

CLASSIFICATION FOR TB.CAT 1.3



CLASSIFICATION FOR TB.CAT 1.3

© 2026 Tradebyte Software GmbH

All rights reserved. This document contains proprietary information of Tradebyte and may not be disclosed or used except in accordance with applicable agreements. This material is protected by copyright laws. This material may not be reproduced, distributed, or altered in any manner by any entity without the written consent of Tradebyte and the owner of this material, unless in accordance with applicable agreements, contracts or licenses.

For permission to reproduce or distribute, please contact:
Tradebyte Software GmbH, Bahnhofplatz 8, D-91522 Ansbach
E-mail: Support@Tradebyte.com
<https://www.tradebyte.com>

It was made every effort to ensure that the information contained in this document is complete and correct at the time of publication. The right to change the information remains reserved.

This customer document describes all the functions known today. It is possible that some functions are described that are not available to the customer. The exact functionality is dependent on the selected edition and the current release.

Trademarks

Microsoft, Windows and Excel are registered trademarks of Microsoft Inc., USA.

Portable Document Format (PDF) and PostScript are trademarks of Adobe Systems, Inc.

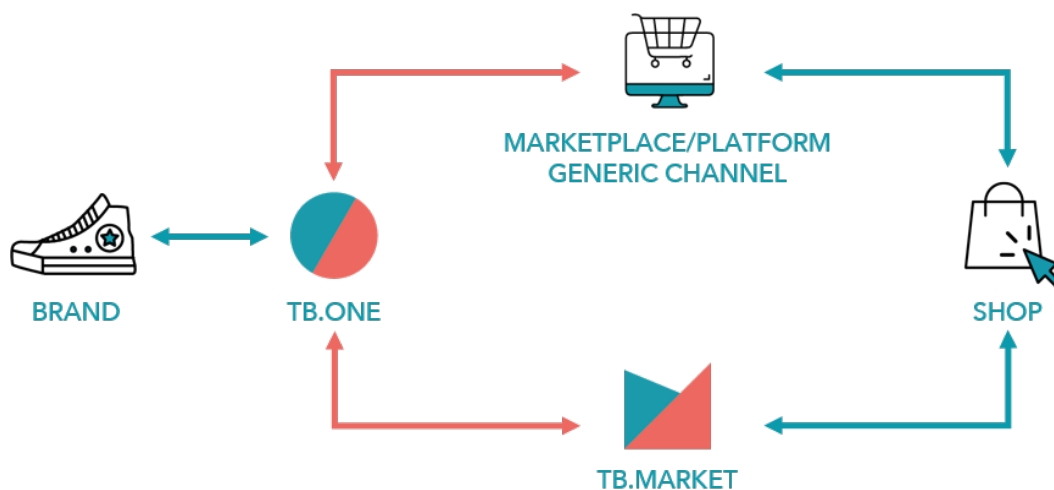
All other company names and logos and brand or product names are trademarks or registered trademarks of their respective owners.

TABLE OF CONTENTS

1. WHAT YOU CAN EXPECT IN THIS MANUAL	5
2. DEFINITIONS AND GENERAL INFORMATION	6
2.1 CHANNEL	6
2.2 STRUCTURE: CLASSIFICATION AND STRUCTURE_ITEMS	6
2.3 CONVENTIONS	6
3. DATA EXCHANGE VIA REST	8
3.1 ACCESSING THE REST INTERFACE	8
3.2 MANDATORY INFORMATION	8
3.3 RETURNED VALUES	9
3.4 CLASSIFICATION PUSH	9
4. REFERENCING DATA	10
4.1 KEY-BASED	10
4.2 NAME-BASED	10
5. STRUCTURE OF THE TB.CAT CLASSIFICATION	11
5.1 TBCATALOG / CLASSIFICATION	11
TBCATALOG / CLASSIFICATION / STRUCTURE	13
TBCATALOG / CLASSIFICATION / STRUCTURE / STRUCTURE_ITEM	15
TBCATALOG / CLASSIFICATION / COMPONENTS	18
TBCATALOG / ... / COMPONENT / DATA_TYPE	20
TBCATALOG / ... / COMPONENT / COMPONENT_LISTS	30
TBCATALOG / CLASSIFICATION / TAGS	32
TBCATALOG / CLASSIFICATION / BRANDS	34
TBCATALOG / CLASSIFICATION / ATTRIBUTE_SETS	35
TBCATALOG / ... / ATTRIBUTE_SETS / ATTRIBUTE_SET_ITEM	37
TBCATALOG / CLASSIFICATION / LANGUAGES	46
TBCATALOG / CLASSIFICATION / SHIPPING_GROUPS	47
TBCATALOG / CLASSIFICATION / GRIDS	48
TBCATALOG / CLASSIFICATION / MEDIA_TYPES	50
6. EXAMPLE CLASSIFICATION	51

