoding:

- #xD800 - #xDFFF (decimal: 55296 - 57343)

Codes #xFFFE and #xFFFF,

These are guaranteed not to be Unicode characters at all.

#xFFFE, #xFFFF (decimal: 65534, 65535)

Beyond that, any Unicode characters can be used.

3.3.2. Non-printable characters

There are several codes defined by Unicode which denote non-printable characters. Some of them may be useful, assumed that the software involved knows how to deal with them, e.g. PARTIAL-LINE-FORWARD (#x8B) or REVERSE-LINE-FEED (#x8D). But most of them definitely won't be useful. Codes like SET-TRANSMIT-STATE (#x93) or DEVICE-CONTROL-STRING (#x90) should be avoided generally in textual data. Similarly, codes like PRIVATE-USE-ONE (#x91) [see http://www.fileformat.info/info/unicode/char/0091/index.htm] and PRIVATE-USE-TWO (#x92) [see http://www.fileformat.info/info/unicode/char/0092/index.htm] are obviously inappropriate for use with global messaging.

As a result, atrify **strongly discourages** using non-printable characters, unless there are very good reasons to do so. That specifically applies to all of the "C1 Controls" codes:

- #x80 - #x9F (decimal: 128 - 159)

3.3.3. Further recommendations

There is another class of pseudo characters, the Combining Diacritical Marks, which may (and most probably will) cause trouble. Diacritical marks are detached ancillary glyphs which would normally be part of a composed letter, the dots of an umlaut for example.

Only in very particular situations, are such pseudo characters meant to follow another letter, which usually would not come with additional diacritical marks. For example, if you wanted an *M* with two dots above it, you could express it in UTF-8 as

#4D #CC #88

where #4D stands for the M and the #CC #88 stands for the additional dots.

Unfortunately, this flexible approach chosen by the Unicode committee carried with it a new source of issues, namely the ambiguity to encode ordinary umlauts in more than one way, e.g.

#C3 #B6

which is the ordinary *ö*, contrary to the sequence

#6F #CC #88

which denotes an *o* with combined dots. Although the pool software is fine with synchronizing such combined characters via messaging, it is important to be aware that it may cause trouble when such characters are received by a retailer. Furthermore, the pool software will regard the ordinary *ö* and the combined *o* with umlaut dots as completely different characters.

atrify therefore discourages using combining diacritical marks, except in situations where no simpler alternative is available, e.g. niqqud, taškīl or similar diacritics.

4. Functional Connectivity

This chapter describes the functional connectivity with the atrify datapool. In this chapter you will find a list of the message types and their particular roles regarding retailer and supplier, respectively.

4.1. Message types and processes

4.1.1. Retailer view

A retailer who wishes to subscribe to item data must send a **CIS** or **RFCIN** to the atrify datapool. atrify customers can also set up subscriptions (CIS and RFCIN) through the WebSubscription tool. The Subscription is forwarded to the Global Registry which in turn will forward it to any Source Data Pool with matching criteria. The results of a subscription will be transmitted via the predefined communication channel via AS2, HTTPS or sFTP. Please see chapter Technical connectivity for details.)

4.1.2. Supplier view

The supplier transmits data to the atrify datapool via **CIN**. Upon receipt, the Data Pool performs validations (see chapter 9 for details). The atrify datapool automatically registers the new trade item data with the Global Registry by sending a **RCI**. The Global Registry confirms this message with a **CIRR** provided the GTIN is question is not yet registered. If the GTIN is already registered, the Global Registry responds with an **Exception**.

To make the data available to either selected retailers or a whole target market, the trade items need to be published. To do this the supplier sends a **CIP** to the atrify datapool.

4.2. Action code handling

Usually a new item has the action code / document command header set to ADD. Any updates will come with CORRECT or CHANGE_BY_REFRESH. In case an incoming CIN has the action code / document command header contains CORRECT or CHANGE_BY_REFRESH and the related items are not stored in atrify datapool, the item is considered new and the action code for the items will be stored as "ADD" in the atrify datapool data base.

If a recipient receives an already existing item with document command header CORRECT or CHANGE_BY_REFRESH the first time, atrify datapool will deliver the appropriate CIN message with action code ADD to the recipient.

4.3. IsReload flag in CIN

The isReload 'true/false' tag in the CIN message is used to send data to retailers as an initial load or new item add.

4.3.1. Supplier view

Initial load

An initial load means an item already exists in the retailer system but it is the first time the supplier is sending the data through the data sync process. To send an item to the retailer using the CIN the supplier should:

Send CIN ADD with isReload = true
Send CIP ADD
The atrify datapool then sends

CIN ADD with isReload = true to retailer

OR

Send CIN ADD with isReload = false
Send CIP ADD with isReload = true

atrify datapool then
sends CIN ADD with isReload = true to retailer

New item add

A new item add is used for items that do not exist in the retailer's system. To send an item to the retailer using the CIN the supplier should:

Send CIN ADD with isReload = false
Send CIP ADD

The atrify datapool then
sends CIN ADD with isReload = false to retailer

Delete item
To delete an **item** a supplier has two options in GDSN:

1. Discontinue Items
2. Cancel Items

Discontinue Items
To express the date when an item is no longer to be manufactured the supplier has to set two discontinue dates as follows:
CIN/catalogueItemState class: discontinueDateTime (Messaging Process relevant) **AND**
CIN/tradeItemSynchronisationDates class discontinuedDateTime (Content Relevant)

Note: Make sure that both date entries have the same values!

Cancelling Items
An item must be cancelled if the appropriate physical product has never been delivered to the recipient, but master data did flow. In this scenario the supplier will just cancel the data by setting two cancel dates as follows:
CIN/catalogueItemState class: cancelDateTime (Messaging Process Relevant) **AND**
CIN/tradeItemSynchronisationDates class cancelledDateTime (Content Relevant)

Note: Make sure that both date entries have the same values!

4.3.2. Retailer view

If a retailer receives a CIN ADD with documentStatusCode = “ORIGINAL” having isReload = false from atrify datapool then the data has been sent as a new item add.

If a retailer receives a CIN ADD with documentStatusCode = “COPY” having isReload = true from atrify datapool then the data has been resent again. Means the recipient has previously received the item and has requested the same item again (triggered via RFCIN by retailer or via CIP with isReload extension by the supplier.

5. User Interfaces Publishing and Approval

5.1. Supplier User Interface Publishing

Suppliers without any AS2 machine-to-machine connectivity to the atrify datapool can use the atrify supplier user interface Publishing. Usually small and medium enterprises make use of the Publishing to load and publish data by manual entry or via Excel uploading. The Excel format has particular restrictions and specifications to be met in order to get the data accepted by the system. Also maintaining digital assets like images and documents and Price data can be done within the Publishing. Customers wishing to transmit data via Publishing must have access to the Publishing, if you do not have access to Publishing, please contact the Sales Department. For further reference please check the Publishing user manual (See Chapter 1.5 “Further reading”).

5.2. Recipient User Interface Approval

Approval is a web application for recipients that allows to search and browse for synchronised item and price data. Approval also supports Excel and PDF download for further analysis and documentation purposes. Furthermore, Approval allows CIC auto-responding and community related extra checks beyond the GDSN basic validation rule set. This is mainly used in some specific industries and/or target markets (e.g. DIY, FMCG, Healthcare, Germany, Austria, The Netherlands). As a summary the following features are supported:

- Search & Browse for items & prices
- Download items in Excel or PDF format
- View item history
- Subscription list to create subscriptions online
- CIC Auto responding
- Extra Validations

To get access to the Approval please contact the Sales Department. For further reference please check the Approval user manual (See Chapter 1.5 “Further reading”)

Operating system	No restrictions
Broad band internet connection	At least 1 MBit DSL
Browser	Google Chrome Mozilla Firefox Windows Internet Explorer 11
Javascript enabled	Required
Accept cookies	Required
Accept CSS	Required
Accept popups	Required
Allow images	Required
Screen resolution	Min 1024*768

Table 5: System requirements for Publishing / Approval

6. Communication atrify to customer

6.1. Messages

The following message types can be sent to the customer by atrify:

GS1 Response Message, or CIC. Users can receive messages via a predefined communication channel, i.e. AS2.

The communication channel and format is set up for users in the WebAdmin upon registering with atrify. If you wish to change the type of communication channel and/or format, please contact the support team (see chapter Contacts and support for details).

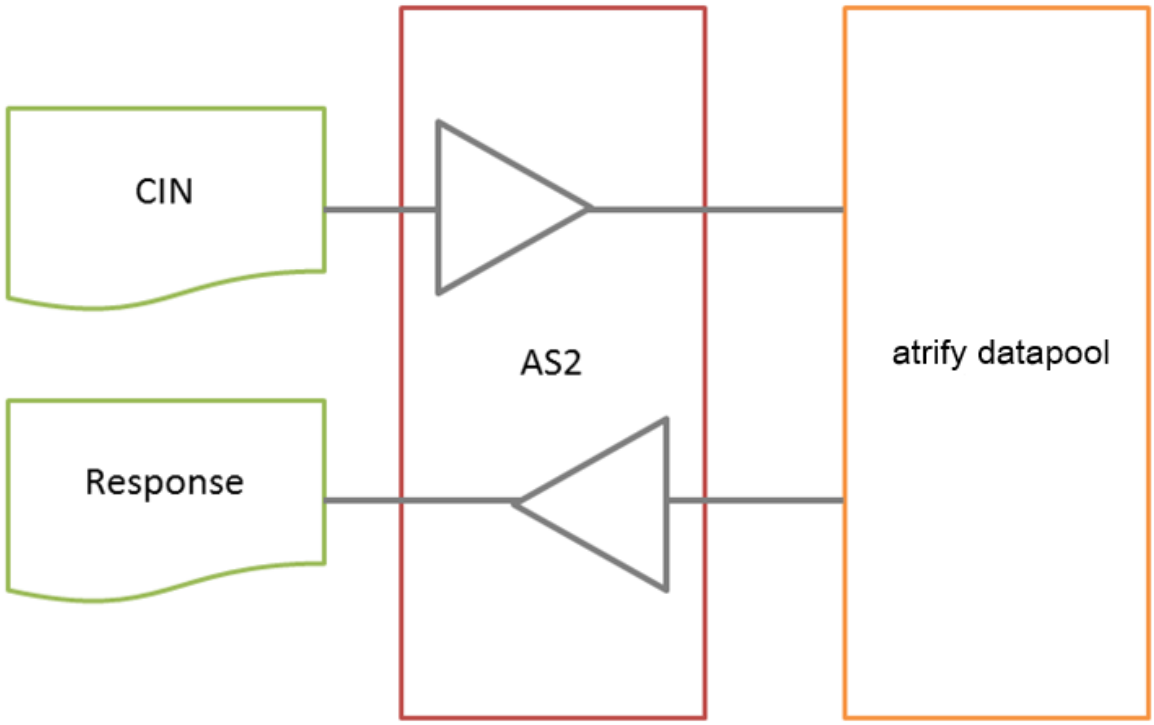


Figure 5: Messages between customer and atrify

6.2. atrify datapool Response handling for basic GDSN use cases

Following table shows the atrify datapool response handling (GDSN-Response / GDSN-Exception)

#	Catalogue Item Publication (CIP) Handling		
	Use Case	**Description**	**atrify datapool reacts with…**
1	CIP “ADD” without item	DS sends CIP with action request ADD and the item which should be published isn´t in the Data Pool.	gS1Exception
2	CIP “ADD” with item	DS sends CIP with action request ADD and the item which should be published is in the Data Pool.	gS1Response
3	CIN – CIP “ADD” - CIS	CIS arrives on an already published item.	gS1Response (CIS)
4	CIN - CIS - CIP "ADD"	CIS arrives before item will be published.	gS1Response (CIS)
5	CIP “ADD” following CIP “ADD”	DS sends same CIP “ADD” twice.	gS1Response (CIP)
6a	CIHW "PUBLICATION_WITHDRAWAL" without item	DS sends CIHW with hierarchyDeletionReasonCode PUBLICATION_WITHDRAWAL the item for which the publication should be deleted isn´t in the Data Pool.	gS1Exception
6b	CIP “DELETE” without item	DS sends CIP with action request DELETE and the item for which the publication should be deleted isn´t in the Data Pool.	gS1Exception
7a	CIHW "PUBLICATION_WITHDRAWAL" with item, CIP „ADD“ is missing	DS sends CIHW with hierarchyDeletionReasonCode PUBLICATION_WITHDRAWAL, but never sent a sufficient CIP with action request ADD.	gS1Exception
7b	CIP "DELETE" with item, CIP „ADD“ is missing	DS sends CIP with action request DELETE, but never sent a sufficient CIP with action request ADD.	gS1Exception
8a	CIHW "PUBLICATION_WITHDRAWAL" with item, having valid CIP "ADD" stored in the atrify datapool	DS sends CIHW with hierarchyDeletionReasonCode PUBLICATION_WITHDRAWAL. Before DS sent a valid CIP with action request ADD.	gS1Response
8b	CIP "DELETE" with item, having valid CIP "ADD" stored in the atrify datapool	DS sends CIP with action request DELETE. Before DS sent a sufficient CIP with action request ADD.	gS1Response
9	CIP "ADD" to GLN with 1WS extension for “isReload-Functionality”, "isReload = true"	DS sends CIP with action request ADD and isReload flag = true.	gS1Response (isReload flag will set to “true” in the first outgoing CIN to the recipient.)
10	CIP "ADD" to GLN with 1WS extension for “isReload-Functionality”, “isReload = false"	DS sends CIP with action request ADD and isReload flag = false.	gS1Response (isReload is “false” per default)
#	**Catalogue Item Notification (CIN) Handling**		
	Use Case	**Description**	**atrify datapool reacts with…**
1	CIN “ADD” and CIN “ADD”	DS sends CIN with action request ADD followed by the same item with same action request ADD.	gS1Response
2a	CIN “CHANGE_BY_REFRESH” or “CORRECT” to change hierarchy having send CIHW "LINK_CORRECTION" before	DS sends update to an item hierarchy after having put the status “ON_HOLD” by sending a CIHW "LINK_CORRECTION" in advance.	gS1Response
2b	CIN “CHANGE_BY_REFRESH” or “CORRECT” to change hierarchy without having send a CIHW "LINK_CORRECTION" before	DS sends update to an item hierarchy without having put the status “ON_HOLD” by sending a CIHW "LINK_CORRECTION" in advance. This is not allowed since the MjR 3	gS1Exception

