



# GS1 XML MR3 eCom Technical User Guide

Issue 1, June 2012



## Document Summary

Document Item	Current Value
Document Title	GS1 XML MR3 eCom Technical User Guide
Date Last Modified	08.06.2012
Current Document Issue	1
Status	For publishing
Document Description	

## Contributors

Name	Organization
Ewa Iwicka	GS1 GO
Coen Janssen	GS1 GO

## Log of Changes in Issue #

Issue No.	Date of Change	Changed By	Summary of Change
1	June 2012	Ewa Iwicka	1 <sup>st</sup> Issue

## Disclaimer

Whilst every effort has been made to ensure that the guidelines to use the GS1 standards contained in the document are correct, GS1 and any other party involved in the creation of the document HEREBY STATE that the document is provided without warranty, either expressed or implied, of accuracy or fitness for purpose, AND HEREBY DISCLAIM any liability, direct or indirect, for damages or loss relating to the use of the document. The document may be modified, subject to developments in technology, changes to the standards, or new legal requirements. Several products and company names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

# Table of Contents

<b>1. Introduction .....</b>	<b>5</b>
1.1. Purpose of this Document.....	5
1.2. Scope of this document.....	5
1.3. Target audience .....	5
<b>2. GS1 set of XML standards .....</b>	<b>5</b>
2.1. Background – GS1 eCom .....	5
2.2. Business Message Standards (BMS) .....	5
2.2.1. Business Message Standard specification.....	6
2.2.2. Implementers Packet.....	6
<b>3. GS1 XML versioning.....</b>	<b>6</b>
3.1. Minor versions .....	7
3.2. Major versions .....	7
3.3. Specifying version numbers .....	7
3.3.1. Specifying version numbers in the XML schema file .....	7
3.3.2. Specifying version numbers in the XML instance file.....	8
3.3.3. Specifying version numbers in the Standard Business Document Header .....	8
3.4. Publication of different schema versions .....	8
<b>4. Code lists .....</b>	<b>8</b>
4.1. Code lists defined and maintained outside of GS1 .....	9
4.1.1. GS1 subsets of code lists.....	9
4.1.2. GS1 extensions of code lists.....	9
4.1.3. Restricted list of code lists.....	9
4.1.4. Bilaterally defined code lists .....	10
4.2. Code lists defined and maintained within GS1 .....	10
4.3. Storing of code list values .....	10
4.4. Code list versioning.....	11
<b>5. Communicating GS1 trade item numbers.....</b>	<b>11</b>
<b>6. XML syntax components used in GS1 XML 3.0.....</b>	<b>12</b>
6.1. XML schema design model.....	12
6.2. XML built-in types.....	12
6.2.1. 'xsd:string' Pattern facet .....	13
6.2.2. Handling time stamps and time zones .....	13
6.2.2.1. Use of milliseconds .....	13
6.2.2.2. Use of time zones .....	14
6.3. XML extension element.....	14
6.3.1. Definition of the extension content .....	15
6.3.2. Validation of the extension components .....	15
6.3.3. Applying an extension to the instance file .....	15
6.4. Use of XML elements vs. attributes .....	16

<b>7.</b>	<b>GS1 namespaces</b> .....	<b>16</b>
7.1.	GS1 namespace structure .....	16
7.2.	GS1 namespace prefix.....	17
7.3.	XML schema namespace and prefix.....	17
7.4.	Qualifying of GS1 XML schema components .....	17
7.5.	GS1 namespace use.....	18
<b>8.</b>	<b>GS1 XML schema hierarchy</b> .....	<b>19</b>
8.1.	Installation of GS1 XML schemas .....	19
<b>9.</b>	<b>GS1 XML message architecture</b> .....	<b>20</b>
9.1.	Overview of eCom Message Architecture.....	20
9.2.	Benefits .....	21
9.3.	Overview of eCom message components .....	21
9.3.1.	Transport and routing component .....	21
9.3.2.	Business document component .....	22
9.4.	Batching of business documents .....	25
<b>10.</b>	<b>Main differences between the GS1 XML 3.0 and 2.0</b> .....	<b>26</b>
10.1.	Reasons for change .....	26
10.2.	Main changes in the GS1 XML design.....	27

# 1. Introduction

## 1.1. Purpose of this Document

This document is intended to clarify the technical aspects of the GS1 XML Business Message Standard Major Release 3.

## 1.2. Scope of this document

This "GS1 XML MR3 eCom Technical User Guide" covers only technical aspects of the GS1 Business Message Standard for eCom Major Release 3. It does not explain the technical details of any earlier releases. The major differences between the current release (MR3) and the previous one (MR2) are listed in the [Section 9](#).

Users who need further information about the Major Release 2, should refer to the document [GS1 XML Technical User Guide to Release 2.0](#)

"GS1 XML MR3 eCom Technical User Guide" does not contain any guidelines for use of the particular business messages in specific business processes. This is covered in the Business Message Standard specification documents, separately for each business message. Certain specific constraints may also apply to other business processes, therefore, it is advisable to contact the respective user group for further guidance (e.g. Global Upstream Supply Initiative).

✔ **Note:** The Global Data Synchronisation Network – GDSN messages are based on Major Release 2 design. Users involved in the GDSN should refer to the [GS1 XML Technical User Guide to Release 2.0](#) and to [GDSN 2.1 Operations Manual](#).

## 1.3. Target audience

This document is intended to serve as an implementation guide for business and technical community who will implement GS1 XML standards. It can also be used by the GS1 Member Organisation employees, to educate their user companies about the implementation of the GS1 XML standards.

# 2. GS1 set of XML standards

## 2.1. Background – GS1 eCom

GS1 provides two complementary sets of standards for electronic business messaging, referred to as eCom: GS1 EANCOM<sup>®</sup> – for “classic” Electronic Data Interchange (EDI) and GS1 XML, using a newer XML syntax. Both sets of standards provide business documents converted into a format that can be electronically exchanged between computer applications without human intervention.

In case of the GS1 XML business messages are represented as the XML instance. The structure and content of these documents is defined in GS1 XML schemas – published as the GS1 standard.

## 2.2. Business Message Standards (BMS)

The documents necessary to implement GS1 XML messages are published as the GS1 BMS packets.

The BMS packets from the MR 3 release can be downloaded from the following link:

[http://www.gs1.org/gsmp/kc/ecom/xml/xml\\_bms](http://www.gs1.org/gsmp/kc/ecom/xml/xml_bms)

All the previous releases can be downloaded from

[http://www.gs1.org/services/gsmp/kc/ecom/xml/xml\\_archive.html](http://www.gs1.org/services/gsmp/kc/ecom/xml/xml_archive.html).

Each BMS packet includes:

- Business Message Standard specification
- Implementers Packet

### 2.2.1. Business Message Standard specification

The BMS is a document containing specification of one or more business messages. Users must refer to this document when implementing the GS1 XML standards, since the XML schemas do not contain all the necessary definitions, due to the syntax limitations (e.g. dependencies).

Besides, the BMS documents contain definitions of all the message components, code list values as well as implementation considerations.

The BMS specification is published as a PDF file.