1. WHAT YOU CAN EXPECT IN THIS MANUAL

This document is for retailers who operate a marketplace, a platform or use TB.Market. It describes how you can transfer your categories and data requirements to connected brands that use TB.One.



2. DEFINITIONS AND GENERAL INFORMATION

2.1 CHANNEL

Be they market places, platforms or generic channels (e.g. a manufacturer's proprietary shop): this document refers to all retailers as channels. Tradebyte will assign your channel a channel ID and a channel key which you need to identify yourself when communicating with a TB.One. Your Tradebyte contact will provide these credentials to you.

2.2 STRUCTURE: CLASSIFICATION AND STRUCTURE_ITEMS

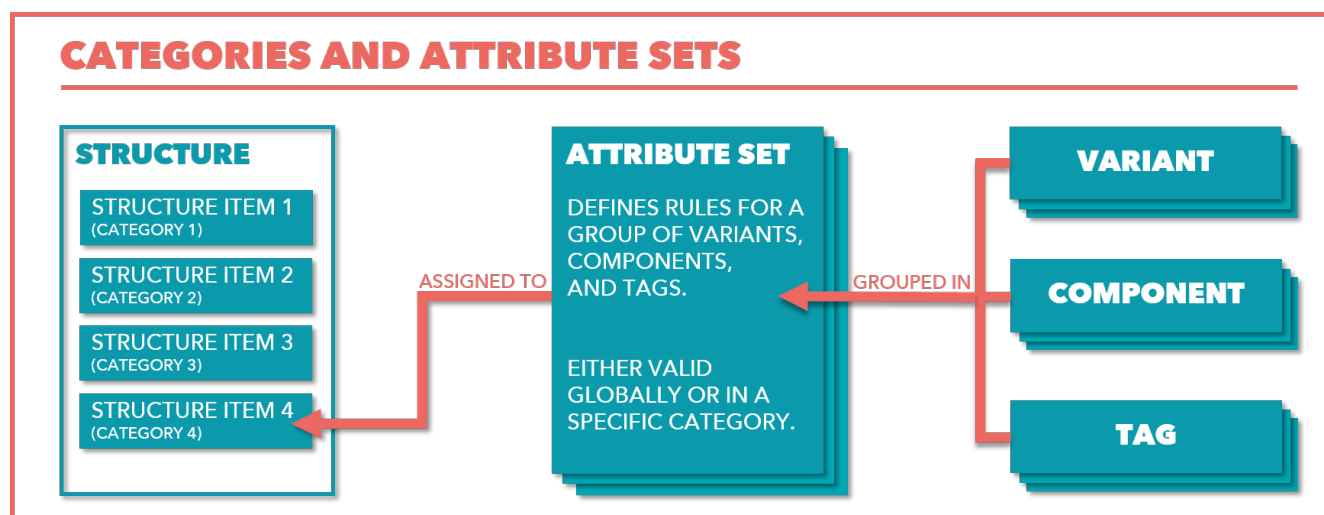
You need to define a **structure (category map)** for your channel so TB.One systems can connect to it. The data providers will then assign their products/articles to individual categories within that structure.

The structure is part of the so-called "Classification". Apart from the category structure, the Classification may also define mandatory information TB.One systems need to provide to you along with the form in which this information is to be provided, such as **brand names**, **components** or **tags**. Please note that this terminology is Tradebyte-specific. Your company may use the same or similar terms for different things.

Let us assume you want to define some mandatory information your partners (the connected TB.One client systems) need to provide. In this case, you need to define required data fields (components, tags, etc.) and define rules for them via **attribute sets**. These attribute sets (and their requirements) may be valid across all categories or category-specific.

Generally, you would provide your Classification to the TB.One clients or TB.Markets in an XML file via a REST interface.