#	Catalogue Item Subscription (CIS) Handling		
	Use Case	**Description**	**atrify datapool reacts with…**
1	CIS “DELETE” on non-existing CIS	DR sends CIS with action request DELETE on non-existing CIS.	gS1Exception
2	CIS “ADD” and CIS “ADD”	DR sends same CIS with action request ADD twice.	gS1Response (no impact)
3	CIS “DELETE” and CIS “DELETE”	DR sends same CIS with action request DELETE twice.	gS1Exception
#	**Catalogue Item Confirmation (CIC) Handling**		
	Use Case	**Description**	**atrify datapool reacts with…**
1	CIC with one of the following status without item in Data Pool: “RECEIVED” “SYNCHRONISED” “REVIEW” “REJECTED”	DR sends CIC with one of the four statuses and the item which was tagged with that special status isn´t in the Data Pool.	gS1Exception
2	CIC with one of the following status with item in Data Pool: “RECEIVED” “SYNCHRONISED” “REVIEW” “REJECTED”	DR sends CIC with one of the four statuses and the item which was tagged with that special status is in the Data Pool.	gS1Response
3	CIC ”REJECTED” followed by CIC “RECEIVED” (item in Data Pool)	DR sends CIC with status “RECEIVED” after CIC with status "REJECTED" for an item which is in the Data Pool.	gS1Response
4	CIC “RECEIVED” followed by CIC “RECEIVED” (item in Data Pool)	DR sends CIC with status “RECEIVED” after CIC with status “RECEIVED” for an item which is in the Data Pool.	gS1Response

Table 6: Message Handling atrify datapool

7. Communication customer to atrify

7.1. Messages

7.1.1. Trading Partner Dependent Attributes (TPD)

Values for an attribute can vary depending on the relationship with the party receiving the data. The Trading Partner Neutral (TPN)/Trading Partner Dependent (TPD) status indicates this rule:

Trading Partner Neutral Attributes: The condition Trading Partner Neutral is applied to any attribute whose value is independent of a buyer and seller relationship. An attribute, which has the condition Trading Partner Neutral, can have only one set of values.

Trading Partner Dependent Attributes: The condition Trading Partner Dependent is applied to any attribute whose value is dependent on a buyer and seller relationship. An attribute, which has the condition Trading Partner Dependent, can have only one set of values per GLN of Party Receiving Private Data. These are attributes whose value is dependent on a specific point-to-point agreement between a buyer and a seller.

Trading Partner Neutral and Trading Partner Dependent Attributes: An attribute which has the condition Trading Partner Neutral and Trading Partner Dependent, can have only one set of values for the Trading Partner Neutral value AND one set of values per GLN of Party Receiving Data (TPD).

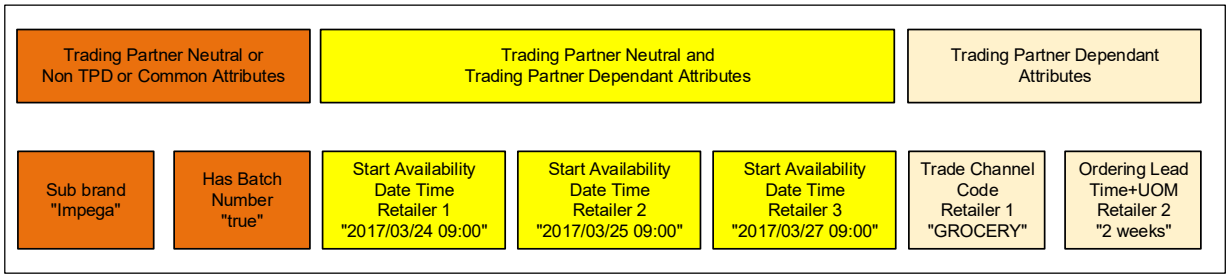


Figure 6: Example of TPD and Non TPD data

The atrify data pool stores all TPD attributes of an item for a dedicated recipient GLN in a so called TPD-GLN blob. To send TPD values for a recipient the supplier must send a separate CIN having the “partyReceivingPrivateData/GLN” information populated. If this information is missing the item is treated as neutral. On exporting an item to the recipient, the data pool checks whether TPD-GLN blobs for that recipient exists and will merge those blobs with the neutral data. If not, only neutral data will be exported to the recipient.

Important Note: Deleting those TPD-GLN blobs is currently supported for Publishing users only. The ability to delete via Hybrid or m2m will follow. Also for 3PE (Third party enrichment) items TPD blobs cannot be deleted. In this case the 3PE provider must delete the entire item record via CIN-DELETE and resend the 3PE item.

7.1.2. Deviation from BMS T.P.D. Definition

Until further notice the atrify datapool will not apply the changes to the below listed attributes moved from Trading Partner Neutral and Trading Partner Dependent with BMS Release 3.1.15.

- carcinogenicMutagenicReprotoxicTypeCode & regulatedChemicalTypeCode (ChemicalRegulationInformationModule)
- certificationExecutionCountryCode (CertificationInformationModule, DietInformationModule)
- clinicalSizeMeasurementPrecisionCode & clinicalSizeValueMaximum (HealthcareItemInformationModule)
- complementAddress, poBox, streetNumber (FoodAndBeverageIngredientModule / GDSN_Ingredient
  Core GDSN_ManufacturerOfTradeItem, GDSN_PartyInRole, GDSN_TradeItemContactInformation)
- globalModelDescription (Core)
- languageSpecificPartyName (Core)
- warningsOrContraIndicationDescription (group already exists as non TPD, HealthcareItemInformationModule)

7.1.3. Implications for an Inbound CIN

A user must send a CIN for each set of TPD values (defined as Trading Partner Neutral or for a special GLN of Party Receiving Private Data).

For CINs with TPD values (for a special GLN of Party Receiving Private Data) the appropriate Party Receiving Private Data GLN must be set in the message. If no Party Receiving Private Data GLN is set, the TPD value will be processed like a neutral value, which is valid for non-specific data receivers.

When transmitting TPD values no special order is required. It is not necessary to send the neutral TPD value in the first CIN.

The trade item status for the first CIN must be “ADD”, the following can be “ADD”, “CHANGE” or “CORRECT”. “ADD” after “ADD” is possible, currently there is no special validation.

Important note: Once an item was declared trading partner dependant for a recipient it will remain like that. It is not possible to make an item back neutral for those retailers again.

7.1.4. Attribute Value Pair (AVP) Handling

AVP attributes are either specified as neutral or trading partner dependant (TPD). As there is no restriction regarding the usage of attributes in the atrify datapool, trading partners can agree upon proprietary AVPs to be exchanged bilaterally or with a dedicated number of recipients, without being officially supported by the atrify datapool. For a list official AVP please refer to the GDSN page for Fast Track attributes or the atrify documentation.

This has the following implications with regards to the AVP handling when updating existing items in the datapool.

Official AVPs supported in GDSN and/or specified by atrify communities

GDSN defined AVP (Fast Track Attributes) and AVP specified by the atrify communities will be supported along with the defined TPD status. This means:

- AVPs specified as **Non-TPD** will be delivered to all recipients that are supposed to receive the item. Any item updates with changes for those AVP values will automatically be transmitted to all recipients.
- AVPs specified as **TPD** will only be delivered to the explicitly dedicated recipients. Any item updates with changes for those AVP values will be updated for those TPD recipients only.

Non atrify official / bilaterally used AVP

Any AVP unknown in the atrify data model, will be treated as TPD always. Here the same rule applies for the data delivery mechanism as for standard TPD attributes:

- AVP will be delivered to the appropriate TPD recipients only.
- Updates for existing Pub-Sub Matches will go to the dedicated TPD recipients as populated in the CIN message.
  If an update is meant to be transmitted to all recipients of an item individual CINs for each recipient must be created and sent to the datapool.

7.1.5. Last Change Date Time Handling

There was some confusion in the past regarding the Last Change Date Time handling in the atrify datapool. Below is a description how it works:

1. Supplier sends Last Change Date Time in the incoming CIN
2. atrify datapool stores the supplier given Last Change Date Time but it will not be used in further sync process
3. atrify datapool saves the date/time when the item was imported to the atrify datapool. This date/time value will be used in the Last Change Date Time attribute, when item is synced to the retailers

7.1.6. Item Update Change Control

When receiving item updates atrify datapool checks whether the update has significant changes contained. If not the update will not be forwarded to the recipient. This was required to avoid sending item "updates" to the recipients where nothing changed in the item content. When receiving item updates atrify datapool removes any line feeds and spaces and compares the entire content between the existing and the input item.

Means removing and/or adding blanks or line feeds will not be considered a change. Workaround would be to add significant characters to the item. This will trigger the change again.

7.2. Responses

The supplier can receive the following message types from the atrify Data Pool

- GS1 Response: If the message is valid according to XML schema and the GDSN validation rule set
- GS1 Exception: If the message violates XML schema and/or GDSN validation rule set

Furthermore, the supplier might get feedback from the data recipient via the Catalogue Item Confirmation message type. CIC can have following status.

- RECEIVED
- SYNCHRONISED
- REVIEW
- REJECTED

In case of “REVIEW” the supplier should send an update to the synchronised item hierarchy. In case of an item “REJECT” the recipient is not interested in further updates, so atrify datapool will skip any item updates for that recipient. Please note this message type is not mandatory.

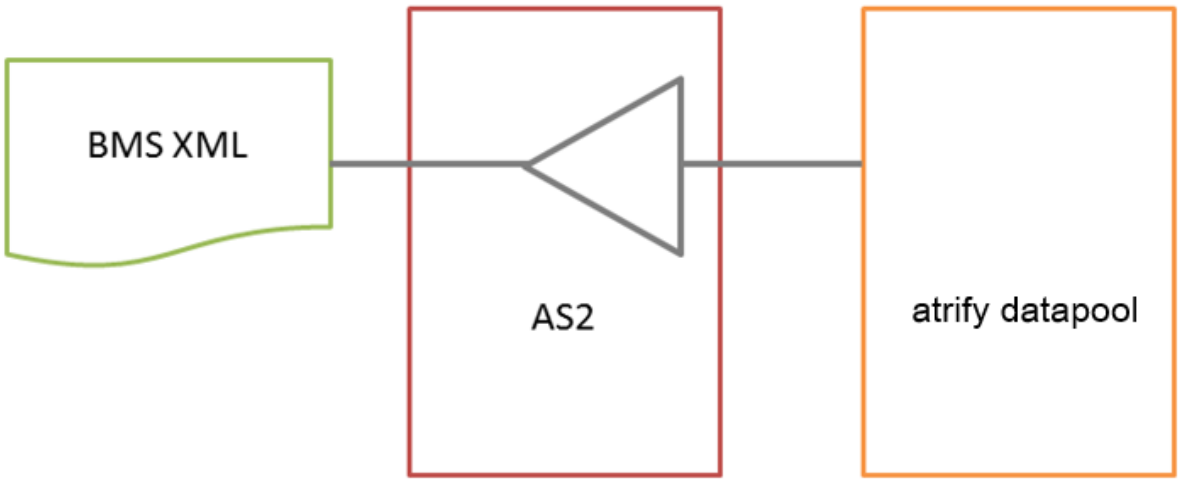


Figure 7: Receiving a message from the atrify datapool

7.3. Reset of Sync List by Data Source / Recipient via CIC/CIP

GDSN provides two possibilities to reset the Sync List entries of a Data Pool by the data recipient (DR):

1. Reset via RFCIN (Request for Catalogue Item Notification)

Containing the "Is Reload" flag an RFCIN-subscription provides the possibility to control, whether the data requested anew shall or shall not include items that were already rejected (by CIC-REJECT) before. The "Is Reload" flag has two options that have the following function:

isReload = true: Only items (matching the subscription criteria) shall be delivered that are already in synchronization with the recipient. I.e. that only those items that have the CIC-status empty, REVIEW, RECEIVED or SYNCHRONISED in the sync list are sent out to the recipient, but not items that have been rejected before by CIC-REJECT. The CIC-status in the sync list remains unchanged.

isReload = false: All items (also rejected ones) that match the subscription criteria shall be delivered anew. The CIC-status in the sync list will be reset to an (internal) default value. Then all items rejected by CIC-REJECT will be active again, and take part in the ongoing data synchronization with this data recipient.

2. Reset via CIC (Catalogue Item Confirmation)

Reset via CIC is not supported in the standard GDSN workflow. But atrify datapool supports for DR to reset the REJECT to any other CIC status by just sending a new CIC with an updated status (RECEIVED, SYNCHRONISED, REVIEW). This will activate the item flow again.

3. Reset via CIP (Catalogue Item Publication)

Further atrify datapool supports to reset the Sync List by data supplier (DS) as well although this is not part of the GDSN choreography. DS can set the isReload flag with the atrify datapool CIP extension as follows:

```
<extension>
        <wwre:isReloadExtension xmlns:wwre="urn:wwre:item:4">
                <isReload>true</isReload>
        </wwre:isReloadExtension>
</extension>
```

IF a CIP-ADD already exists in the pool another CIP-ADD for the same item / recipient will retrigger the item sync! This is called the “re-publish” feature also.