### 2.2.2. Implementers Packet

The implementer's packet is a ZIP file, containing all the XML files necessary for validating the given XML message. It consists of:

- **TableofContents.txt** – a text file listing all the files included in the given packet
- **Instance File** folder - contains at least one (out of many possible) sample XML file of the message
- **HTML Sample** folder - contains the HTML representation of the sample XML file from the Instance File folder
- **Schemas** folder, with the following content:
  - **gs1** folder – contains two subfolders:
    - **ecom** – includes one schema defining the [eCom Common](#) library - these are components that can be reused in any eCom business document - and a schema (or schemas) defining the business document structure
    - **shared** – includes one schema defining the [Shared Common](#) library - these are components that can be reused in any eCom or GDSN business document
  - **sbdh** folder – contains the [Standard Business Document Header](#) schemas

## 3. GS1 XML versioning

The GS1 XML suites of standards are published as major and minor versions. The minor versions are backward compatible.

**Backward compatibility** means that a message sender can create a business document based on an earlier schema and send it to a recipient, who can successfully validate it against a later schema.

The GS1 versioning uses three numbers to denote a change or update to the standards:

- **N** – used for major changes that are not backward compatible to earlier publications of XML standards
- **N.n** – used for publication updates that are backward compatible within the same major release
- **N.n.n** – used for minor changes: errata, navigation, document updates, new guidelines, etc., that are backward compatible within the same major release

### Example

If the Order BMS 2.6.0 required a change to the common library that affected all messages, the new version was renamed Order BMS 3.0.0.

## 3.1. Minor versions

Minor versions are interim releases of an XML schema that contain only the changes that are [backwardly compatible](#) with the existing version of this schema.

The following changes can be incorporated in a minor version:

- Adding new optional elements or optional attributes
- Changing attributes cardinality from mandatory to optional
- Changing element multiplicity from old schema [0..1] to new schema [0..\*]
- Changing element multiplicity from old schema [1..1] to new schema [1..\*]
- Adding a value to an enumerated list

## 3.2. Major versions

Major versions are the releases of a suite of XML schemas that contain changes **NOT** [backwardly compatible](#) with the existing suite of the same schemas.

The following changes require incrementing the major version:

- Changing an attribute cardinality from optional to mandatory
- Adding a mandatory element
- Changing an attribute or element tag name
- Changing element multiplicity from old schema [0..\*] to new schema [0..1]
- Changing element multiplicity from old schema [1..\*] to new schema [1]
- Changing the sequence of elements in a `<xsd:sequence>` tag

## 3.3. Specifying version numbers

### 3.3.1. Specifying version numbers in the XML schema file

The major version number of the schema is specified in the schema namespace.

#### Example

```
urn:gs1:ecom:order:xsd:3
```

The minor versions are reflected in the version attribute of the `<xsd:schema>` element. For the ease of implementation, the internal schema version attribute reflects the entire version (both major & minor) of the schema.

#### Example

```
<xsd:schema version="3.0">
```

This approach ensures [backward compatibility](#). Due to the fact that the minor versions are not reflected in the namespace of a schema, instance documents can be validated against the older minor version of the schema.

The full version number (major and minor) is also specified in the `<xsd:annotation>` element of the schema. This is a non-parsed piece of text containing information for the users.

#### Example

```
<xsd:annotation><xsd:documentation>
....
Version Information:
Version Number: 3.0
Date of creation: August 2011
....
</xsd:annotation></xsd:documentation>
```

### 3.3.2. Specifying version numbers in the XML instance file

Since the major version is specified in the namespace, the business document (XML instance document) contains the major version by default. However, the complete version also needs to be specified in the instance document, as the application systems that receive the business documents need to know what version the trading partners are using.

The minor version is reflected in the optional 'documentStructureVersion' element. For the ease of implementation, it reflects the entire version (both major & minor) of the schema used as a base for creating this document. E.g. for the Order.xml message, the 'documentStructureVersion' reflects the version number of the Order.xsd schema.

#### Example

```
<documentStructureVersion>3.0</documentStructureVersion>
```

### 3.3.3. Specifying version numbers in the Standard Business Document Header

The [Standard Business Document Header](#) schema contains the 'TypeVersion' element, which is the placeholder for the version of the business document or business documents sent with the one header. The SBDH specification requires that all the documents sent with one header have the same version number.

To comply with this requirement, GS1 recommends that only business documents belonging to the same BMS publication release and having the same version number are included in the payload, if sending more than one document in the same message (see: [Batching of business documents](#)).

## 3.4. Publication of different schema versions

The [minor versions](#) of GS1 XML schemas are published as a part of a given [major version](#) of the standards. The next minor release always contains only these messages that have been changed in any way. All the other messages will remain the part of the previous minor release. It should be noted, however, that due to the [backward compatibility](#) of the common components, there is no problem in using messages from different minor releases in the same message choreography, e.g. Order 3.0, Despatch Advice 3.2 and Invoice 3.5.

- ✔ **Note:** The file names of the GS1 XML schemas do not contain version numbers, so e.g. both Order 3.0 and Order 3.1 will have the file name: Order.xsd.

## 4. Code lists

Code lists are the standardised lists of values allowed at the given location of the eCom message. GS1 uses values from two types of sources:

- Code lists defined and maintained outside of GS1 - by external organisations (agencies)
- Code lists defined and maintained by GS1



## 4.1. Code lists defined and maintained outside of GS1

Most code lists used in GS1 eCom standards are defined and maintained outside the GS1 community, usually by other standard bodies, e.g. Country Codes, ISO 3166-1:1997. GS1 uses international standard code lists whenever available in the following order of priority:

- ISO
- UN/ECE
- UN/CEFACT
- International Vertical Organisations, e.g. World Customs Organisation
- Codes created by GS1

If a suitable code list exists at a higher level, a code list from a lower level is not used. For example, if there are a suitable code lists managed by ISO and UN/ECE, the ISO list will be used in the GS1 XML 3.x

Regional standards, such as CEN and ANSI, are not used except for purely regional standards and extensions.

The re-use of existing standard lists allows implementers of GS1 standards avoid mapping of different code values from different code lists representing the same information.

### 4.1.1. GS1 subsets of code lists

Based on the user requirement, GS1 can create a code list that is a subset of the managing agency's list. It ensures interoperability for trading partners by adopting the list of values from the managing agency, while limiting the list to values needed to support the specific business requirements of the GS1 community.

#### Example

The UN/EDIFACT code list: 4221 "Discrepancy nature identification code" has been restricted to three values relevant for the GS1 users. The list has been named 'RemainingQuantityStatusCode' and is published in the BMS eCom Common Library.

### 4.1.2. GS1 extensions of code lists

Whenever a value needed by the GS1 users is missing in a non-GS1 code list, it can be added as a "GS1 Code". In parallel GS1 will submit a request to the managing agency to formally add the code value to the original list.

This approach allows extensibility of existing Agency Code Lists, enabling trading partners to use GS1-developed codes when needed.

#### Example

The UN/ECE code list: 'PackageLevelCode' (see: BMS eCom Common Library) has been extended by adding one GS1 value "1E Highest (GS1 Code)".

- ✔ **Note:** GS1 assigns a temporary code value that will be changed by the managing agency once the code is approved as a formal addition to the original code list. Thus, the early implementers should be aware that the code value will change over time.