The chart below (Categories and Attribute Sets) illustrates how the category structure, attribute sets, components/variants, and tags interact:

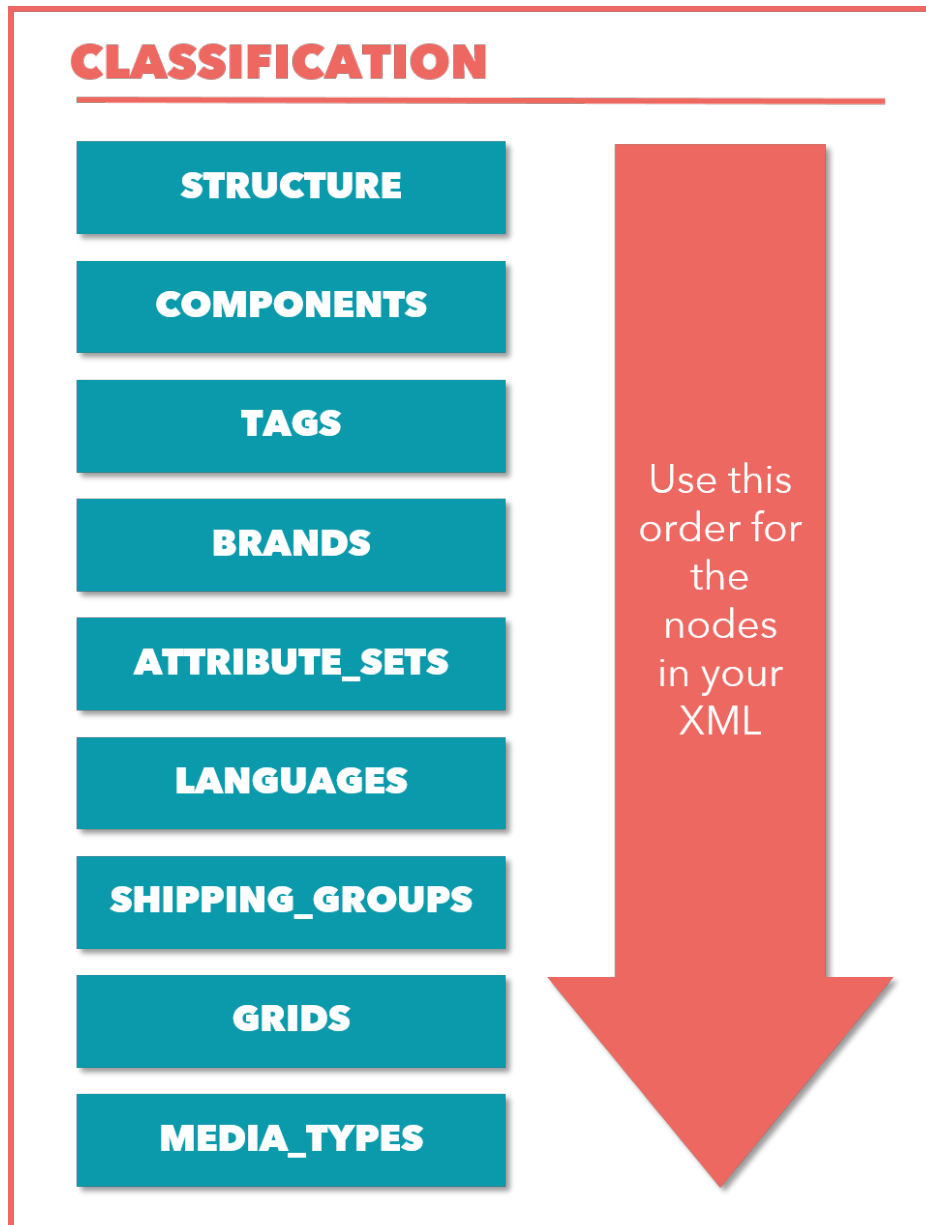


2.3 CONVENTIONS

This manual does not describe all nodes your Classification may technically contain (see the link to the XSD file in the next chapter). However, generally speaking, **any nodes not described here should not be part of your Classification XML**. They may be needed for internal communication between Tradebyte systems (in which case our systems will add them themselves).

In edge cases where you do need to include a node not listed in this manual, your Tradebyte contact will notify you.