8. Item Hierarchies

When sending hierarchies to the pool, it is important to note that they must be complete. It is possible for suppliers to transmit individual hierarchy levels but they must be complete and every level that is higher than a base unit must contain the child link information. atrify only sends out complete hierarchies to the recipient.

8.1. Item Hierarchy Basic Rules

Each level of the product packaging hierarchy requires its own unique GTIN (in conjunction with the Information Provider GLN and the target market country code). For example, a consumer unit packaged in an inner pack, loaded in a case, and shipped on a pallet would need four GTINs —

- One GTIN representing the base consumer unit
- One GTIN representing the inner pack
- One GTIN representing the case
- One GTIN representing the pallet

About item hierarchies
You should not publish any items in a hierarchy until the hierarchy is complete. Please be aware that the process to modify an already synced hierarchy is complex. Any hierarchy related mistakes should be avoided before publishing and synchronising to the retailer. Otherwise the CIHW (catalogue item hierarchy withdrawal) process has to be executed.

To build a hierarchy, you must start at the lowest level — items must be linked in the order of lowest to highest. For example, let's say you have 3 items —

- Pallet item, which contains 3 Case items; the Retailer orders by the pallet.
- Case item, which contains 12 base units (Cans); the Retailer sells by the case to the consumer.
- Can item (base unit); the Retailer also sells by the can to the consumer.

In this case, you would first add the CAN item to the Catalogue.

Next, you would add the Case item, and

- For the tradeItemIdentificationOfNextLowerLevelTradeItem field, you would enter the GTIN of the CAN.
- For the quantityOfChildren, you would enter 12, because there are 12 cans per case.

Finally, you would add the Pallet item, and

- For the tradeItemIdentificationOfNextLowerLevelTradeItem, you would enter the GTIN of the Case.
- For the quantityOfChildren field, you would enter 3, because there are 3 cases per pallet.

The previous example applies to item loading via the web or any other machine-to-machine format you may be using. This order is important because, in order to send a case item and a link to its lower-level item, the lower-level must first exist as an item.

Rules
The following rules apply when creating a hierarchy.

- All items in a hierarchy MUST have the same GLN of Information Provider and Target Market Country Code.
- In order to publish the hierarchy, the hierarchy must contain at least one orderable unit AND one invoice unit.
- For a BASE_UNIT_OR_EACH item, there can be no lower-level, or child, item.

Modifying item hierarchies
atrify datapool supports hierarchy changes. Means you can swap the GTINs within a product hierarchy. If you want to change the following information

- tradeItemUnitDescriptor
- isTradeItemABaseUnit
- isTradeItemAnOrderableUnit

There are two ways to modify the structure of a hierarchy, indicated by the attributes above, is to
1. Unpublish and republish again
    - Revoke publication to all Retailers
    - Fix the hierarchy
    - Re-publish the hierarchy

2. Put publication „ON_HOLD“ via CIHW Link Correction Message
Since the Major Release a new message type Catalogue Item Hierarchy Withdrawal (CIHW) was introduced either to delete publications (Publication Withdrawal) or to edit hierarchy configurations. The standard choice to change the hierarchy is to
    - send a CIHW with hierarchyDeletionReasonCode= HIERARCHY_LINK_CORRECTION. This will set the publication state of the published hierarchy to “ON_HOLD” in the atrify datapool. The CIHW will be forwarded to the impacted recipient(s) to delete the current hierarchy and replace by the next delivered hierarchy for that published level. Please note: The hierarchy cannot be published to further recipients, if the publication state is “ON_HOLD”.
    - Fix the hierarchy
    - Resend the fixed hierarchy per CIN-ADD. The publication status becomes active again.
    - Note: Sending an update for the hierarchy will chamge the publication state from “ON_HOLD” to active again.

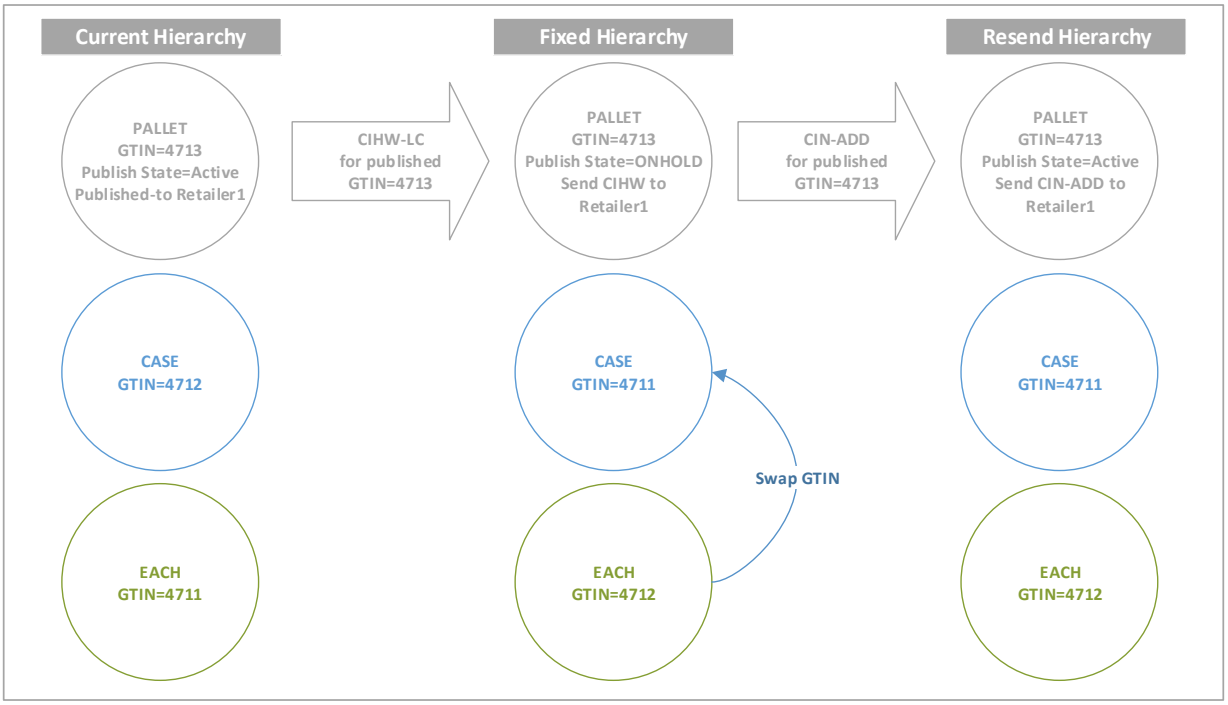


Figure 8: Hierarchy Link Correction process via CIHW (Basic Workflow)

8.2. Nested

The term nested hierarchy is used for item data that is sent with the information contained in a hierarchy. Illustrated with example below (not XML well-formed):

```
<eanucc:transaction>
[...]
  <gdsn:catalogueItemNotification>
    <catalogueItem>
      <tradeItemUnitDescriptor>CASE</tradeItemUnitDescriptor>
      [...]
      <catalogueItemChildItemLink>
          <catalogueItem>
            <tradeItemUnitDescriptor>BASE_UNIT_OR_EACH</tradeItemUnitDescriptor>
            [...]
            <catalogueItemChildItemLink>
          </catalogueItem>
        </catalogueItemChildItemLink>
      </catalogueItem>
    </catalogueItemChildItemLink>
  </gdsn:catalogueItemNotification>
</eanucc:transaction>
```

Figure 9: XML example for nested message

8.3. Unnested

The term unnested hierarchy is used when users send GDSN CIN Messages in which the items are not nested in a hierachy structure. In this case the system internally links these items to the corresponding structure. Illustrated with example below (not XML well-formed):

```
<eanucc:transaction>
[...]
  <gdsn:catalogueItemNotification>
    <catalogueItem>
    <tradeItemUnitDescriptor>CASE</tradeItemUnitDescriptor>
    [...]
    </catalogueItem>
  </gdsn:catalogueItemNotification>
  <gdsn:catalogueItemNotification>
    <catalogueItem>
    <tradeItemUnitDescriptor>BASE_UNIT_OR_EACH</tradeItemUnitDescriptor>
    [...]
    </catalogueItem>
  </gdsn:catalogueItemNotification>
</eanucc:transaction>
```

Figure 10: XML example for unnested message

Transmission of unnested hierarchies:

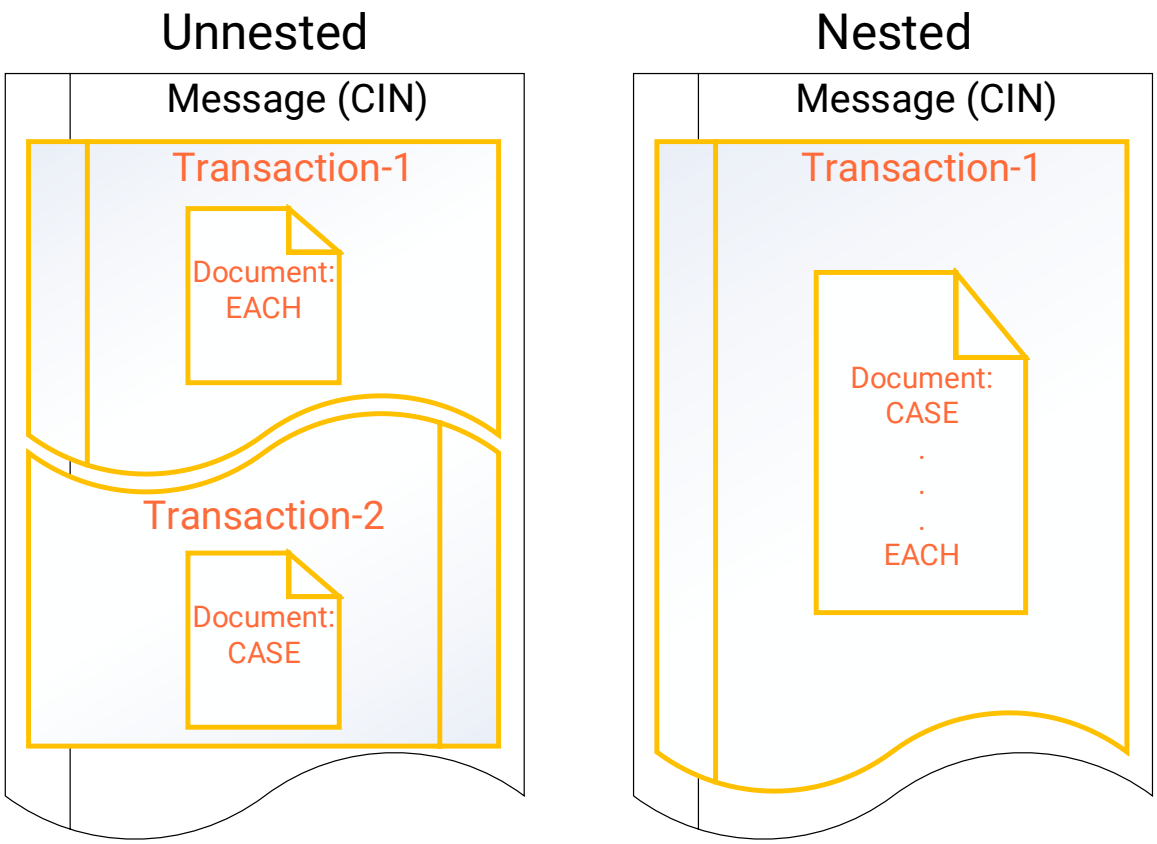


Figure 11: Illustration of un-nested hierarchies vs. nested hierarchies

Hierarchies can be transmitted in different messages. Please note that it is **mandatory** to transmit the message containing the base item **first**, followed by the case which must contain the link to the base in question. **Transmitting the message with the case first will result in an error**.

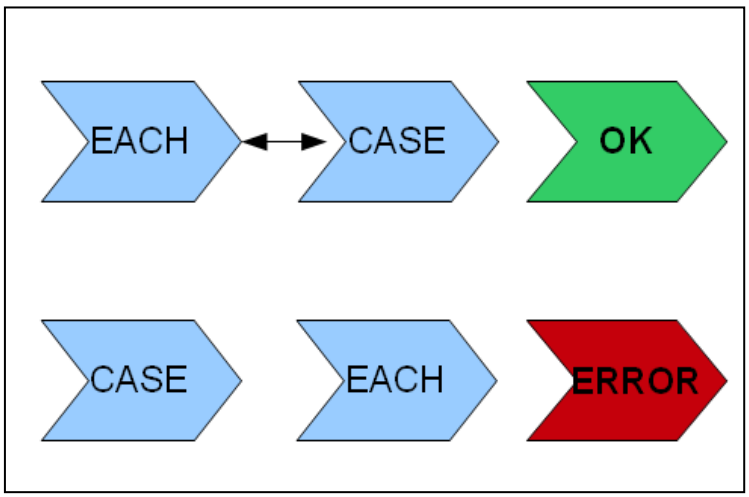


Figure 12: Transmission of hierarchies in different messages (for ADD)

9. Validations

To ensure high data quality atrify datapool performs two different kinds of validation – Schema Validations and Content Validations. It is possible that two validations may check the same content.

Note: For users of the Publishing UI both types content and schema validations are performed directly in the software.