### 4.1.3. Restricted list of code lists

This is a type of code list where each value identifies a code list that can be referenced for the list of values. Trading partners are expected to mutually agree which list will be used, then select and send the code list value that matches the business requirement.

This approach provides flexibility for trading partners to select the code list appropriate to their usage, while also restricting the list to those that have been identified as valid options to support a specific context.

#### Example

A Colour Code List with values such as “1” representing the “National Retail Federation (NRF) – Standard Colour & Size Codes”; “2” representing the “Pantone Matching System”.

If the restricted list selected by trading partners is the “National Retail Federation Standard Colour Code List”, and the manufacturer wants to send a colour code that represents “dark blue”, they could send the code value of “401” (the NRF 3-digit numeric code for dark blue).

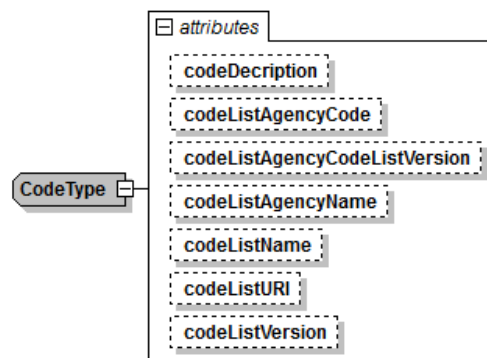
### 4.1.4. Bilaterally defined code lists

These code lists should be mutually agreed between trading partners, because GS1 does not specify the code list to be used. They are typically used when there are multiple code lists that could be implemented depending on the industry, geopolitical region, e.g. regulatory or tax codes.

It provides flexibility for trading partners to mutually define the code list appropriate to their usage.

#### Example

These lists are defined as the ‘CodeType’ containing a number of optional attributes that allow provide the code list properties necessary to fully identify the list used.



The ‘CodeType’ structure

## 4.2. Code lists defined and maintained within GS1

GS1 develops its own code lists only if no suitable international standard exists. Whenever possible, the international standard code lists are used.

#### Example

The ‘BarCodeTypeCode’ published in the BMS Shared Common Library.

## 4.3. Storing of code list values

Most code lists values are stored outside of the XML schemas. Thus, the user needs to look up the permissible values in the BMS documentation or the GS1 Global Data Dictionary ([GDD](#)).

This approach allows decoupling the code list management from the schema versioning – the values can be added between schema releases. It increases speed of implementation for messages where only a code addition is needed.

Code list values are enumerated in the schema only if all the following criteria are met:

- Values are “Intrinsic” to the message – essential to the processing of the message (e.g. type of invoice) and not reused by any other messages
- Code lists are immutable – not capable of or susceptible to change
- Code lists have manageable size – under 50 values

The code lists whose values are enumerated within the XML schema are referred to as “enumerations”.

Example

The “Day of the Week Enumeration” or “Document Status Enumeration” published in the BMS Shared Common Library.

## 4.4. Code list versioning

The code lists not enumerated in the schema can be updated when needed, which means that new values requested by the users are added as soon as they are approved.

Each revision will be identified with a sequential number, starting from 1, the next one will be 2, 3 and so on.

Example

GS1 Code List	TransportPaymentMethodCode
<b>GS1 Code List Version</b>	<b>1</b>
Managing Agency	UN/ECE
Based on Code List	UN/EDIFACT 4215
Type Of Management	GS1 Restricted / Extended

The code lists are versioned independently of the Business Message Standards, although the BMS publications may recommend use of the minimum code list version or higher version that is applicable to a specific release.

Example

[Code List] Version 3 or higher is recommended. Refer to the [Global Data Dictionary](#) or [Business Message Standard] for the list of values.

## 5. Communicating GS1 trade item numbers

The Global Trade Item Number – GTIN can have different number of characters, depending on the encodation used. Most bar code symbologies require a number shorter than the full 14-digit number. GS1 recommends that GTIN is always stored as a 14-digit number in the data bases. Shorter formats should be filled in with leading zeroes up to 14 characters – see below:

GTIN encodation	Global Trade Item Number (GTIN) in 14-digit format
ITF-14	N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14
GTIN-13	0 N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13
UPC-A	0 0 N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12
GTIN-8	0 0 0 0 0 0 N1 N2 N3 N4 N5 N6 N7 N8

In GS1 XML, the data type of GTIN is defined as 14 numeric characters ([see 6.2.1](#)). Thus, the full 14 digits must be transmitted. Shorter formats **MUST** be filled in with leading zeroes up to 14 digits.

- ✓ **Note:** Due to differences in syntax, the format of GTIN in both GS1 eCom standards: EANCOM® and GS1 XML are different.
  - In GS1 XML the format is defined as exactly 14 digits, which means that any GTIN that is originally shorter than 14 digits needs to be filled up to 14 characters by adding the leading zeroes
  - In GS1 EANCOM® the format is numeric, which means that the non-significant leading zeroes must be removed, since its data type is defined as numeric up to 14 digits (n..14). EDIFACT syntax does not allow transmitting non-significant leading zeros for numeric fields.

However, all these formats still convey the same GTIN identifier. GS1 recommends that GTIN should be stored in a 14-digit format. It can be transmitted in eCom messages or encoded in bar codes in a format that suites the syntax rules or bar code symbology.

## 6. XML syntax components used in GS1 XML 3.0

### 6.1. XML schema design model

GS1 uses the schema design model commonly referred to as Venetian Blind, where all the types are defined globally (nested only in the `<xsd:schema>` element) and all the elements and attributes are defined locally, within the type definitions.

### 6.2. XML built-in types

The XML standard developed by the World Wide Web Consortium (W3C) lists 44 data types that are [defined in the specification](#). This means that these data types can be used in XML schemas with no need to define them. These data types should be implicitly understood by all the XML aware software tools. The built-in data types have certain standard facets to represent them in the schema or to restrict their range.

The following W3C data types **ARE USED** in GS1 standards:

- |            |              |                      |
|------------|--------------|----------------------|
| ▪ string   | ▪ date       | ▪ base64Binary       |
| ▪ boolean  | ▪ gYearMonth | ▪ integer            |
| ▪ decimal  | ▪ gYear      | ▪ nonPositiveInteger |
| ▪ float    | ▪ gMonthDay  | ▪ negativeInteger    |
| ▪ duration | ▪ gDay       | ▪ nonNegativeInteger |
| ▪ dateTime | ▪ gMonth     | ▪ unsignedInt        |
| ▪ time     | ▪ hexBinary  | ▪ positiveInteger    |

The following W3C data types **ARE NOT USED** in GS1 standards:

- |                    |                 |                 |
|--------------------|-----------------|-----------------|
| ▪ normalizedString | ▪ ENTITIES      | ▪ unsignedLong  |
| ▪ token            | ▪ NMTOKEN       | ▪ unsignedShort |
| ▪ Name             | ▪ NMTOKENS long | ▪ unsignedByte  |
| ▪ NCName           | ▪ language      | ▪ double        |
| ▪ ID               | ▪ int           | ▪ anyURI        |
| ▪ IDREF            | ▪ short         | ▪ QName         |
| ▪ IDREFS           | ▪ byte          | ▪ NOTATION      |
| ▪ ENTITY           |                 |                 |