The Classification XML should be structured as follows:



3. DATA EXCHANGE VIA REST

Data are transferred between your channel and TB.One/TB.Market via XML files. This data exchange takes place via a REST interface. All data you send to or receive from a TB.One must be validated against the current TB.CAT data schema.

Validate each file you dispatch or receive against the current schema to ensure you keep up with possible updates.



You can find the XML schema on our server under the following URL:
https://api.core.tradebyte.com/schema/all_in_one/tb-cat_1_3_all_in_one.xsd

3.1 ACCESSING THE REST INTERFACE

With the REST interface, you can send your Classification directly to TB.One/TB.Market systems. It becomes valid instantly after a successful PUT. Please note: if you define new requirements/definitions in the Classification (e.g. a new mandatory field), some articles may no longer be exported to your system as long as the data provider does not comply with the new requirements yet.



General structure of the REST URI:
https://rest.core.tradebyte.com/ACCOUNT-NR/TARGET/?channel=CHANNEL_ID

To access the REST interface, you need to identify yourself with your user name and password.

3.2 MANDATORY INFORMATION

PLACEHOLDER	DESCRIPTION
USR/PWD	User name and password for the REST access.
ACCOUNT-NR	Number of the TB.One account for which the Classification will become valid. Your Tradebyte contact will provide this number/these numbers to you when you set up the channel.
TARGET	Defines the resource group which is the target of this call. In this case, the target is "products".
CHANNEL_ID	Your Tradebyte contact provide your channel ID. In the examples, the ID "CHID" is used.

3.3 RETURNED VALUES

Generally, you will receive a return message in the form of an XML for each REST call. You will either receive the requested result or a status message (indicated by a status code in the HTTP header) which signals that your file was processed successfully or unsuccessfully.

The following HTTP status codes may be returned:

- 200 (all codes starting with 2): call/processing successful.
- 400 (all codes starting with 4): call unsuccessful/request failed.
- 500 (all codes starting with 5): processing failed.

3.4 CLASSIFICATION PUSH

METHOD	PUT	Sends...
TARGET	products	... a file that is valid for all products ...
ACTION	classification	... in the form of a TB.CAT XML classification which is valid for the indicated channel. During product creation it is used for the primary categorization. It defines mandatory categories and values, to which the suppliers have to map their own values.
CALLER	channel	
EXAMPLE	...ACCOUNT-NR/products/classification?channel=CHID For an example of the XML file itself, see chapter 6 .	
RESPONSE	200 OK	

4. REFERENCING DATA

The files you transfer can use different ways of referencing data. This choice will impact how exchanged information is to be interpreted.



The referencing information is transferred in so-called XML attributes. In the list of XML nodes, the attributes are displayed in [square brackets]. In all other texts, they are in *italics*.

Currently, there are two methods:

4.1 KEY-BASED

The transferred information is referenced via a predefined key. You can recognize this by the attribute "identifier=key". The key itself will be indicated in the attribute "key=[defined key]". If the field's content is relevant for the transferred piece of information, it is provided as an XML value.

Use this way of referencing, when you will interpret and assign the transferred information precisely.

```
<ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1" overridePosition="article">  
  <COMPONENT identifier="key" key="FarbNr" name="FarbNr. Lieferant"/>  
</ATTRIBUTE_SET_ITEM>
```

4.2 NAME-BASED

The transferred information is referenced via a name (designation) that is chosen by the sender. You can recognize this by the attribute "identifier=name". The name/designation itself is indicated with the attribute "key=[name/designation]". If the field's content is relevant for the transferred information, it is provided as an XML value.

Choose this way of referencing when you can assign the transmitted information by its context (component/attribute/brand), but not every piece of information can/should be interpreted precisely.

```
<ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1" overridePosition="article">  
  <COMPONENT identifier="name" key="FarbNr. Lieferant"/>  
</ATTRIBUTE_SET_ITEM>
```

5. STRUCTURE OF THE TB.CAT CLASSIFICATION



A note on the graphical representation of the XML structure in the following chapters' diagrams:

Solid lines denote mandatory fields, while dotted lines denote optional fields.