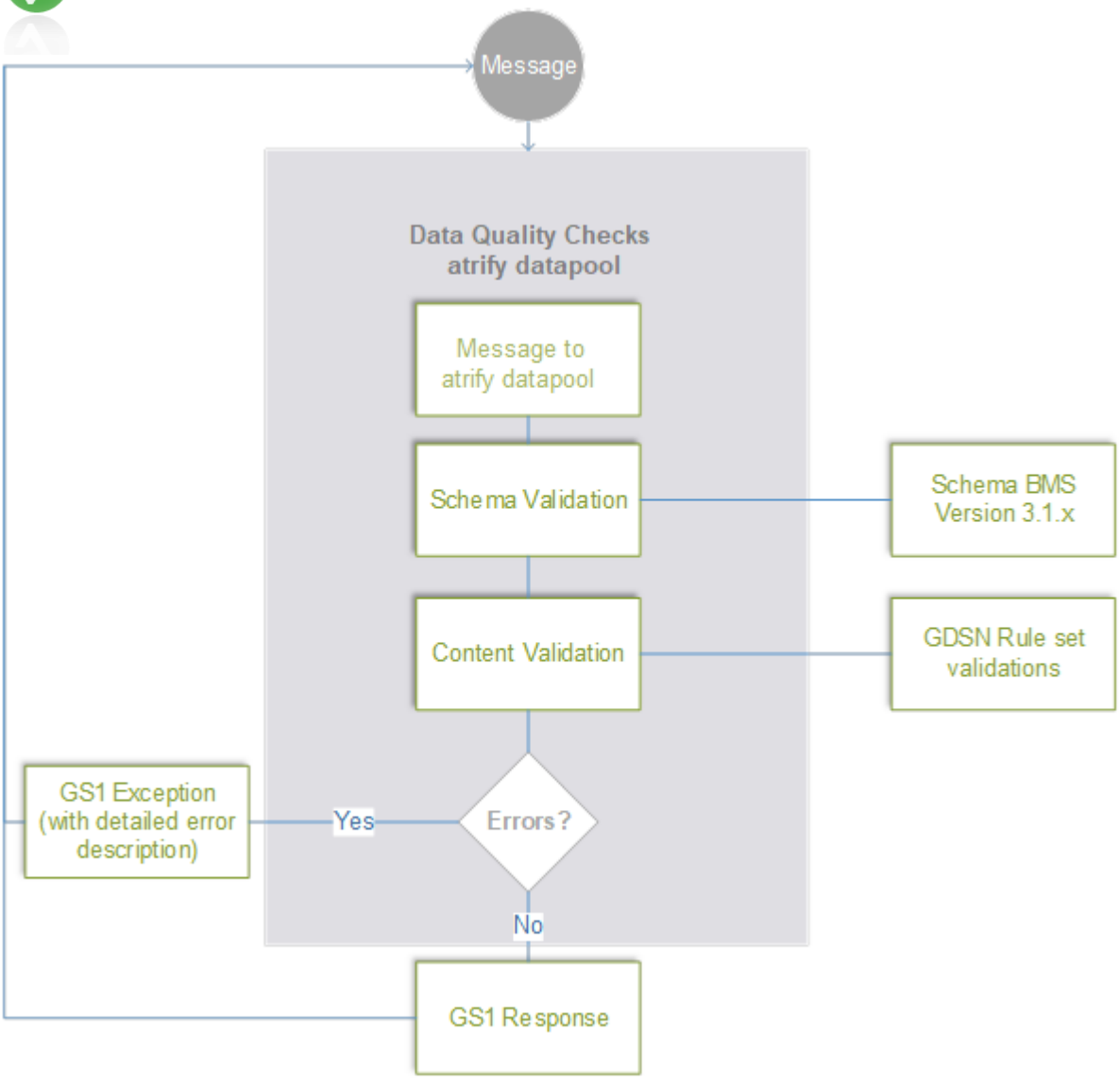


Figure 13: To ensure data quality: validation process

9.1. Schema Validations

With Schema Validations incoming data is checked to be compliant to a currently valid GDSN scheme (see http://www.gdsregistry.org/3.1/schemas/). This is the first set of validations performed upon receipt of data from a Supplier.

If the system finds XML schema errors, then the message will be rejected by the system and an error GS1 Exception message sent to the customer. No further processes are performed. If the message is XML schema valid, the system will continue with Content Validations.

Incorrect:

<isTradeItemABaseUnit>yes</isTradeItemABaseUnit>

<isTradeItemAConsumerUnit>no</isTradeItemAConsumerUnit>

<isTradeItemADespatchUnit>1</isTradeItemADespatchUnit>

<isTradeItemAVariableUnit>0</isTradeItemAVariableUnit>

Correct:

<isTradeItemABaseUnit>false</isTradeItemABaseUnit>

<isTradeItemAConsumerUnit>false</isTradeItemAConsumerUnit>

<isTradeItemADespatchUnit>true</isTradeItemADespatchUnit>

<isTradeItemAVariableUnit>false</isTradeItemAVariableUnit>

9.2. Content Validations

The Content Validations check whether the message content corresponds to the rules determined by the GDSN (example primary key, interdependencies between data elements or whether particular entries correspond to the format demanded).

If the message content contains errors, the message is rejected by the system and the customer receives a GS1 Exception message containing the detailed error description.

The following table lists some examples of the content related validations:

Text of rule	Related attribute(s)	Note
If tradeItemUnitDescriptor is equal to 'BASE_UNIT_OR_EACH' then ChildTradeItem/gtin must be empty.	Trade Item Unit Descriptor Child Trade Item – Trade Item Identification: GTIN	
If gross weight is not empty and net weight is not empty then gross weight must be greater than or equal to net weight.	Net Weight Gross Weight Gross Weight UMO	
The effective date time must not be empty.	Effective Date	
A value for Classification Category Code was given, but the indication is missing.	If AdditionalClassificationCategoryCode Then -> EANUCC ClassificationAttributeTypeCode AND -> EANUCC ClassificationAttributeValueCode (Core Item)	
The value “x” in attribute “y” is outside of the permissible range (1 - 999) or the number of digits after the decimal point is incorrect.	allowed range 1 - 999 for attributes: QuantityOfInnerPack CouponFamilyCode quantityOfNextLevelTradeItemWithinInnerPack PegHoleNumber allowed range: 1 - 999	
The value “x” in attribute “y” is already being used as primary key value for another product.		DuplicatePrimaryKey

Table 7: Examples for content validations

The full list of all GDSN validations can be found here: http://www.gs1.org/gdsn/gdsn-validation-rules/3-1

A customer wise configuration of the standard GDSN validations (switch on or off validations) per GLN is not possible in the atrify datapool as they are GDSN basics. There are some on demand validation rules that can be switched on/off in the atrify datapool party UI (e.g. Price Duplication check, FDA rules).

10. Price Synchronisation

This section provides some background to the GDSN **Price** and **Price Relationship** synchronization.

10.1. GDSN Background

The Price Synchronisation message set includes the following messages:

- Price Synchronisation Document (PSD)
- Price Synchronisation Confirmation (PSC)
- GDSN Response Messages
    - GS1 Response (positive use case)
    - Gs1 Exception (negative use case)

The price message is made up of a header segment and up to four distinct segments (relationship, condition, item depiction, price type) with specific purposes. Each segment has an action code and segment ID to ensure referential integrity. A single price message can be used for many different purposes including the following:

- Establishing a Trading Partner Relationship
- Communicating elements of prices that are included on an invoice in an effort to equal the actual payment and the expected payment

If a change is required for an attribute, all mandatory and any desired optional attributes originally included in that segment must be included in the message. A segment is modified by a full refresh.

The **Price Synchronisation Document (PSD)** message is a complex message that carries the price information in multiple segments to address the needs of a business relationship. The table below provides a conceptual layout of the message and the supported segment types.

Segment	**Description**
Price Synchronisation Header Segment	• Identifies both Information Provider & Party Receiving Private Data • Contains a Price Document ID and references a Relationship ID • Mandatory, header travels with each Price Document
Relationship Segment	• Establishes Price Sync Relationship parameters (similar to TPD) • Contains a Relationship ID and an Action Code • Qualifies the Business Location of Party Receiving Private Data
Condition Segment	• Used to sync Summary Conditions and conditions that are not item specific • *(out of scope, not used in GS1 Australia)*
Item Depiction Qualifier	• Header for the Price Type Segment • Used to qualify the Trade Item • Multiple Item Depiction Qualifiers are allowed per document
Price Type Segment	• Used to associate price components with the Trade Item • Contains a Price Type ID and an Action Code • At least one Price Type is required per Trade Item depicted • May contain a Bracket Qualifier subclass • Multiple Price Types are allowed per Trade Item

A segment Action Code must be present for each segment. The supported Action Codes are:

Action Code	Description
ADD	to add a new segment or a new item price type record within an existing Item Depiction segment
CHANGE_BY_REFRESH	to change the values for selected mandatory and all optional attributes
CORRECT	may only be used to change the value of some mandatory key attributes or an attribute where the change results in a material financial impact. The other mandatory attributes that need to be revised require a “DELETE” followed by a new “ADD”
DELETE	to remove an existing segment or record within a segment
NO_ACTION	to indicate that a previously existing segment or record is being sent, but none of the values have changed **NOTE**: The GS1 Australia community currently does not make use of action code “NO_ACTION”.

The **Price Synchronisation Confirmation** message is used to indicate the acceptance or rejection status for each segment. The underlying requirement is that a confirmation status needs to be provided by the data recipient for each segment. A “No Response” is considered a negative status or non-acceptance, in all cases except when adding a relationship segment. The confirmation message refers to the Price Synchronisation Document and the segment when providing a confirmation status.

The Price Synchronisation Document can be packaged in a single GDSN message either as 'multiple documents in a single transaction' or 'multiple transactions with each containing a single document' or single transaction containing a single document'. The rule requiring that the SDP provide a sequential identifier (Price Synchronisation Document Identification) for each PSD will be enforced irrespective of how the document is packaged in a transaction inside a GDSN message.

The transaction as identified in the GDSN message will dictate the following rules around processing of the message, as has been GDSN practice:

- if a single Price Synchronisation Document fails during processing, all Price Synchronisation Documents in that transaction will fail;
- a transaction will be processed successfully only when all contained Price Synchronisation Documents have been processed successfully.

The GDSN use cases itself are specified in the “Business Message Standard (BMS) – Price Synchronisation” document and are listed here in summary.

UC-1 Add Trading Relationship – SDP creates a relationship by sending a Price Synchronisation message with a document command of “ADD” with a relationship segment action code of “ADD” to the RDP.

UC-2 Update Trading Relationship – SDP updates the relationship by sending a Price Synchronisation message with a document command of “CHANGE_BY_REFRESH” with a relationship segment action code of “CHANGE_BY_REFRESH” (for a modification) or “CORRECT” (for a correction) to the RDP.

UC-3 Cancel Trading Relationship – SDP terminates the relationship by sending a price synchronisation message with a document command of “CHANGE_BY_REFRESH” with a relationship segment action code of “DELETE” to the RDP.

UC-4 Discontinue Trading Relationship – SDP terminates the relationship by sending a price synchronisation message with a document command of “CHANGE_BY_REFRESH” with a relationship segment action code of “CHANGE_BY_REFRESH” and an appropriate relationship effective end date to the RDP.

UC-9 Add Item Price Type – SDP creates a price synchronisation message with a document command of “CHANGE_BY_REFRESH” (if the trading relationship has already been established) and the price type segments with a segment action code of “ADD” to the RDP and updates the price synchronisation list.

UC-10 Modify Item Price Type – SDP creates a price synchronisation message with a document command of “CHANGE_BY_REFRESH” and the price type segments with a segment action code of “CHANGE_BY_REFRESH” (for a modification) or “CORRECT” (for a correction) to the RDP and updates the price synchronisation list.

UC-11 Withdraw Item Price Type – SDP creates a price synchronisation message with a document command of “CHANGE_BY_REFRESH” and the price type segments with a segment action code of “DELETE” and updates the price synchronisation list.

UC-12 Discontinue Item Price Type – SDP creates a price synchronisation message with a document command of “CHANGE_BY_REFRESH” and the price type segments with a segment action code of “CHANGE_BY_REFRESH” and a populated price type effective end date to the RDP and updates the price synchronisation list.

Actor	Event	Action	Comment
Data Source	ADD a relationship segment	PSD ADD with relationship segment and Action Code ADD is synchronised (sent to RDP).	
Data Source	ADD an item depiction segment (contains at least one item price type segment)	PSD CHANGE_BY_REFRESH with item depiction segment and the associated item price type segment with Action Code ADD is synchronised.	
Data Source	ADD a relationship segment and one or more item depiction segments	PSD ADD, with all segments and associated Action Codes ADD is synchronised.	
Data Source	Update a relationship segment with action CHANGE_BY_REFRESH	PSD CHANGE_BY_REFRESH with relationship segment and Action Code CHANGE_BY_REFRESH is synchronised.	Positive confirmation status on the segment is required except when the previous action was ADD.
Data Source	Update an item depiction segment and/or item price type segment with action CHANGE_BY_REFRESH	PSD CHANGE_BY_REFRESH with item depiction and the associated item price type segment and Action Code CHANGE_BY_REFRESH is synchronised.	The relationship and item price type segment must have a positive confirmation status. The item depiction itself does not have an associated Action Code or a sync list.
Data Source	Update any combination of relationship and item depiction segments with individual actions CHANGE_BY_REFRESH	PSD CHANGE_BY_REFRESH with the segments and Action Code CHANGE_BY_REFRESH is synchronised.	The relationship and each of the segments must have a positive confirmation status.
Data Source	Correct a relationship segment with action CORRECT	PSD CHANGE_BY_REFRESH with the relationship segment and Action Code CORRECT is synchronised.	The relationship segment must have a positive confirmation status except when the previous action was ADD.
Data Source	Correct an item price type segment with action CORRECT	PSD CHANGE_BY_REFRESH with item depiction and the associated item price type segment and Action Code CORRECT is synchronised.	The relationship and item price type segment must have a positive confirmation status.
Data Source	Correct any combination of relationship and item depiction segments with individual actions CORRECT	PSD CHANGE_BY_REFRESH with the segment and Action Code CORRECT is synchronised.	The segments must each have a positive confirmation status.
Data Source	Delete a relationship segment with action DELETE	PSD CHANGE_BY_REFRESH with the relationship segment and Action Code DELETE is synchronised.	The relationship must not have any existing segments associated that have not been deleted.