### 6.2.1. 'xsd:string' Pattern facet

The pattern (also referred to as regular expression) is one of the facets restricting the `<xsd:string>` built-in type. Patterns are used in GS1 XML schemas to define the [GS1 Identification Keys](#). The following patterns have been defined:

Key	Pattern	Description
GIAI	<code>[!%"&amp;'()*+,-/0-9:;&lt;=&gt;?A-Z_a-z]{4,30}</code>	Minimum 4, Maximum 30 alphanumeric characters restricted to A-Z (uppercase), a-z (lowercase), 0-9, and the special characters listed (e.g. %, &, etc..)
SSCC	<code>\d{18}</code>	Exactly 18 numeric digits
GTIN	<code>\d{14}</code>	Exactly 14 numeric digits
GLN	<code>\d{13}</code>	Exactly 13 numeric digits
GSIN	<code>\d{17}</code>	Exactly 17 numeric digits
GINC	<code>[!%"&amp;'()*+,-/0-9:;&lt;=&gt;?A-Z_a-z]{4,30}</code>	Minimum 4, Maximum 30 alphanumeric characters restricted to A-Z (uppercase), a-z (lowercase), 0-9, and the special characters listed (e.g. %, &, etc..)
GRAI	<code>\d{14}[!%"&amp;'()*+,-/0-9:;&lt;=&gt;?A-Z_a-z]{0,16}</code>	Exactly 14 numeric digits, followed by Minimum 0, Maximum 16 alphanumeric characters restricted to A-Z (uppercase), a-z (lowercase), 0-9, and the special characters listed (e.g. %, &, etc..)
GSRN	<code>\d{18}</code>	Exactly 18 numeric digits

### 6.2.2. Handling time stamps and time zones

For data elements where time is required, two built-in data-types are used within GS1 XML standards: `<xsd:dateTime>` and `<xsd:time>`. Their value space is defined in article 5.3 of ISO 8601, the details can be found at the following links:

<http://www.w3.org/TR/xmlschema-2/#time>

<http://www.w3.org/TR/xmlschema-2/#dateTime>

For the value space of `<xsd:dateTime>` and `<xsd:time>`, additional fractional seconds (milliseconds) and Time zone information can be specified.

#### 6.2.2.1. Use of milliseconds

Both `<xsd:dateTime>` and `<xsd:time>`, data types allow use of additional digits increasing the precision of fractional seconds if desired, in the format `ss.s`.

'ss.s' denotes two digits of a second (00 through 59) followed by one or more digits representing a decimal fraction of a second (milliseconds). The fractional second part is separated from the two digits of second by the use of a 'dot' as a separator.

Though any number of digits for the fractional seconds is supported, only 3 precision digits should be used within GS1 XML messages to denote the milliseconds.

##### Example

The following values are true for the attribute 'creationDateTime' which is of type `<xsd:dateTime>` in all versions of GS1 XML Standards.

- `<creationDateTime>2011-03-22T09:30:47</creationDateTime>`  
indicates 47 seconds and 0 milliseconds

- `<creationDateTime>2011-03-22T09:30:47.0</creationDateTime>`  
indicates 47 seconds and 0 milliseconds like in the example above
- `<creationDateTime>2011-03-22T09:30:47.233</creationDateTime>`  
indicates 47 seconds and 233 milliseconds

### 6.2.2.2. Use of time zones

Both `<xsd:dateTime>` and `<xsd:time>` data types allow specifying the time zone, following the time information. The time zone information for non-UTC<sup>1</sup> times within GS1 System must be expressed by specifying the difference between the local time and UTC. This is indicated by immediately following the time representation by a sign, + or -, followed by the difference from UTC represented as hh:mm. Please note that the minutes part is required.

#### Example

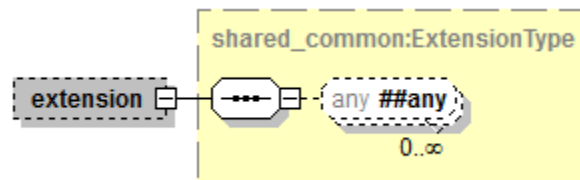
- `<creationDateTime>2011-11-22T12:43:17.000+09:00</creationDateTime>`  
indicates Tokyo Time, which is 9 hours ahead of UTC
- `<creationDateTime>2011-03-22T09:30:47.233-05:00</creationDateTime>`  
indicates Eastern Time Zone (EST), which is 5 hours behind UTC

where:

- '2011' denotes the year
- '03' denotes the month
- '22' denotes the day
- 'T' denotes the time separator
- '09' denotes the hours
- '30' denotes the minutes
- '47' denotes the seconds
- '.' denotes the fractional seconds separator
- '233' denotes the fractional seconds (milliseconds)
- '-' denotes the time zone offset indicator indicating 'behind UTC'
- '05' denotes the hours
- ':' denotes the minutes separator and '00' is the minutes

## 6.3. XML extension element

Every GS1 XML schema contains a placeholder where some additional, context-specific components can be inserted. This placeholder is defined as the optional "extension" element containing the unbounded number of the XML schema '##any' elements:



<sup>1</sup> UTC Stands for Coordinated Universal Time

Definition of the 'extension' element:

```
<xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
```

### 6.3.1. Definition of the extension content

The content defined in the extension schema can be added to the "core" message when the instance document is being created.

Although from technical point of view the extension can be applied to any extension point in any business document, the documentation strictly limits its use to one specific extension point in a specific message. This decision is based on the business requirement and the context which the extension will be used in.

The users decide whether to use it with the core message while creating the XML instance document. If they decide to use the extension, they must follow the usage rules specified in the BMS document.

#### Example

The ebMeat - Meat Item Containment Extension should be applied to the Despatch Advice line item extension point, as specified in the BMS.

### 6.3.2. Validation of the extension components

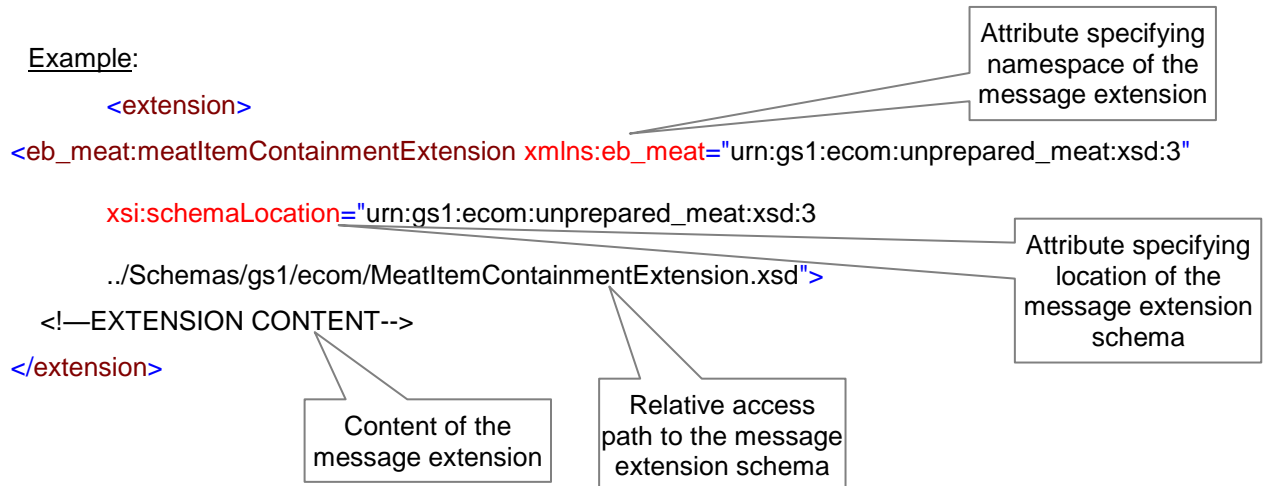
The XML 'any' element has a built-in attribute: 'processContents'. It is an indicator for an application or XML processor how to handle the validation of XML documents against the elements specified by the 'any' element. In the GS1 extension component, the value of this attribute is set to 'lax', it means that the XML processor attempts to obtain the schema for the required namespaces and validate any element from those namespaces, but if the schema cannot be obtained, no errors will occur.