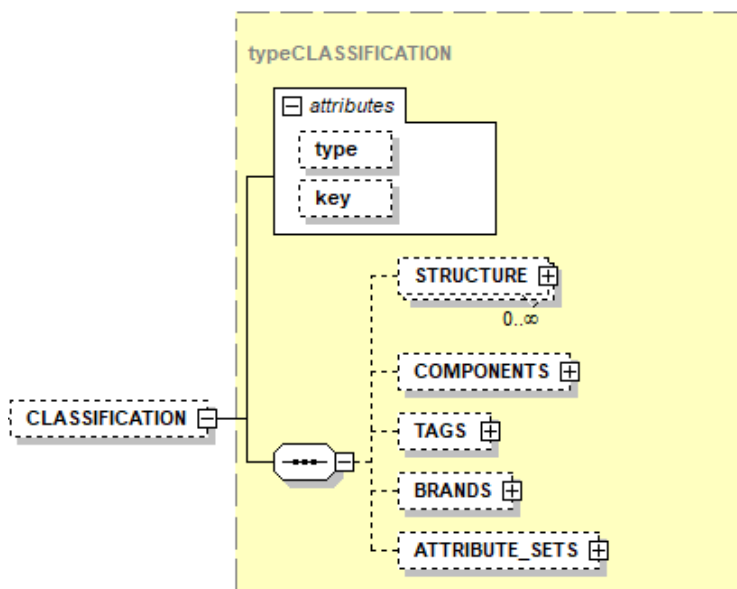
For child-level nodes, this information only relates to the context of the respective parent node, not to the entire structure. That means if a parent node is marked as optional, but one of its children is marked as mandatory, that child node is only mandatory if its parent is included in the file. If a child node and all its ancestor/parent nodes are marked as mandatory, you have to include them all in the file.

5.1 TBCATALOG / CLASSIFICATION

This section of the TB.CAT contains lists of categories and brands. Use it to define values (components, tags) that are valid or even mandatory for your channel.



CLASSIFICATION is a child node of TBCATALOG. To find out, how a complete Classification XML looks like, see the [example](#) at the end of this manual.



NODE	REQ.	TYPE	DESCRIPTION
CLASSIFICATION[type]	No	ENUM	The Classification feed's type. Currently there is only one possible value: "replace". For feeds with the type "replace", values that are not included in the file are deleted.

NODE	REQ.	TYPE	DESCRIPTION
CLASSIFICATION[key]	No	ENUM	This Classification's key. You can freely define this key, but it has to be unique. It should never change.
STRUCTURE	No	XML	Defines your category structure.
COMPONENTS	No	XML	Properties which are usually shown to the end customer in the shop.
TAGS	No	XML	Properties which usually contain meta information that is not displayed in the shop directly, but may enhance the shop's performance (filter information, etc).
BRANDS	No	XML	A list of the brands your channel sells.
ATTRIBUTE_SETS	No	XML	Definition of mandatory values.
LANGUAGES	No	XML	Define mandatory and optional languages for your shop.
SHIPPING_GROUPS	No	XML	A list of possible carriers for shipping.
GRIDS	No	XML	You can use this node to define length/size grids.
MEDIA_TYPES	No	XML	Here, you can define media types for your shop (model image, frontal view, etc).

EXAMPLE:

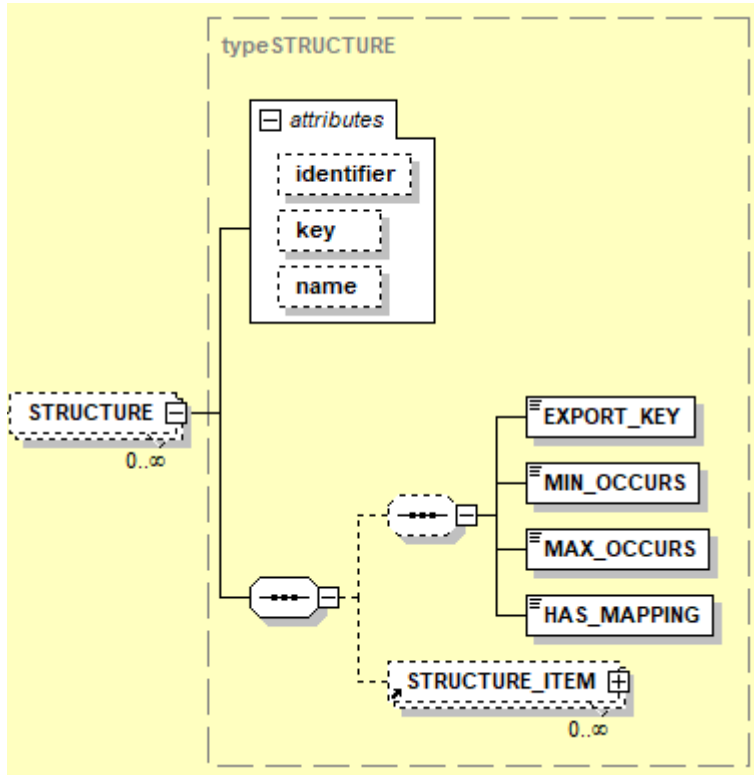
```

<CLASSIFICATION>
  <STRUCTURE/>
  <COMPONENTS/>
  <TAGS/>
  <BRANDS/>
  <ATTRIBUTE_SETS/>
</CLASSIFICATION>