Data Source	Delete an item depiction segment and an item price type segment with action DELETE	PSD CHANGE_BY_REFRESH with item depiction and the associated item price type segment and Action Code DELETE is synchronised.	The relationship must have a positive confirmation status, and the dependent segments must all be deleted.
Data Source	Delete a combination of relationship, item depiction and item price type segments with individual action DELETE	PSD CHANGE_BY_REFRESH with the relationship, item depiction and item price type segment with individual Action Code DELETE is synchronised.	Individual restrictions will apply as specified for each of the segments.
Data Source	A group of segments with distinct Action Codes is provided	The corresponding PSD CHANGE_BY_REFRESH with all processed segments and their Action Codes will be synchronised.	Individual restrictions will apply as specified for each of the segments.
Data Recipient	A positive confirmation message is received for a segment	Any pending updates for this segment as well as any depending segments will be synchronised as a PSD CHANGE_BY_REFRESH.	Individual restrictions will apply as specified for each of the segments.

Table lists the events triggering the Price messaging use cases.

10.2. Processing PSD Messages

The atrify datapool supports two modes of operation for processing incoming PSD messages (similar to CIN processing)

- “pass-through” mode for data sent from other Data Pools
- “local” mode for data sent from suppliers for which this atrify datapool instance acts as Home Data Pool

10.2.1. Processing PSD Messages in Common

atrify datapool processes an incoming PSD message as follows:

- All messages received from a Data Source are processed in the order they were received.
- An inbound PSD is validated against the schema and associated code lists.
- A PSD with no segments or invalid segments will be rejected.
- The system validates that the PSD is associated with a Document Command Header type of either “ADD” or “CHANGE_BY_REFRESH”. No other commands (“CORRECT”, “DELETE”) are supported for this message and will cause the transaction to be rejected.
- The Content Owner at each of the segment level should be consistent with the Information Provider defined for the Relationship.
- The PSD message must always contain the Header Segment to identify a relationship between the source and the recipient.
- The Price Synchronisation Document ID is mandatory and must be unique and sequentially incremented within a particular Price Relationship. The SDP maintains for each Data Source and Data Recipient relationship a sequence for the Price Synchronisation Document ID. The messages from the data source will not be subjected to this same validation and will be allowed to use their own specific scheme for the ID. The outbound message will contain an updated ID that reflects the sequential ID rule.
- In the absence of any reference to an item in the segments, the SDP creates the synchronisation list entry and sends the information to the Data Recipient or RDP.
- If a segment is synchronised with a positive status, an update to the segment will cause the sync list status to be reset to a negative value of “No Response” pending the receipt of a positive status from the recipient.
- The atrify datapool will reject the failing transactions and continue processing only the successful transaction.
- A GS1 Response will be generated for each PSD processed, indicating the success or failure appropriately.

10.2.2. Special Requirements for Processing Price Type Segments

The Item Price Type segment is used to synchronise Price Types for the item/s depicted in the Item Depiction segment. Data Recipients cannot accept or reject individual Brackets Qualifiers. If the requirement is to discontinue a Price Type, the End Effective Date must be populated or updated. If the action code is “DELETE” there should be no dependent Price Types, and the price should not yet be in effect.

- The Price Type segment is mandatory and repeatable within each Item Depiction Qualifier.
- If the Item Depiction segment is available, the SDP checks against the item synchronisation list that the confirmation status is not “REJECTED”. If the item synchronisation is rejected, price synchronisation segment is also rejected for the item. The system shall monitor the item sync list every time a price is synchronised for the item to insure that the item has not been rejected.
- A Price Type can be synchronised only if:
    - The segment itself has a positive status if previously synchronised;
    - The Relationship has a non-REJECT Status
    - The associated Item Depiction segment has a non-REJECT status (that is none of the Price Types associated with the Item Depiction segment has a REJECT status);
    - The Target Price Type if it exists has a positive status;
- Multiple Price Types may exist simultaneously for a Catalogue Item and each Price Type will have it’s own confirmation status.
- The Bracket Qualifier group of attributes is not a segment and hence does not have an associated confirmation status. It is repeatable within each price type segment. A bracket can only exist once in the Relationship; however, this will not be checked for uniqueness by the atrify datapool.

10.2.3. Merge Item Depiction Qualifier

Unfortunately, there is no clear GDSN guideline how to structure the relation of item depiction qualifier and the price types. Based on the GDSN rule to process the price confirmations per itemPriceType and internal processes, atrify supports as standard functionality the most flexible way: The 1 : 1 approach. Means if a supplier wants to send two price records for the same item (e.g. catalogue price and discounts), they are repeating the itemDepictionQualifier. But in exceptional cases some suppliers use to send multiple itemPriceTypes under a single itemDepictionQualifier. But regardless of how the supplier structures the records the data pool stores as itemDepictionQualifier/itemPriceType combination.

In case any recipient or recipient data pool expects one <itemDepictionQualifier> instance including all related <itemPriceType> instances a party attribute can be configured to “merge” all related <itemPriceTypes>below one <itemDepictionQualifier>.

atrify access management will set the party attribute “mergeItemDepictionQualifier” for a dedicated retailer or external data pool GLN and set value = “Y”. If an atrify recipient has set this party attribute the price exporter will merge the itemDepictionQualifier as described above to that recipient only. In case of external data pool recipients this setting will apply to any recipient of those external data pool.

10.2.4. Special Requirements for Processing Relationship Segments

- IMPORTANT: Price Relationship segments are synchronised independently of items. There is no dependency on a subscription.

- If the Document Header Command is “ADD”, this is the first message for a Relationship and hence the Price Synchronisation Document ID assigned by the SDP must = “1”.
- If the Document Header Command is “ADD”, the PSD must include a relationship segment at the minimum.
- An Action Code of “DELETE” on a Relationship is only allowed when the Effective Start Date is in the future, will trigger the SDP to delete the relationship. This will require that all associated segments be deleted prior to processing the Relationship segment delete.
- The Relationship Identification is mandatory, and must be unique within the data source, and must equal the Relationship Identification in the header. The Data Source cannot create two Relationships with the same ID.
- The “No Response” status for the sync list is considered to be a negative status along with the “REJECTED” status and will cause future synchronisation to stop. The only defined exception to this rule is when synchronising an update on a relationship segment following a “No Response” to a Relationship segment with the Action Code of “ADD”. The definition for a positive status on the confirmation of a Relationship segment, when the action was “ADD” includes the status of “No Response”. The synchronisation of other segments associated with a segment of an Action Code of “ADD” and a status of “No Response” will be stopped.
- If the PSD contains at least a Relationship segment with an Action Code of “ADD”, there are no dependency checks executed across the segments. The sync list for each of the segments will not be checked for a positive status in this case.
- If the PSD contains different segment types and the Relationship segment Action Code does NOT equal “ADD”, a processing must follow the sequence below starting at the top of the list, and following the order of the next entry in the list:
    - Relationship Segment
    - Dependent Price Type Segments
    - Segment Depends on Price Type Segments
- The validation will confirm a positive sync list status and verify the sync list of any dependency between segments.
- If ending a Relationship segment, all Price Type segments for the relationship segment must be deleted/end dated before a delete/end date can be sent for the Relationship segment.

10.3. Processing PSC Messages

The Price Synchronisation Confirmation PSC message is used to communicate the status at the recipient end to the source of the price information one segment at a time. Unlike item confirmation (via CIC), price confirmation is mandatory in GDSN, i.e. a supplier cannot update a price or relationship unless it has been responded by the recipient via PSC. In this context “No Response” is effectively a negative confirmation and stops the synchronisation process.

The set of synchronisation confirmation status codes is defined in the XML schema and is the same as used for CIC:

Confirmation Status Codes	Description
RECEIVED	Data has been received by the Recipient, but no business decision has been made on the data.
REVIEW	A request to the Data Source to “review” their data because the Data Recipient has received discrepant data which they cannot synchronise.
SYNCHRONISED	Data is integrated and in sync. Usually it implies that the data has been implemented into the Data Recipient’s back-end system.
REJECTED	The Recipient requests that no further updates are desired. Data will no longer be synchronised or updates will no longer be provided.

atrify datapool is processing an incoming PSC as follows:

- All messages received from a Data Recipient will be processed in the order they were received.
- An inbound PSC will be validated against the schema and associated code lists.
- The system will validate that the PSC is associated with a Document Command Header type of “ADD”. No other commands (“CHANGE_BY_REFRESH”, “CORRECT”, “DELETE”) are supported for this message and will cause the transaction to be rejected.
- A PSC message must refer to a segment that was received for the Data Recipient. Else the message will be rejected.
- The Content Owner at the transaction and command level should be the data recipient, however the content owner at each of the segment level should be consistent with the data source (information provider) defined for the relationship.
- The atrify datapool will reject the failing transactions and continue processing only the successful transaction.
- A GDSN response will be generated for each PSC processed, indicating the success or failure appropriately.
- A Data Recipient can send multiple confirmations for a single price message or message segment.

10.4. Price Validations

All inbound messages (PSD and PSC) are validated by the atrify datapool. For external Trading Partners from other Data Pools (pass-through) XML schema validation is performed only.

10.4.1. Basic Validation

Basic validation is already an integral part of the existing atrify datapool, and includes the following checks:

- XML schema validation
- Valid use of code list attributes
- GTIN/GLN check digit validation
- Document type consistency (cannot mix document types in a single file)
- BMS version number consistency

10.4.2. GDSN Rules Validation

Validating the GDSN rules (as specified in “BMS Validation Rules for Global Data Synchronisation Network” document) includes:

- Business validations on the relevant attributes as indicated for each segment.
- Validate the changes to the segment are in agreement with the Action Code specified.

For more details on GDSN price rules check documents in the following link:
http://www.gs1.org/gdsn/gdsn-validation-rules/3-1

Note: In the pass-through mode GDSN validation rules are not performed by the atrify datapool as this is the Source Data Pools responsibility to send valid GDSN data only.

10.5. PSD Message Structure

The PSD message is no different to other GDSN messages (e.g. CIN) in its internal structure (see XML Schema for details):

```
Header

Message

+-- Transaction [1..*]

    +-- Command [1..*]

        +-- DocumentCommand

            +-- DocumentCommandHeader

            +-- DocumentCommandOperand

                +-- PriceSynchronisationDocument
```

Multiple transactions per message are allowed and governed by the following rules:

- For incoming messages, each relation and price segment is processed in the order they appear in the message.
- For synchronizing with the Retailers, one transaction is created for each change of Item Depiction + Recipient + Relationship combination.
- All prices having the same combination of Item Depiction + Recipient + Relationship combination will be in the same transaction.

10.6. Price Duplication

10.6.1. Overview

Price Duplication occurs if prices which for the same Data Recipient, under very specific conditions, overlap geographically and temporally. 'Very specific conditions' indicate what determines two or more equivalent prices that must then be checked for geographical and temporal overlap.

Validations around Price Duplication will be based on both geographical and temporal (start / end date) overlap. These validations will only be checked against pricing for recipients who have indicated that they require the prevention of duplicate pricing.

Note that for clarity in this document, a Price Type being added or modified (changed or corrected) either online or via M2M is known as the '**input price**'. Any price currently in the suppliers' catalogue is referred to as the '**existing price**' (there may be 0...many of these).

Also, it is a requirement for duplicate checking of the input prices to occur across the input prices within an upload file, in addition to duplicate checking the input prices against the prices already existing in the atrify datapool database. A Supplier must be able to end-date a price and also add a new price in the same upload.

For the rest of this document, where the expression 'equivalent existing price' is used, it should be interpreted as 'equivalent existing price or equivalent price within the same load file' (see section 10.6.1 for definition of price equivalence).

Price Relationships do not undergo price duplication checking.

10.6.2. Validations to Prevent Price Duplication

The following rules apply for Price Type segment maintenance when processing prices for Recipients that require Price Duplication checking.

Price Duplication can only occur between equivalent prices. There are three steps to checking for duplicate prices, each of which is explained in detail in subsequent sections.

NOTE: In regards to the Price Types checked, only parent Price Types, i.e. Price Types that have an Application Sequence Number of '1', are to be considered for Price Duplication Checking. This excludes Price Types of 'ALLOWANCE' or 'CHARGE'.

For each Price Type added or modified (change or correct) via M2M or the web user interface, the following steps must occur to check for duplicate pricing:

1. Does the input price contain any conditions under which price duplication checking does not occur?
    a. If yes, continue with normal price processing.
    b. If no, go to step 2.
2. Do equivalent prices exist?
    a. If yes, go to step 3.

b. If no, continue with normal price processing.

At this point, the process must loop through all equivalent existing prices until price duplication checking has been performed for each equivalent existing price found.

3. For each equivalent existing price found, does the existing price contain any conditions under which Price Duplication checking does not occur?
    a. If yes,
        - Fetch the next existing price.
        - Return to step 3.
    b. If no, go to step 4.
4. Is there **geographical overlap** between the input price and the current equivalent existing price selected?
    a. If yes, then continue to step 5 to check for temporal overlap. It is fine for a price to have geographical overlap and not temporal overlap. It is not fine for a price to have both.
    b. If no,
        - Fetch the next existing price.
        - Return to step 3.
5. Is there **temporal overlap** between the input price and the current equivalent existing price selected?
    a. If yes, then continue to step 6 to check for bracket overlap.
    b. If no,
        - Fetch the next existing price.
        - Return to step 3.
6. Is there a **bracket overlap** between the input price and the current equivalent existing price selected?
    a. If yes, fail the price type being added or modified.
    b. If no, and this is the last equivalent existing price in the loop, continue with normal price processing.
    c. If no, and this is not the last equivalent existing price in the loop
        - Fetch the next existing price.
        - Return to step 3.