Although the XML syntax rules for the 'any' element allow any well-formed XML to be placed within the instance file, the users must remember that their business partners will not be able to understand the non-standard information. It is recommended to use only the extension attributes approved within the [GSMP](#) process. Otherwise the messages will not be globally interoperable. However, because the 'processContents' attribute is set to 'lax', the validation of the extension element is optional: an XML schema parser will validate elements for which it can find declarations (extension schemas) and raise errors if they are invalid. If it does not find context specific schemas for a certain set of context specific elements, the parser will not report errors for those elements.

### 6.3.3. Applying an extension to the instance file

In the instance file, the content of the extension should be placed within the 'extension' element and replace the '##any' placeholder element. The first element (root) of the extension must contain two additional attributes:

- 'xmlns' specifying the namespace of the extension schema
- 'xsi:schemaLocation' specifying the access path of the extension schema

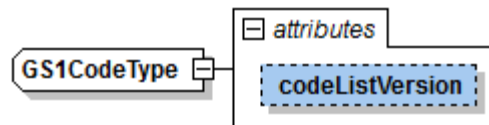


## 6.4. Use of XML elements vs. attributes

In the GS1 XML MR 3 attributes are used only to define data type supplementary components.

Example:

The 'codeListVersion' is an optional attribute of the 'GS1CodeType':



The XML elements are used to define all other constructs.

## 7. GS1 namespaces

### 7.1. GS1 namespace structure

The namespaces in GS1 reflect the domain, the interchange which the schema is used in and the [major version](#) of the schema release. The GS1 namespaces have the format of the Uniform Resource Names (URN). All URNs have the following structure:

`<URN>= "urn:" <NID> ":" <NSS>`

where:

`<NID>` is the Namespace Identifier

`<NSS>` is the Namespace Specific String

In all the GS1 namespaces, "gs1" is used for the NID. The NSS of all URNs assigned by GS1 have the following hierarchical structure:

`urn:gs1:<domain>:<interchange>:<type>:<version-id>`

where:

- domain – GS1 Business Area, Service, Technology, e.g. eCom
- interchange – Artefact within the GS1 Domain, e.g. Order
- type – Resource type, e.g. xsd
- version-id – Version of the Interchange, e.g. 3



#### Examples:

- urn:gs1:ecom:ecom\_common:xsd:3 - the GS1 namespace for eCom common library
- urn:gs1:shared:shared\_common:xsd:3 - the GS1 namespace for shared common library
- urn:gs1:ecom:order:xsd:3 - the GS1 namespace for eCom Order

- ✔ **Note:** NSS in GS1 namespaces is case-sensitive, even though in the RFC 2141 is specified as not sensitive.

## 7.2. GS1 namespace prefix

GS1 XML schemas do not use any default namespace. The prefix should be explicitly specified for each namespace. The GS1 standard specifies the standard namespace prefixes that are created from the name of the schema file.

#### Examples:

- "order" for Order.xsd:  
**order**="urn:gs1:ecom:order:xsd:3"
- "invoice" for Invoice.xsd:  
**invoice**="urn:gs1:ecom:invoice:xsd:3"
- "despatch\_advice" for DespatchAdvice.xsd:  
**despatch\_advice**="urn:gs1:ecom:despatch\_advice:xsd:3"
- "ecom\_common" for EcomCommon.xsd:  
**ecom\_common**="urn:gs1:ecom:ecom\_common:xsd:3"
- "shared\_common" for SharedCommon.xsd:  
**shared\_common**="urn:gs1:shared\_common:xsd:3"

For the sake of consistency and interoperability in GS1 System it is recommended that for the XML instance documents the implementers use the same namespace prefixes as those used in the standard GS1 XML Schemas.

Although, from the XML syntax point of view, any prefix can be used as long as it points to the correct namespace, assigning non-standard prefixes would compromise the global interoperability of standards.

In addition, certain mapping and processing XML tools are prefix sensitive and use of non-matching prefixes by the business partners causes serious validation and processing problems.

## 7.3. XML schema namespace and prefix

For the XML schema components, the World Wide Web Consortium (W3C) URL <http://www.w3.org/2001/XMLSchema> is used as the namespace. Again, the prefix for that namespace has been standardised in the GS1 System and all the existing schema modules use **xsd:** for the XML Schema namespace.

## 7.4. Qualifying of GS1 XML schema components

In the GS1 XML schemas the values of the `<xsd:schema>` "form" attributes are set to "unqualified" both for 'elementFormDefault' and 'attributeFormDefault'. Thus, the namespace prefixes are not exposed in the instance files, except for the root element of the business message and extension – if used.

The SBDH schemas, however, set the value of the 'elementFormDefault' to "qualified" and 'attributeFormDefault' to "unqualified". Thus, all the SBDH elements need to start with the namespace prefix: 'sh:'.

Example:

- GS1 XML schema:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified" attributeFormDefault="unqualified">
```

definition of unqualifying elements and attributes

- GS1 XML instance:

```
<order:orderMessage xmlns:order="urn:gs1:ecom:order:xsd:3"
  xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:gs1:ecom:order:xsd:3 ../Schemas/gs1/ecom/Order.xsd">
```

Qualified root element