```

TBCATALOG / CLASSIFICATION / STRUCTURE

This section is used to define a category tree, to which your connected brand/sales partners will assign their products. Your STRUCTURE node has to contain at least one STRUCTURE_ITEM.



NODE	REQ.	TYPE	DESCRIPTION
STRUCTURE[identifier]	No	ENUM	Indicator for the type of referencing used.
STRUCTURE[key]	Yes	CHAR (100)	Value for the referencing type provided in <i>identifier</i> .
STRUCTURE[name]	Yes	CHAR (300)	Name for the referencing type provided in <i>identifier</i> .
MIN_OCCURS	Yes	INT	"0" = not mandatory. "1" = the brand/sales partner has to provide at least one value (e.g. the primary channel category).
MAX_OCCURS	Yes	STRING	"1" = one value may be assigned at maximum. "unbounded" = the brand/sales partner can indicate an unlimited number of assignments for one product (e.g. by assigning it to multiple shop categories).

NODE	REQ.	TYPE	DESCRIPTION
HAS_MAPPING	Yes	STRING	Use the value "first", which means the STRUCTURE is used as the primary category tree.
STRUCTURE_ITEM	No	XML	Individual category node.

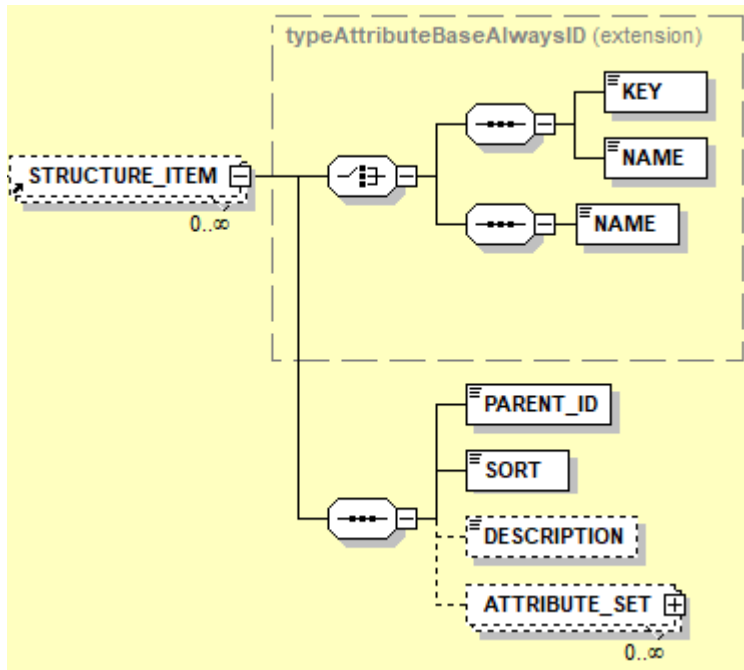
EXAMPLE:

```

<STRUCTURE identifier="key" key="category_tree" name="Shop Categories">
  <EXPORT_KEY>cat_tree_1</EXPORT_KEY>
  <MIN_OCCURS>1</MIN_OCCURS>
  <MAX_OCCURS>1</MAX_OCCURS>
  <HAS_MAPPING>first</HAS_MAPPING>
  ...
</STRUCTURE>

```

TBCATALOG / CLASSIFICATION / STRUCTURE / STRUCTURE_ITEM



Provide a flat category structure in the Classification. All categories should exist on the same technical level (no hierarchy in the XML-file). Indicate the entire category path for each category you use in its respective `NAME` node. We recommend separating the individual levels of the category path with the pipe symbol ("|").