Figure 14: Price Duplication Checking

10.6.3. Conditions under which Price Duplication Checking does not occur

Whether or not a Supplier price for a Recipient requiring duplicate price checking is the input price or an existing equivalent price, if either price meets any of the following criteria, the price CANNOT be considered for Duplicate Price checking.

If the **input** price meets at least one of the following conditions, Duplicate Price checking need not be performed. In addition to the below conditions, if the **input** price has non-null Price End Date value (regardless of Start Date value) between NOW and “NOW+48 hours” (where NOW is the instantaneous processing time for this input price), then the price is excluded from Duplicate Price checking.

If an **equivalent existing** price meets at least one of the following conditions, it will be skipped for Duplicate Price checking with the input price. (See next section for equivalence checking between the input and existing price).

The price is **excluded** from Duplicate Price checking if the price:

1. is deleted (via Action Code “DELETE”)
2. has Price Application Sequence > 1
3. has more than 1 type of ‘location’ information populated at the same time from any of ‘Alternate Location Grouping’ + ‘Ship To’ + ‘TM-SD’
4. has more than one Effective Start Date/Time populated
5. has more than one Effective End Date/Time populated
6. Effective Start Date/Time Context is **not** ‘FIRST_ORDER_DATE’
7. Effective End Date/Time Context is **not** ‘LAST_ORDER_DATE’
8. has equal Effective Start Date/Time and Effective End Date/Time (zero duration)
9. has a past Effective End Date/Time (historic price).
   This can only occur on an existing price being considered for Duplicate Price checking as a new price cannot be added with a past End Date or modified to set a past end date.
10. has more than one bracket
11. Bracket Qualifier (if present) is not ‘MEASUREMENT_RANGE’

NOTE: The order of exclusion checking in the list above does NOT imply the order in which these checks should be executed. Each condition alone causes a price to be excluded from Duplicate Price checking. Thus, the first condition found to be true excludes the input or existing price being checked from Duplicate Price checking and no other conditions need to be checked for that price.

10.6.4. Equivalent Price Checking

A price is **'equivalent'** to another price if and only if they share the same 'null' or 'not null' values for the following attributes:

Attribute	GDSN Attribute Name	Comment
GTIN	<itemDepictionQualifier> <catalogueItemReference> <**gtin**>	The GTIN, GLN, TM, TM-SD must match exactly for a price to be considered equivalent with another price.
Data Source GLN	<itemDepictionQualifier> <catalogueItemReference> <**dataSource**>	
Target Market	<targetMarket> <**targetMarketCountryCode**>	
Target Market Subdivision	<targetMarket> <**targetMarketSubdivisionCode**>	
Recipient GLN	<partyReceivingPrivateData>	
Price Sequence	<priceTypeApplicationSequence>	Must be '1'. The price will already have been excluded if application sequence is not '1'
Price Type	<priceTypeCode>	
Price Basis Quantity	<priceValueInformation> <priceBasisQuantity> <**value**>	Null decimal considered identical to absence of decimal (i.e. 1 = 1.000)
Price Basis Quantity UOM	<priceValueInformation> <priceBasisQuantity> <**unitOfMeasure**>	
Price Value Type	<priceValueInformation> <**priceValueType**>	

The following attributes are configurable (by Recipient) to be included in the equivalence check.

Optional Attributes

Attribute	GDSN Attribute Name	Comment
Price Type Description	<priceTypeDescription>	Case insensitive match
Distribution Method Code	<distributionMethodCode>	
Price Action Reason	<priceActionReason>	
Reference Doc ID + Desc	<referenceDocumentInformation> <**referenceDocumentIdentifier**> <referenceDocumentInformation> <referenceDocumentDescription> <**text**>	Case insensitive match of both attributes
Relationship ID	<priceSynchronisationRelationshipIdentification>	

If the configuration is changed, newly added or modified prices are checked against the changed configuration from that point of time where the configuration change is saved in the system. No retrospective check across existing price data when (and if) the Recipient's equivalence requirements change.

It is to be specifically stated that price 'equivalence' takes into **no account**:

- Action Code for Price ("ADD", "CORRECT", "CHANGE_BY_REFRESH")
- Price Value
- any other price attribute now (or added later), unless otherwise stated

NOTE: A price is never 'equivalent' to itself, which is important for the purpose of price Change / Correct actions.

10.6.5. Checking Prices for Geographical Overlap

If an input price is equivalent to one or more existing prices, then the first Duplicate Price check performed is to determine whether or not there is geographical overlap. This check is being performed between the input price and every existing equivalent price as well as every other input price in the same load file for the same item.

A price has **'Geographical overlap**' with an **'equivalent'** price, if they have 1 (or more) shared 'Target Market Sub-divisions' or 'Ship To GLNs' or 'Alternate Location Groupings' – that is to say the applicable geographical regions for the price overlap wholly or partially. There is one exception to this rule which is noted below where TM-SD and ShipTo for a price are both <null>.

Attribute	GDSN Attribute Name	Repeats
Target Market Subdivision	`<priceTargetMarketSubdivision>`	Yes
Alternate Location Grouping	`<alternateLocationGrouping>`	No
Ship To	`<shipTo>`	Yes

An input price can be constructed in 8 scenarios for TM-SD, Alternate Location Grouping, and Ship To (as below) - the same scenarios apply for an existing price.

Scenarios

Scenario	TM-SD	Alternate Location Grouping	Ship To	National Pricing
1	<null>	<null>	<null>	Yes
2	populated (1..*)	<null>	<null>	No
3	<null>	<null>	populated (1..*)	No
4	<null>	populated	<null>	No
5	populated (1..*)	populated	<null>	Excluded (*)
6	populated (1..*)	<null>	populated (1..*)	Excluded (*)
7	<null>	populated	populated (1..*)	Excluded (*)
8	populated (1..*)	populated	populated (1..*)	Excluded (*)

(*) Scenarios 5 - 8 are explicitly excluded from Price Duplication check for either ‘input’ or ‘existing’ prices as this is an invalid business case in most pricing communities as only 1 set of TM-SDs or Ship To GLNs or Alternate Location Grouping should be populated for any 1 price.

Each of the 4 valid input price scenarios for TM-SD, Alternate Location Grouping, and Ship To needs to be cross-referenced with the same 4 valid existing price scenarios; in total 16 discrete Use Cases (see table below). The final column in the combined table shows whether geographic overlap occurs.

Geographical Overlap

Use Case	Input Price Scenario	Existing Price Scenario	Geographic Overlap
1	1	1	**Yes**
2	1	2	**Yes**
3	1	3	**Yes**
4	1	4	**Yes**
5	2	1	**Yes**
6	2	2	**Possible, if values are the same (set intersection non-empty)**
7	2	3	No
8	2	4	No
9	3	1	**Yes**
10	3	2	No
11	3	3	**Possible, if values are the same (set intersection non-empty)**
12	3	4	No
13	4	1	**Yes**
14	4	2	No
15	4	3	No
16	4	4	**Possible, if values are the same (set intersection non-empty)**

NOTE: It is important to realize that the absence of TM-SD and Ship To (null for both attributes) is interpreted in Australia / NZ as price applies to the 'entire' Target Market (that is all Australia or all NZ).

10.6.6. Checking Prices for Temporal Overlap

Once 'Geographical Overlap' has been established between an input price and an existing price, then '**Temporal Overlap**' must be established.

A price must have both geographical and temporal overlap to be considered a Duplicate Price. It is fine for the input and existing prices to have 'Geographical Overlap' if they do not have 'Temporal Overlap'.

Example: A $10 List Price for TM-SD 'Queensland' clearly has 'Geographical Overlap' with an 'equivalent' $11 List Price for TM-SD 'Queensland', but if one price only applies for calendar year 2012 and the other only for calendar year 2013, then there is no 'temporal overlap'. Thus, these are not Duplicate Prices

'Temporal Overlap' is a little more complex to establish than 'Geographical Overlap'.

Attribute	GDSN Attribute Name	Valid Context Type
Effective Start Date/Time	<priceTypeEffectiveStartDate> <effectiveStartDateTime>	FIRST_ORDER_DATE
Effective End Date/Time	<priceTypeEffectiveEndDate> <effectiveEndDateTime>	LAST_ORDER_DATE

Prior to checking temporal overlap, the exclusions and equivalent price checking have confirmed that the input price and existing price both contain only 1 set of Start and End Dates. All Start Dates are FIRST_ORDER_DATE Context Type and all End Dates are LAST_ORDER_DATE context type.

Of 5 total scenarios, there are 4 valid scenarios for the way in which an input price might be constructed for Start and End Date/Times as below noting the following:

- Start Date/Time are required, thus scenarios without Start Date/Time do not have a use case below.
- Date/Time can only be Historic (H) or Future (F) at the time a load file is being processed by atrify datapool. The notion of 'current date' or date TODAY has no meaning because this only applies instantaneously. Thus use cases only consider historic (past) or future dates.

The following use cases assume that all prior checking has confirmed that the input and existing price are equivalent prices that pass requirements for Duplicate Price checking and have geographical overlap.

Scenarios

Scenario	Start Date	End Date	Conditions
1	H	<null>	
2	H	H	Invalid use case as a price cannot be added with both a historic start and end date, nor can it be modified to set the end date to a past end date. Thus an **input** price can never have a historic end date.
3	H	F	
4	F	<null>	
5	F	F	End Date >= Start Date (both are future)

NOTE: A <null> end date is a Future end date set at infinity. In other words, it will always occur further in the future than a Future end date. This distinction is important in establishing when there is Possible vs. Definite overlap.

There are 5 valid scenarios for the way in which an existing price might be constructed for Start and End Date/Times, as below. The reason there are 5 (and not 4) valid scenarios is that an existing price may have an historic end date (which is not possible for an input price).

Each of these 4 input price scenarios needs to be cross-referenced with the 5 existing price scenarios; in total 20 discrete Use Cases (see table below).

Temporal Overlap

Use Case	Input Start Date	Input End Date	Existing Start Date	Existing End Date	Temporal Overlap
1	H	<null>	H	<null>	**Yes**
2	*H*	*<null>*	*H*	*H*	*Ignore (*)*
3	H	<null>	H	F	**Yes**
4	H	<null>	F	<null>	**Yes**
5	H	<null>	F	F	**Yes**
6	H	F	H	<null>	**Yes**
7	*H*	*F*	*H*	*H*	*Ignore (*)*
8	H	F	H	F	**Yes**
9	H	F	F	<null>	Possible
10	H	F	F	F	Possible
11	F	<null>	H	<null>	**Yes**
12	*F*	*<null>*	*H*	*H*	*Ignore (*)*
13	F	<null>	H	F	Possible
14	F	<null>	F	<null>	**Yes**
15	F	<null>	F	F	Possible
16	F	F	H	<null>	**Yes**
17	*F*	*F*	*H*	*H*	*Ignore (*)*
18	F	F	H	F	Possible
19	F	F	F	<null>	Possible
20	F	F	F	F	Possible

NOTE: ‘Possible’ means: If date intervals overlap (for each ‘Possible’ use case there is a positive and negative example).

NOTE: Duplication does not occur where temporal overlap between an input price and an existing price is ‘wholly historic’ based on timestamp, not just date (all overlap is in the past meaning the overlap period ends before today – this second).

NOTE: Temporal overlap, where it does occur, is defined to be non-zero duration. This means an input price starting at 00:00:00 does not overlap temporally with an existing price ending at 00:00:00 the same day.

10.6.7. Checking Prices for Bracket Overlap

If there is a geographical and a temporal overlap between an input price and an existing price, then ‘**Bracket Overlap**’ must be established.

Just like the absence of TM-SD, ALG and Ship To means that the ‘superset’ NATIONAL geographic region is defined, similarly the absence of Brackets means the price has the ‘virtual’ definition of having ‘infinite’ brackets – in other words a price without any Brackets is deemed to have Brackets from minus infinity to plus infinity; thereby creating possible overlap scenarios with those prices which do have Brackets defined.

Example:

- NATIONAL $10 LIST_PRICE for GTIN(x) 2013-01-01 to 2013-12-31 (no Brackets)
  vs.
- NATIONAL $12 LIST_PRICE for GTIN(x) 2013-01-01 to 2013-12-31 for brackets 1-10

This is a case of Price Duplication because for units of GTIN(x) from 1 to 10 there are 2 possible prices.

In order to test for ‘Bracket Overlap’ two steps must be performed:

1. Check Bracket UOMs.

The possible maintenance combinations create three valid scenarios (Input Price UOM Bracket = Existing Price UOM Bracket).

Scenarios

Scenario	UOM Bracket Min	UOM Bracket Max	Comment
1	<null>	<null>	
2a	populated	<null>	
2b	<null>	populated	Invalid
3	populated	populated	

Each of these 3 input price scenarios needs to be cross-referenced with the 3 existing price scenarios; in total 9 discrete Use Cases (see table below).

Bracket Comparison

Use Case	Input Price Scenario	Existing Price Scenario	Bracket Comparison Possible
1	1	1	Yes
2	2a	1	Yes
3	3	1	Yes
4	1	2a	Yes
5	2a	2a	Yes, if UOM Min equal
6	3	2a	Yes, if UOM Min equal
7	1	3	Yes
8	2a	3	Yes, if UOM Min equal
9	3	3	Yes, if UOM Min/Max equal

2. If UOMs are the same, check the Bracket min/max values.

Bracket Overlap

Use Case	Input Price Scenario	Existing Price Scenario	Bracket Overlap
1	1	1	Yes
2	2a	1	Yes
3	3	1	Yes
4	1	2a	Yes
5	2a	2a	Yes
6	3	2a	Possible
7	1	3	Yes
8	2a	3	Possible
9	3	3	Possible

NOTE: 'Possible' means: If bracket intervals overlap (for each 'Possible' use case there is a positive and negative example).