```
<sh:StandardBusinessDocumentHeader>
```

Qualified SBDH element

```
<!--SBDH CONTENT-->
```

```
</sh:StandardBusinessDocumentHeader>
```

```
<order>
```

Unqualified elements of business document

```
<!--ORDER DOCUMENT CONTENT-->
```

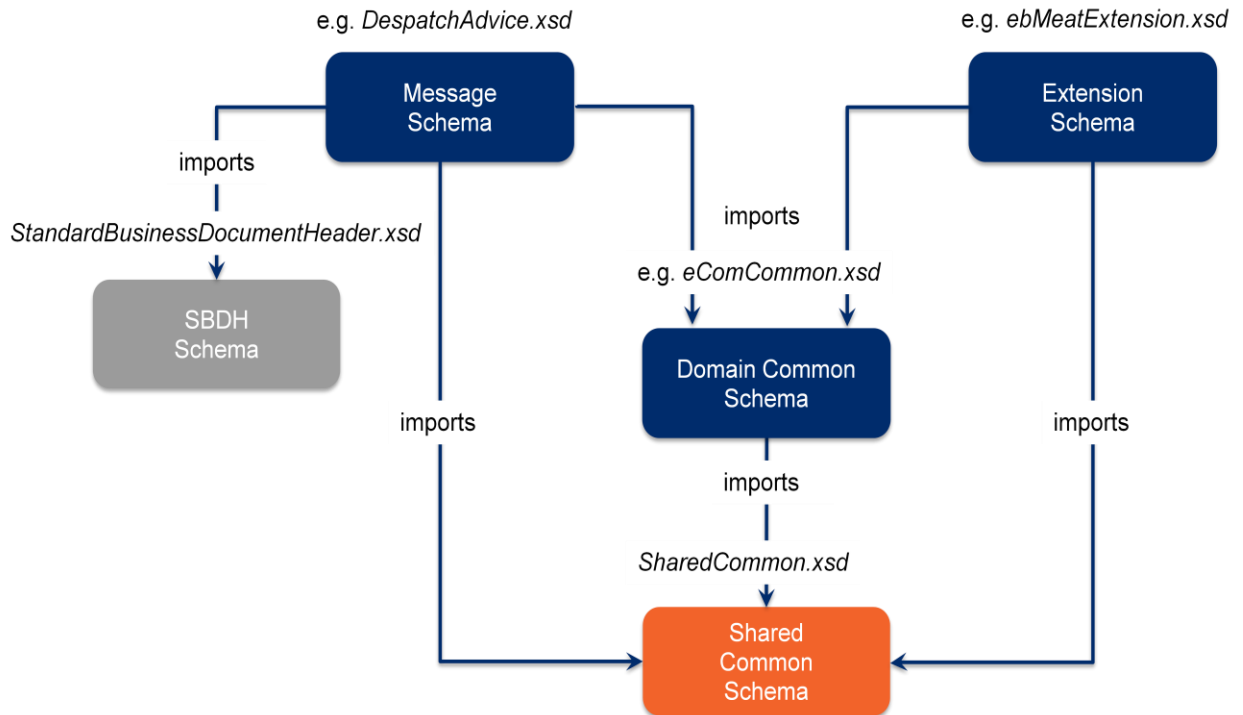
```
</order>
```

```
</order:orderMessage>
```

## 7.5. GS1 namespace use

In GS1 XML 3.0 every schema document has a separate namespace. The namespace are no longer shared between multiple schema files. Thus, when re-using the schema artefacts defined in another schema file, the `<xsd:import>` element is used.

## 8. GS1 XML schema hierarchy



In the GS1 XML release 3.0 the business message components are defined in the XML schemas of three levels:

- Shared Common schema – contains all the components that can be re-used across domains, e.g. eCom and GDSN, such as GS1 Identification Keys, Party Identification, etc.
- Domain Common – contains all the components that can be re-used within one domain, e.g. eCom, such as Transactional Party Identification, Order Type Code. The domain specific components are defined using the shared components when available. Thus, the *eComCommon.xsd* needs to import the *SharedCommon.xsd*
- Message and extension schema – defines the structure of the actual message and the business document, as well as the content of the potential extensions. The document and extension specific components are defined using the shared and domain specific components when available. Thus, the *BusinessMessage.xsd* (e.g. *Order.xsd*, *Invoice.xsd*) needs to import the *SharedCommon.xsd* and *eComCommon.xsd*

The [Standard Business Document Header](#) component is defined in its own set of schemas, while the structure of the header is defined in the *StandardBusinessDocumentHeader.xsd* that needs to be imported into the *BusinessMessage.xsd*.

### 8.1. Installation of GS1 XML schemas

GS1 business message schemas re-use components from the common library, thus it must import the respective schemas. The `<xsd:import>` element contains a built-in attribute 'schemaLocation' that specifies the access path to the imported schema. If the re-used schema is saved in another place than the value of the location attribute has to be updated accordingly. Otherwise the validating tool cannot find the imported schema and declares the file (either schema itself or the instance file) as not valid. Therefore, it is crucial either to install the schemas as provided by GS1 in the Implementer's Packet or – if this is not possible - edit the access path accordingly.

### Example

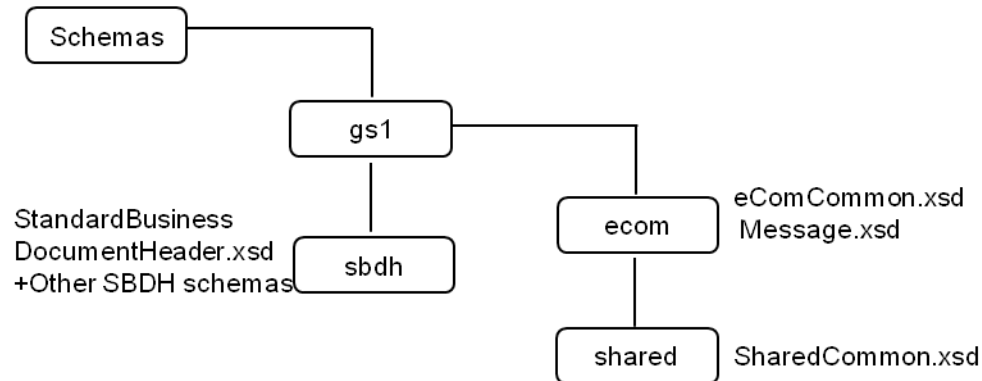
Every business message schema imports at least the following three schemas:

```
<xsd:import namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
schemaLocation=" ../sbdh/StandardBusinessDocumentHeader.xsd" />
```

```
<xsd:import namespace="urn:gs1: shared:shared_common:xsd:3"
schemaLocation=" ../shared/SharedCommon.xsd" />
```