The category name for your women's sneakers might look like this: "Women | Shoes | Sneakers".

NODE	REQ.	TYPE	DESCRIPTION
STRUCTURE_ITEM	No	XML	Individual category node.
PARENT_ID	Yes	INT	Indicates the ID of the parent category. Since the Classification should contain a flat category structure, always use "0" for this node.
SORT	Yes	INT	Defines the order of the categories. Categories without SORT values are ordered alphabetically in TB.One.
DESCRIPTION	No	TEXT	A text describing the category node (e.g. examples showing which products belong in this category).

NODE	REQ.	TYPE	DESCRIPTION
ATTRIBUTE_SET	No	XML	Used to assign mandatory fields to the category. Those mandatory fields are defined in the node ATTRIBUTE_SETS .
KEY-BASED IMPORT			
KEY	Yes	CHAR (100)	Usually identical to NAME.
NAME	Yes	CHAR (300)	Indicate the entire category path. Contains the name of the category path as shown to your partners in TB.One.
NAME-BASED IMPORT			
NAME	Yes	CHAR (300)	Indicate the entire category path. Contains the name of the category path as shown to your partners in TB.One.

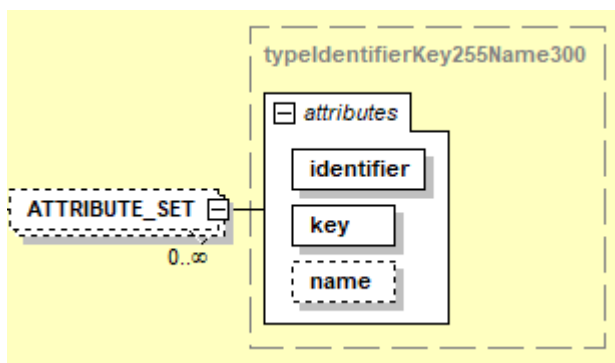
TBCATALOG / ... / STRUCTURE_ITEM / ATTRIBUTE_SET

You can use the node [TBCATALOG / CLASSIFICATION / ATTRIBUTE_SETS](#) to define required and recommended fields for your categories.

In this node (ATTRIBUTE_SET) the requirements from the node ATTRIBUTE_SETS are assigned to the respective structure item. If you provide the node ATTRIBUTE_SET, you have to indicate an *identifier* and a *key*. The *name* is optional (if you have not selected it as the *identifier*).



You can define attribute sets that are valid globally and attribute sets that are node-specific. You do not need assign global attribute sets to the STRUCTURE_ITEMS since they are automatically valid for all categories.



NODE	REQ.	TYPE	DESCRIPTION
ATTRIBUTE_SET	No	XML	Attribute definition for a category node.
ATTRIBUTE_SET [identifier]	Yes	ENUM	Indicator for the type of referencing used.
ATTRIBUTE_SET [key]	Yes	CHAR (50)	Value for the referencing type provided in <i>identifier</i> .
ATTRIBUTE_SET [name]	No	CHAR (100)	Name for the referencing type provided in <i>identifier</i> .

EXAMPLE (key-based):

```
<ATTRIBUTE_SET identifier="key" key="time"/>
```

EXAMPLE (name-based):

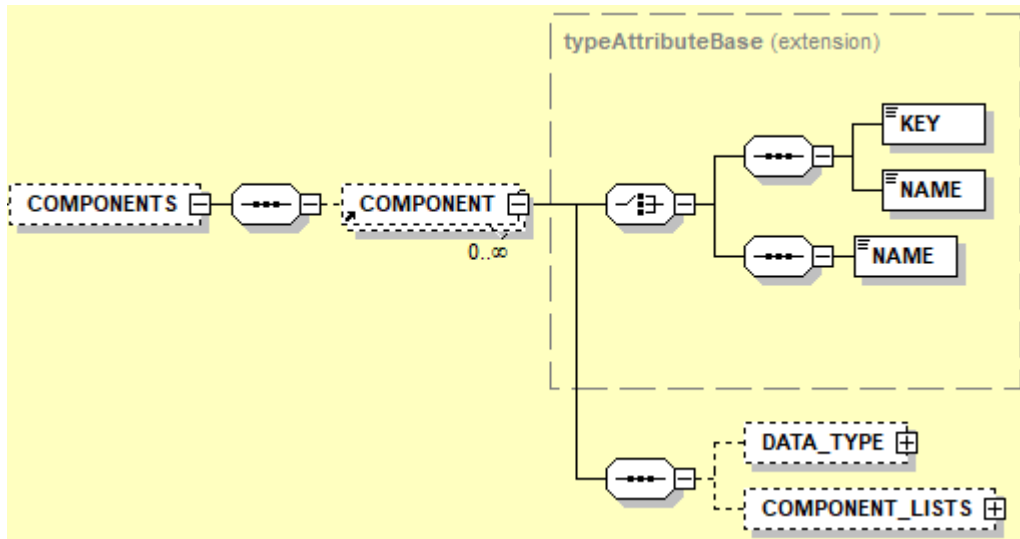
```
<ATTRIBUTE_SET identifier="name" key="Times"/>
```