11. Message grouping

The GDSN choreography distinguishes between four types of GDSN response messages (called GDSN Responses):

Catalogue Item Registration Response (positive message),

Party Registration Response (positive message),

GS1 Response (positive message) and

GS1 Exception (negative message).

The Catalog Item Registration Response and the Party Registration Response serve the same purpose of accepting requesting message, but the message is exclusively sent by the Global Registry. Only GS1 Response and GS1 Exception are relevant for this document.

Customers can opt to receive only one validation report per message instead of multiple reports. The validation report contains all the validation information for all the hierarchies sent in a single message. For all correct items in the message the user will receive one GS1 Response and for all incorrect items the user will receive one GS1 Exception.

CIN XML messages with multiple transactions coming from another Data Pool can either be split into N messages each containing one transaction, before they are forwarded to the atrify Data Recipient or customers can opt to receive all transactions grouped into one message. Customers that have not selected the message grouping option will receive multiple reports. Number of maximum transactions per message is configurable for each customer GLN.

12. Out of Network

If atrify datapool parties are set “IN Network” = “false” then those parties do not participate in the GDSN network. Means that no data exchange happens to Trading Partners using other Data Pools in the GDSN network. Those “Out-of-network” parties exchange data only with Trading Partners (Supplier, Retailer) within the atrify datapool community.

As a result items of atrify datapool Suppliers will not be registered in the Global Registry, and subscriptions of atrify datapool Retailers will not be stored in the Global Registry.

Usually local communities like the German AGRO community have their parties configured out of network, as both Retailer and Supplier are members of the atrify datapool and those parties don´t want their data to be exchanged globally.

13. Batch Processing

Data Recipients can enable M2M batch processing and can define their individual interval for their own profile, e.g. every five minutes, every hour or once per day (so called “Subscription Frequency”). Once the processing time is reached the system sends the latest version of all new and changed items to the DR. Data will be packaged depending on the maximum transaction configuration (e.g. 100 per message). Alternatively, real time processing can be chosen.

The figure below shows an example of a batch process that works with 7 hours’ interval of processing. During that period the Data Sender sends a new item (GTIN1) with the action request ADD on hour one. On hour three, the data sender provides an update of GTIN 1 with the action request CHG (change). Finally, the Data Sender sends a new update of the item on hour six. If the processing time is reached (hour seven in this example) the data will be sent out to the Data Receiver. In this example the atrify datapool would send out only the last version of GTIN 1. This would be CIN 3, which was received by the Data Pool on hour 6.

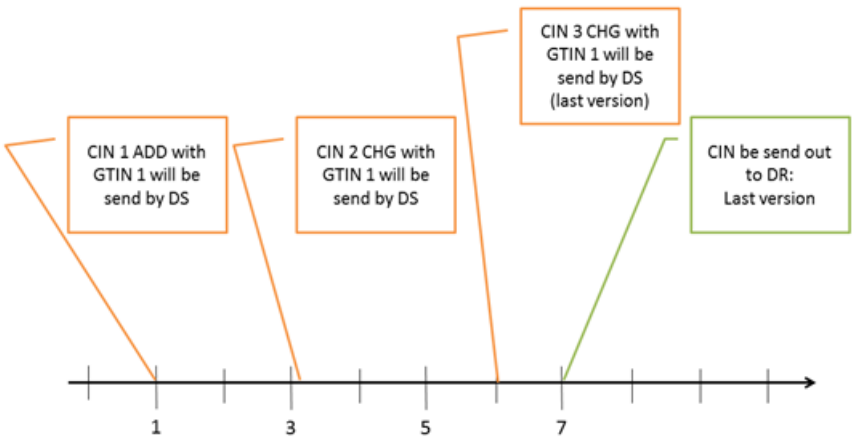


Figure 15: Example Batch Processing: 7 hours

Please contact the support team if you want to set up a special batch processing interval.

14. 1SYNC XML Interface Support

atrify datapool supports regular standard GDSN BMS message types and choreography. In addition the datapool has the ability to support the basic 1SYNC XML message choreography. To activate the 1SYNC XML flow for your GLN get in touch with the atrify Support Team or your assigned contact.

14.1. How to setup the 1SYNC XML Flow

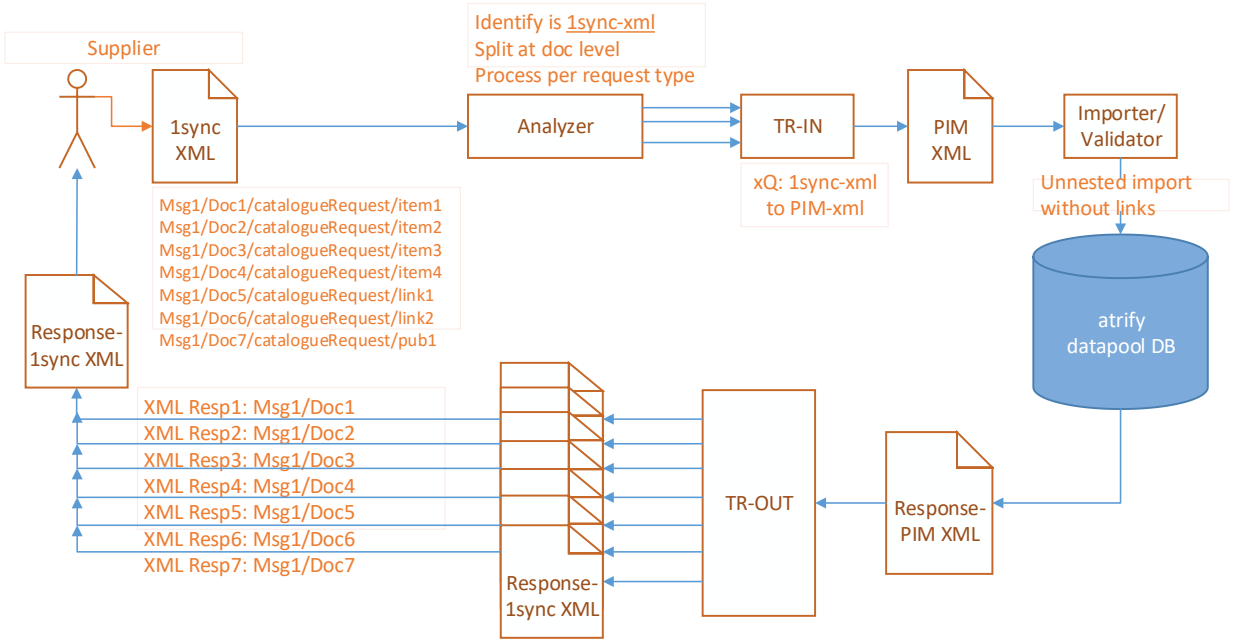
To enable the 1SYNC XML flow open the Administration / Party menu. Either edit existing parties or add a new party and set

- Message Format Type: 1SYNC XML
- Message Format Version: 2.0

Important note: Supplier must decide for a given GLN whether to use the GDSN **OR** the 1SYNC XML flow. Using both under the same GLN remains possible, but due to the above setting will trigger a CIC to IAR mapping, if the recipient sends back a CIC feedback.

14.2. How the 1SYNC XML Process flows

Below illustration shows how the 1SYNC XML workflow works for the item, link and publication scenarios. Also, the response mechanism is shown here:

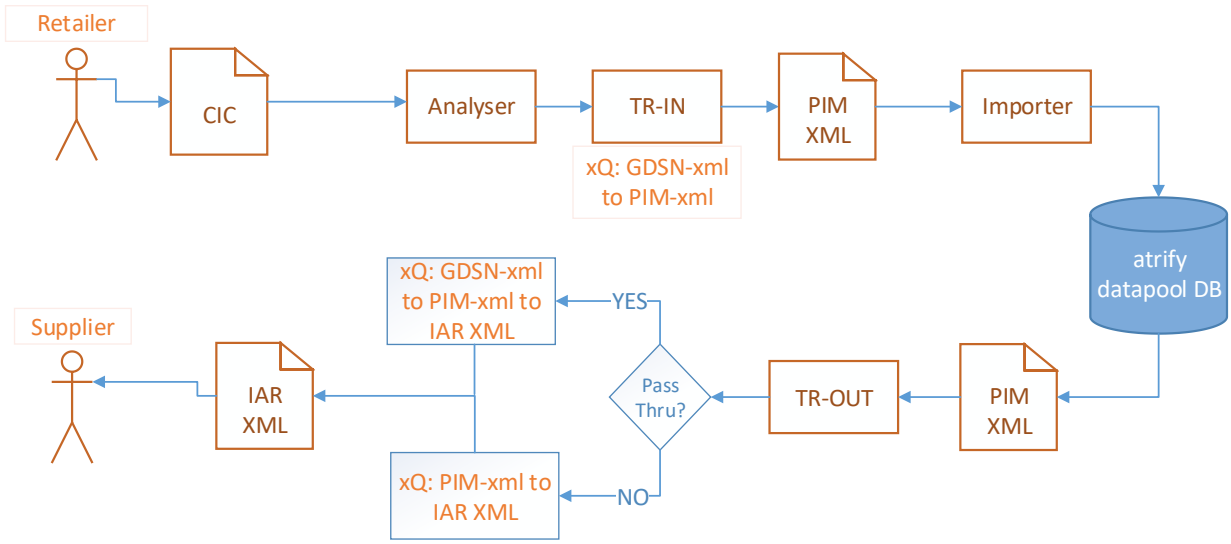


The diagram depicts the following scenario:

1) Data Source (Supplier) sends new item add as 1sync XML to the atrify datapool. Message may contain single catalogue requests of type item and/or link and/or publication. If many catalogue requests are sent in one message, they will be structured per message id / document id. Processing of the multiple requests will happen per request type. Means first all items will be processed, then links and then publications.

2) The analyzer component will receive the xml file and will recognize as 1sync XML format. Then the 1sync message handler will split the messages per document level and send to the Translator-in (TR-IN) component. The translator will trigger the xQuery mapping which will generate the native datapool PIM-XML file.
3) The Importer will process the PIM-XML file as unnested hierarchy and hand over to the validator to get the validation result. In success case datapool will store the unnested hierarchy level without the link information.
4) The importer will receive the good / error response and hand over to the Translator-out (TR-OUT) component. The translator will trigger the xQuery mapping to generate a response file in 1sync XML format.
5) NOTE: The datapool validator is using GDSN attribute names to express the error details. atrify will send these GDSN attribute names to the supplier back.
6) Per each request type the above process steps will be executed. As a result, a response message will be sent to the supplier referring to all documents send in the original message.

Next illustration shows how the process for recipient feedback with 1SYNC XML IAR (Item Authorization Response) messages associated to the GDSN CIC works. Simply the process is like the above scenarios. We will get GDSN CIC messages and convert to IAR messages:



14.3. Supported Attributes and Message Types

The interface supports attributes on demand. Current focus lies on Healthcare FDA, NHS and FMCG for Consumer Electronics. The attribute scope will be enhanced based on customer demands.

Message Request Types

- Item
- Link
- Publication

Message Response Types

- Acknowledgement Response
- Exception
- IAR

14.4. How to Track&Trace 1SYNC XML messages

To track 1SYNC XML messages go to menu “Data Sync / Track Messages” and check for one of the following document types:

- CR: catalogueResponse
- CRQ: catalogueRequest (Item, Link, Pub)
- IAR: itemAuthorisationResponse

14.5. Further Notes

1) Attribute naming in responses and CIC/Item Authorisation messages are based on the GDSN + FMCG names
2) Any attributes that are not part of the mapping scope will be ignored. Please make a change request if further attributes are needed.

15. Special atrify datapool Features

atrify datapool offers some add-on features as described in the following chapters. To get these features please get in touch with the atrify Support Team or your assigned contact.

15.1. Hybrid Client

15.1.1. Hybrid Client Exchange

The ability for Publishing to support routing and special party administration for Hybrid Client. This allows M2M suppliers to load their initial items in XML format (BMS/1SYNC) via atrify datapool into the Publishing and further manually maintain additional data here.

Special impacts for Hybrid Client M2M Suppliers

- If Supplier further maintains their data in Publishing all further updates to that hierarchy must happen in Publishing! If supplier sends updates via M2M all changes in Publishing will be over-written.
- Data will be immediately released in Publishing if the CIN message was valid.
- So far Publishing generated CIN messages and sent to the atrify datapool the first time when items were published. Now items will be send right after being released.
- Any invalid item data will be set to working state in Publishing if the data caused an exception in the atrify datapool. For corrupt data (e.g. missing primary key information GTIN/GLN/TM) the data will even not be imported into working state.
- Invalid price data will never be imported into the hybrid Publishing, valid prices only.

15.1.2. Hybrid Copy Feature

The Hybrid Copy feature enables the atrify datapool to copy existing items & prices from M2M suppliers into the Hybrid Publishing. This helps M2M Supplier to get their already stored data copied into the appropriate Hybrid Publishing. Usually the M2M Supplier data gets routed to the current Hybrid Client workflow is serving the regular M2M process and is routing the Hybrid Publishing if they send messages. This feature allows to copy Items & Prices per Supplier GLN. Also copying single hierarchies or GTIN lists is possible.

Note: Copy works for CIN, CIP, PSD messages only. So CIC and PSC copy is not supported!

After completion of the Hybrid Copy process, the party attribute “hybrid” for the Supplier GLN will be set to 'Y'. Also the party attribute “publishingGln” will be set for the selected Supplier GLN through the UI.

15.2. email Exception Report

atrify datapool is able to generate a human-readable csv file containing any exceptions for a given message (CIN / PSD). For M2M suppliers having activated this feature a csv report will be send via email containing the atrify datapool load errors for ITEM and/or PRICE files. This report will be send in addition to the GS1 Exception message that is part of the regular workflow. Mostly the stuff maintaining the master data for the items is not the EDI expert who is in charge of the EDI processes. That´s why Supplier can activate this feature to get the exceptions send via email in csv format to the master data manager´s dedicated email address.