```
<xsd:import namespace="urn:gs1:ecom:ecom_common:xsd:3"
schemaLocation=" eComCommon.xsd" />
```

These paths reflect the following folder structure:



- ✓ **Note:** The file names of the GS1 XML schemas do not contain version numbers, therefore special care must be taken if multiple versions of business messages are used in the same system. For more details about versioning please refer to [GS1 XML versioning](#)

## 9. GS1 XML message architecture

In GS1 MR3 the message architecture for the two main domains: eCom and GDSN<sup>2</sup> has been decoupled following requirements of the user community.

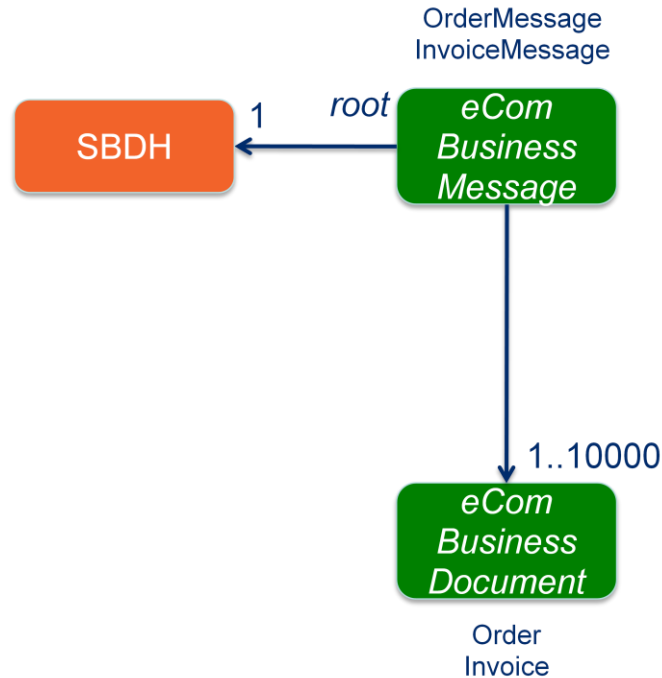
### 9.1. Overview of eCom Message Architecture

The main principle of the GS1 eCom message architecture is separation of two functional components:

- Transport and routing component, containing information about routing and processing of the XML instance document
- Business document component, containing the actual business document or documents, e.g. Order or Invoice. Each message can contain up to 10,000 documents of one type (e.g. ONLY Orders or Invoices).

Both components combined constitute a GS1 XML message, e.g. Order Message or Invoice Message.

<sup>2</sup> The architecture currently used by the GDSN is described in the [GS1 XML Message Architecture Implementation Guide](#)

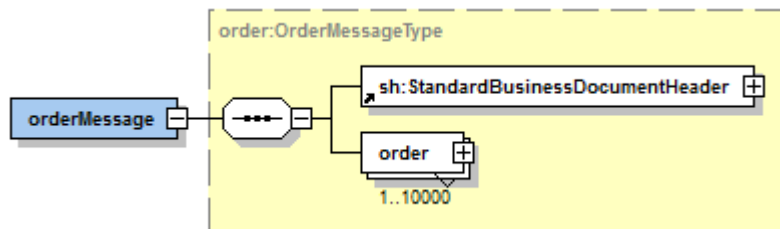


The structure of the eCom message architecture

## 9.2. Benefits

The separation of transport and routing information from the business document allows delivering the message to the right addressee and the correct business application without accessing the content of the business document. The later may be either impractical or impossible, if the document content is encrypted for security reasons.

Allowing for only one document type in the message ensures the correct routing within the company. For example, all the business documents are passed to the financial or ordering systems.



The structure of the Order Message

## 9.3. Overview of eCom message components

### 9.3.1. Transport and routing component

GS1 System uses the **Standard Business Document Header (SBDH)** for routing and processing information in the GS1 XML standard.

The specifications for SBDH have been developed by the UN/CEFACT, the document is available at the GS1 website: [www.gs1.org/services/gsmf/kc/ecom/xml/xml\\_sbdh.html](http://www.gs1.org/services/gsmf/kc/ecom/xml/xml_sbdh.html).

The SBDH has been designed to be independent of the specific transport protocol used. Information contained in the SBDH can be used by communication applications to determine routing, whether the transport protocol used is ebMS, AS2, or any other protocol.

Using one common document header enables integration of documents between internal applications, enterprise applications, and business-to-business infrastructure by providing a consistent interface between applications. It allows any application to determine the logical routing requirements and/or the logical processing requirements of a document based on the information contained in the SBDH.

SBDH contains information for communication applications about:

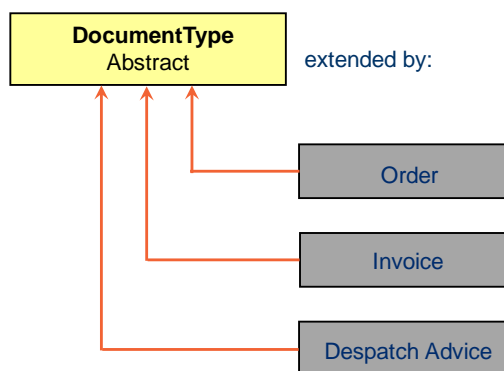
- **Document Routing** – identifies message sender and receiver; used by message exchange hubs, marketplaces, etc.
- **Document Identification** – used by the middleware to identify and route the message to the appropriate business application without opening or parsing it
- **Document Processing Context** – provides parameters for processing the business document in the context of a business choreography exchange and identifies specific guidelines document that has been used as a base for creating the instance document
- **Payload** – a container for the business document

Detailed guidelines how to implement the SBDH with the GS1 XML standard have been documented separately in the ‘Standard Business Document Header (SBDH) Version 1.3 Technical Implementation Guide’, available at the GS1 website: [www.gs1.org/services/gsm/kc/ecom/xml/xml\\_sbdh.html](http://www.gs1.org/services/gsm/kc/ecom/xml/xml_sbdh.html).

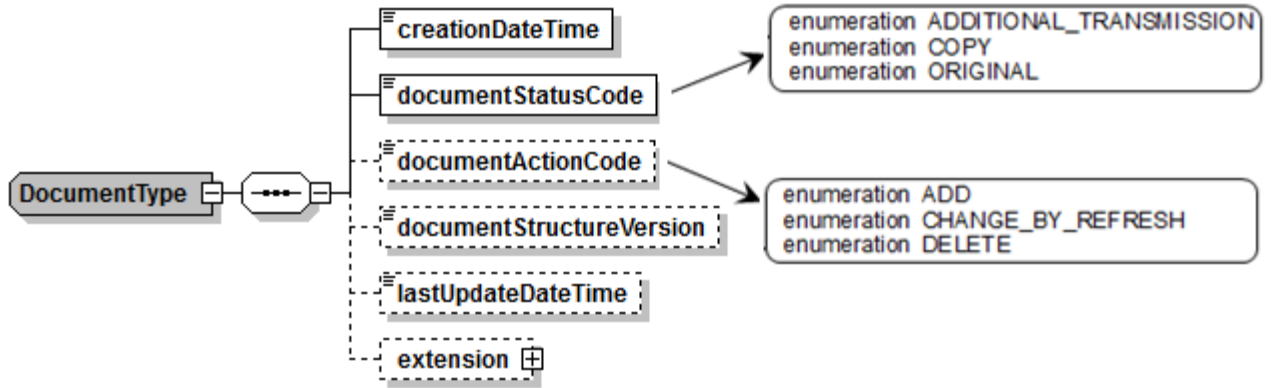
### 9.3.2. Business document component

The Document contains the business information components, specific for the type of document, i.e. Order, Despatch Advice or Invoice. These documents are fully documented in GS1 Business Message Standards and will not be discussed in this guide. All the documents, however, need to contain certain common information such as document identification, document date, etc.

All the business document types are based on the common ‘DocumentType’ and inherit all the necessary components. This ‘DocumentType’ is defined as abstract, which means that this structure cannot be used in the instance file directly, but it must be extended by a concrete type, such as ‘OrderType’, ‘InvoiceType’, etc.



The abstract ‘DocumentType’ has the following content:

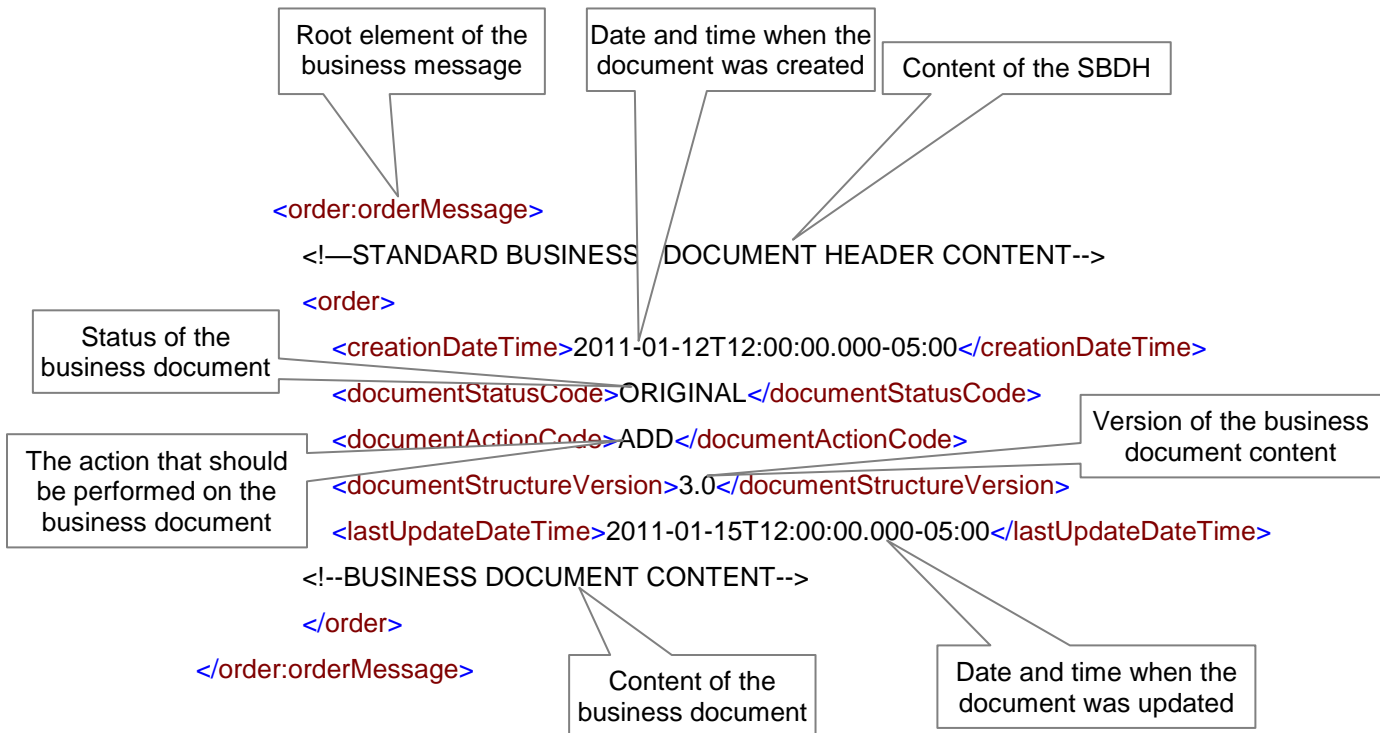


## Document component tags

XSD Element / Attribute	XSD Type	Occ	Usage Guidelines
	shared_common:DocumentType		This is an abstract type that cannot be directly instantiated, but must be extended by the concrete business message type (e.g. orderMessageType, InvoiceMessageType, etc.). The list of possible documents contains all the messages defined within the given major release.
creationDateTime	xsd:dateTime	1..1	An element used to specify the date and time when the document was created.
documentStatusCode	shared_common:DocumentStatusEnumerationType	1..1	This element contains an enumeration list of the three possible status values of the document.
documentActionCode	shared_common:DocumentActionEnumerationType	0..1	This element contains an enumeration list of the three possible action values that can be performed on the business document.
documentStructureVersion	xsd:string	0..1	Reflects the complete (major and minor) version of the business message schema. E.g. for the Order.xml message, the document structure version reflects the version number of the Order.xsd schema.
lastUpdateDateTime	xsd:dateTime	0..1	Specifies the date when the document was last updated.
extension	shared_common:ExtensionType	0..1	See: <a href="#">XML extension element</a>



### Document component instance example



## 9.4. Batching of business documents

The GS1 message architecture enables batching of business documents – transmitting them as a part of one message – up to 10,000 documents per one message. All documents have to be of the same type, e.g. only Orders or only Invoices. Mixing various types of documents is not allowed and not technically possible with the current design.



Below is an example of an XML instance containing one SBDH and three Business Documents – in this case the Orders:

```
<order:orderMessage>
  <sh:StandardBusinessDocumentHeader>
    <!-- STANDARD BUSINESS DOCUMENT HEADER CONTENT-->
  </sh:StandardBusinessDocumentHeader>
  <order>
    <!--FIRST ORDER DOCUMENT CONTENT-->
  </order>
  <order>
    <!--SECOND ORDER DOCUMENT CONTENT-->
  </order>
  <order>
    <!--THIRD ORDER DOCUMENT CONTENT-->
  </order>
</order:orderMessage>
```

## 10. Main differences between the GS1 XML 3.0 and 2.0

The review of the GS1 XML design has been done based on feedback from user community who experienced difficulties in implementation of MR2 and requested significant changes.

### 10.1. Reasons for change

The design of the previous Major Release 2 was very flexible, but at the same time difficult to implement and understand. Over the years user requirements changed demanding more simplicity in place of flexibility that turned out to be unnecessary. The main reasons for changing the previous GS1 XML design were:

- Reduce overhead when used in eCom messaging: Transaction and Batching were not used in eCom exchange
- Improve tool support (parsers, validators and mapping tools)
- Make the design easier to understand and implement:
  - Provide single 'root' schema for validation
  - Clearly define what types of documents are allowed in one message
- Simplify maintenance of code lists and thus accelerate implementation

## 10.2. Main changes in the GS1 XML design

FUNCTIONALITY	MR3	MR2
<b>SBDH</b>	Mandatory	Optional
<b>Transaction</b>	Not used	Used, optional
<b>Commands</b>	Not used, command instructions moved to <a href="#">Document</a> component	Used, optional
<b>Root schema</b>	One	Many – no clear ‘root’
<b>Document types allowed in one message</b>	One document type	Various document types
<b>Number of documents allowed in one message</b>	Up to 10.000	Any
<b>Proxy schemas</b>	Not used	Used, optional
<b>Namespaces</b>	Different for every schema file	One for multiple schema files
<b>Context categories</b>	Not used	Used in namespaces
<b>Actions</b>	Enumerated in the <a href="#">Document</a> component, value CORRECT removed from eCom list	Enumerated in the Command layer
<b>Code list values</b>	Most (those that are likely to evolve) stored outside of the schema	Enumerated in the schema
<b>Common library</b>	Three library schemas: shared and domain common (separate ones for eCom and GDSN)	Library of multiple schemas separately defining common components
<b>Use of elements vs. attributes</b>	Attributes used only for data types, elements for everything else	No specific rules, except for use of attributes for metadata

- ✔ **Note:** The table above does not cover the changes made in the messages, based on the user requirements. For that please refer to the respective BMS, section “Summary of Changes”.