Three types of errors are identified in the report.

- Item Errors
- Price Errors (excluding Price Duplication Errors)
- Price Duplication Errors (Price Duplication Errors are separately identified as they require an extra level of detail.)

Beyond the email exception feature the standard atrify datapool Party settings allow to “Accept Exception and/or Response” messages to be activated or deactivated. So a M2M party can decide not to receive and GS1 Responses or Exception messages.

15.3. Price Auto Publication

Price Auto Publication allows trade items automatically published to the Recipient declared in the Price message. After receiving a new price in the atrify datapool, all hierarchies for the referenced trade item will be identified and published based on the below conditions:

- If the hierarchy has not been published
- If the publication of a hierarchy has not been deleted.

This is not applicable for Price Relationship and applicable only for Price Segment with Action Code as “ADD”. Auto Publication with Price can be configured on Supplier level per Retailer. The Retailers benefit from this function as they always will get trade item data with prices, if prices exist.

15.4. Item Append

This is the ability to add proprietary action code for atrify Suppliers to avoid previous version is overwritten by new input version. Supplier has to set the document command header action code to “APPEND”. This will be treated as a “CHANGE_BY_REFRESH” but will trigger the new APPEND feature.

How the atrify datapool APPEND works:

When supplier sends action code “APPEND” atrify datapool will load the existing data and merge with the incoming data only if the existing attribute class is not given in the input file. If the attribute class exist in the existing version but not in the new version atrify datapool will keep the attribute class from the existing (old) version. Further if existing and input item contain an attribute class (e.g. nutrient information) then the incoming content will completely remove the existing content. Please note this works for M2M and hybrid supplier only. So for pure publishing UI Supplier this feature is not supported.

In the hybrid use case usually the incoming CIN XML message will be sent to the Hybrid Client Importer in the publishing. But if a hybrid supplier sends a CIN XML with APPEND it will be merged with the existing item record and then the “appended” version will be sent to the hybrid client importer of the publishing. Important note: APPEND is not supported for the 1sync xml choreography.

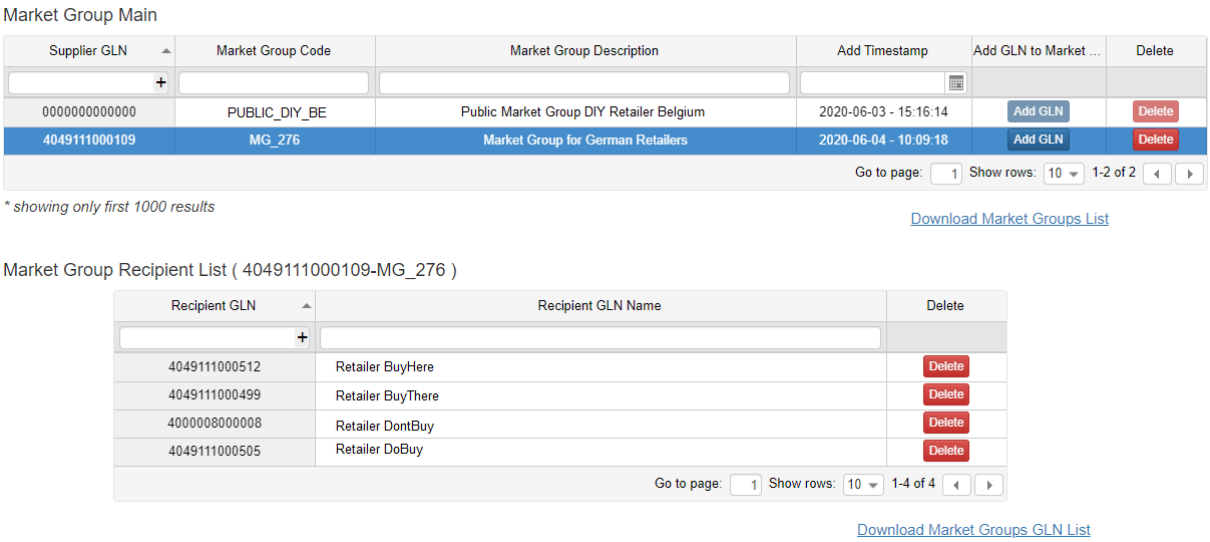
15.5. Publish to Market Groups

This is the ability to publish to a defined set of GLN named as “Market Group”. The GDSN standard supports publish to target market or publish to single GLN only. There is no way to define a set of GLNs and publish to all in one go.

How the atrify datapool Market Group Publication works:

1. Define Market Group

First the supplier must define the market groups they want to publish to in one go. Every market group must have at least 2 GLN and maximum can have 50 GLN. Using the datapool UI “Market Groups Main” the supplier has to define an alphanumeric upper case Market Group Code and add the appropriate recipient GLNs. All defined market groups will show up in the Market Group UI hitlist. Clicking on a row will show the Market Groups Recipient List below. Suppliers will see their own defined private market groups as well as the public market groups setup by the atrify support administrators.



2. Publish to Market Group

After the market group codes were defined the supplier can use those to publish to the appropriate market group. To publish to market group the supplier uses the regular CIP message and adds the market group code to the “publishToTargetMarket” field (see example below).

```
<catalogue_item_publication:catalogueItemPublication>
  <creationDateTime>2019-12-12T14:27:30.010Z</creationDateTime>
  <documentStatusCode>ORIGINAL</documentStatusCode>
  <documentStructureVersion>3.1</documentStructureVersion>
  <lastUpdateDateTime>2019-12-12T14:27:30.010Z</lastUpdateDateTime>
  <catalogueItemPublicationIdentification>
    <entityIdentification>4cfd6df2-b5ed-45cc-8ffc-79518c25d403</entityIdentification>
    <contentOwner>
      <gln>4049111000109</gln>
    </contentOwner>
  </catalogueItemPublicationIdentification>
  <publishToTargetMarket>
    <targetMarketCountryCode>FMCG_DE</targetMarketCountryCode> <!-- Trigger for Market Group
Publication Process -->
  </publishToTargetMarket>
  <catalogueItemReference>
    <dataSource>4049111000109</dataSource>
    <gtin>08901111831144</gtin>
    <targetMarketCountryCode>276</targetMarketCountryCode>
  </catalogueItemReference>
</catalogue_item_publication:catalogueItemPublication>
```

The atrify datapool will create individual CIP to GLN for each recipient GLN in the market group. All auto generated publication messages will be validated as usual. A response will be sent back for the initiating market group publication message. As all auto generated GLN publications refer to the market group publication message, they will be accepted or rejected as a whole. Means only if ALL GLN publications are valid the market group publication message will be stored and will get back a positive response. If only one publication is invalid the complete market group GLN publication will be rejected and will receive an exception with the causing error message.

3. Editing Market Groups

3.1 Add GLNs from existing Market Groups

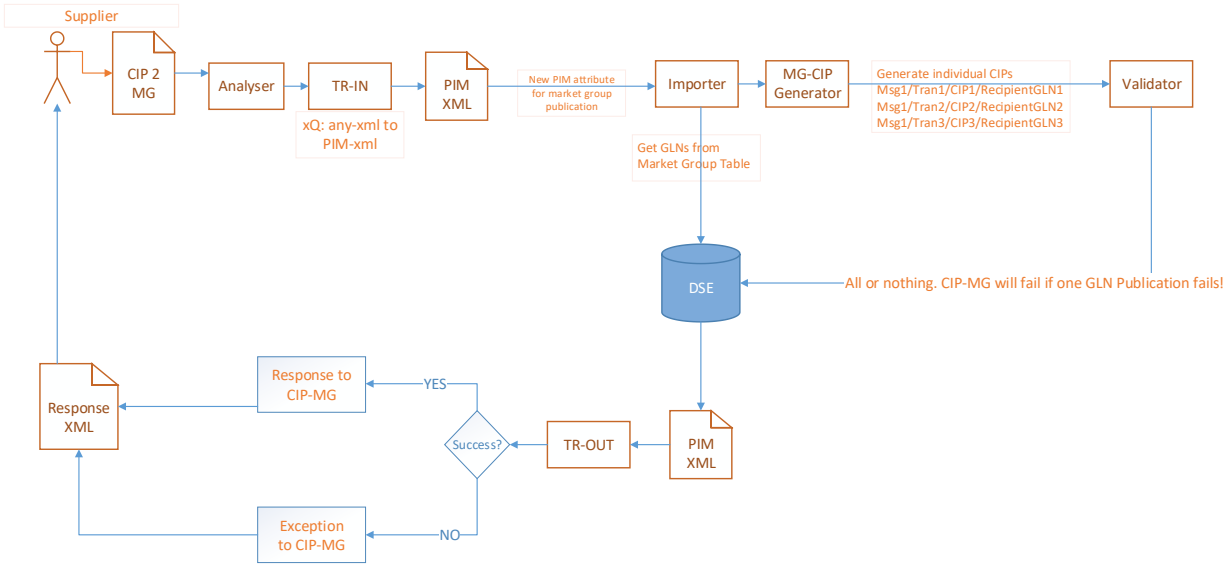
Adding a new GLN to an existing market group will not reprocess all previously published items. It will be considered for future item publications only. If the supplier wants to have previously published items to be synced with the new recipient GLN a single CIP to GLN must be created for that new recipient.

3.2 Remove GLNs from existing Market Groups

Supplier can remove GLNs from existing groups. For future item publications the deleted GLN will not be considered anymore when published to this market group. But for existing item the publication will remain. Deleting a GLN will not trigger a Publication Withdrawal/Delete message. This must be done by the supplier separately.

3.3 Delete Market Groups

Supplier can delete the complete market group from the “Market Group Main” UI. This will remove the market group from the hitlist. When afterwards publishing to a deleted market group code the system will throw an error “Unknown Market Group Code”. Again, deleting complete market groups will not delete the individual publications to the recipient GLNs. If needed those must be deleted separately.



Note: Publication to market groups works with BMS CIP message only and covers the hybrid use case. 1SYNC XML is not supported.

16. Index

17. History of changes

Date	Chapter / Page	Change / Enhancement / Innovation
April 2014	**Version 1.000**	
	Document first published April 2014,Version 1.0	
Created by	Nora Schuler & Sascha Kasper	
May 2014	**Version 1.100**	
	Chapter 6.1: Deleted E-Mail and valid formats. Chapter 10: Deleted E.Mail and XML-Upload. Chapter 11: Deleted Japan Out of Network aspects.	
Changed by	Sascha Kasper	
September 2014	**Version 1.200**	
	Chapter 4.2: Change sentence Chapter 6.2: Add overview message handling changes between WS2 and DSE Chapter 7.3: Stop service of CDN-message Chapter 7.4: Reset of Sync List by Data Source via CIP Delete Chapter 9.2: Switch on or off of validations per GLN isn´t possible Note: Handling of line breaks are not part of this documentation.	
Changed by	Sascha Kasper	
May 2015	**Version 1.300**	
	Chapter 8: New subchapter “Item hierarchy basic rules added” Chapter 10: New chapter “Price Synchronisation” added Appendix A – Error List DSE added	
Changed by	Selcuk Övüc	
July 2015	**Version 1.500**	
	Appendix A – Error List DSE updated	
	Selcuk Övüc	
June 2017	**Version 2.700**	
	Complete rework of the M2M Guide. Updated from BMS 2.7 to 3.1 (Major Release).	
	Appendix A – Error List DSE removed. Please refer to GDSN standard validation rule set.	
Changed by	Selcuk Övüc	

September 2019	**Version 3.0**
	Made changes to replace the 1WorldSync logo and name by atrify logo and name.
Changed by	Selcuk Övüc
March 2020	**Version 3.1**
	Changed DSE naming to atrify datapool in the text
	Chapter 3: AS2 connectivity details updated.
	Chapter 14: New section for 1SYNC XML messaging added.
	Table2: List of important messages Updated the Catalogue Item Hierarchy Withdrawal explanation regarding deletions for target market publications.
Changed by	Selcuk Övüc
March 2020	**Version 3.1.1**
	Changed DSE naming to atrify datapool on images
Changed by	Selcuk Övüc
April 2020	**Version 3.1.2**
	Chapter 4.3. IsReload flag description updated.
Changed by	Selcuk Övüc
April 2020	**Version 3.1.3**
	Chapter 15.4. Item Append description updated. With 20.05 release m2m **and** hybrid supplier can use the item append feature.
Changed by	Selcuk Övüc
May 2020	**Version 3.2**
	New chapter 15.5 “Market Group Publication” added.
Changed by	Selcuk Övüc
August 2020	**Version 3.2.1**
	Figure 11: Unnested vs. Nested hierarchy diagram changed.
Changed by	Selcuk Övüc
Dec 2020	**Version 3.2.2**
	Chapter 15.5 “Market Group Publication” updated. Increased limit for Market Groups GLN to 50.
Changed by	Selcuk Övüc

Feb 2021	**Version 3.2.3**
	Chapter “Terms of use, disclaimer” added.
Changed by	Selcuk Övüc
Mar 2021	**Version 3.2.4**
	Chapter 1.1. “About atrify” updated
Changed by	Selcuk Övüc
Apr 2021	**Version 3.2.5**
	Chapter 7.1.2 “Deviation from BMS T.P.D. Definition” added.
Changed by	Selcuk Övüc
Feb 2022	**Version 3.2.6**
	7.1.1. Trading Partner Dependent Attributes (TPD) New TPD Blob Delete feature via Publishing added plus further explanation how TPD blobs are stored
	10.2.3. Merge Item Depiction Qualifier Explanation for ability to merge multiple price types below one Item Depiction Qualifier added.
Changed by	Selcuk Övüc

Table 8: History of changes