TBCATALOG / CLASSIFICATION / COMPONENTS

Use components to define, which values brand/sales partners may use to describe their products and articles when selling on your channel. The partners will map their own data to the components you provide. Via components and attribute sets you can define rules for free text fields where component values are entered or provide a fixed list of component values which partners can map (node "COMPONENT_LISTS").



If you need to define the different values within a fixed list **with both key and name for each value**, use TBCATALOG / CLASSIFICATION / TAGS instead of TBCATALOG / CLASSIFICATION / COMPONENTS.



NODE	REQ.	TYPE	DESCRIPTION
COMPONENTS	No	XML	Component node.
COMPONENT	No	XML	Single component.
DATA_TYPE	No	XML	This node can be used to define data types and ranges of validity for the component's values.
COMPONENT_LISTS	No	XML	This node is used to transmit a list of values which are valid for the respective component. If you provide this node, the brand/sales partner will have to choose from the list when mapping values in their TB.One system.
KEY-BASED IMPORT			
KEY	Yes	CHAR (50)	The component's import key.

NODE	REQ.	TYPE	DESCRIPTION
NAME	Yes	CHAR (200)	The component's designation.
NAME-BASED IMPORT			
NAME	Yes	CHAR (200)	The component's designation.

EXAMPLE (key-based):

```

<COMPONENTS>
  <COMPONENT>
    <KEY>color</KEY>
    <NAME>Farbe</NAME>
  </COMPONENT>
</COMPONENTS>

```

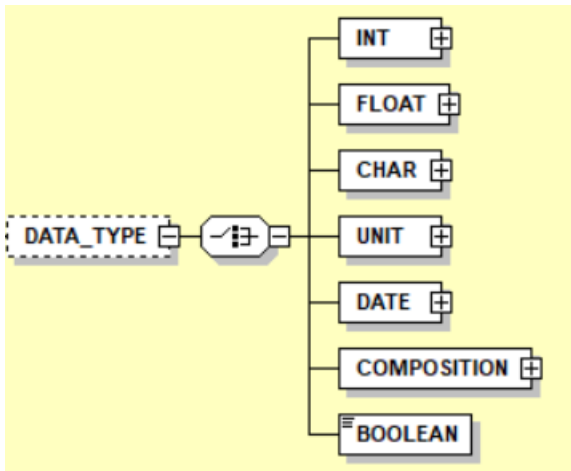
EXAMPLE (name-based):

```

<COMPONENTS>
  <COMPONENT>
    <NAME>Farbe</NAME>
  </COMPONENT>
</COMPONENTS>

```

TBCATALOG / ... / COMPONENT / DATA_TYPE



When you include the node DATA_TYPE, it must contain one of the child nodes.

NODE	REQ.	TYPE	DESCRIPTION
DATA_TYPE	No	XML	Use this node to restrict the values which the component may take to a specific data type. Within the individual data types' nodes, you can restrict the values further by defining a range of validity.
INT	One of these child nodes required	XML	Brand/sales partners must provide integers as the component's values.
FLOAT		XML	Brand/sales partners must provide floating-point numbers as the component's values.
CHAR		XML	Brand/sales partners must provide texts as the component's values.
UNIT		XML	With the node UNIT you can define in which unit the component's values are to be provided.
DATE		XML	Use this node if the component is used to transfer dates. You can define a format via the child node FORMAT.
COMPOSITION		XML	If you use the data type COMPOSITION, the brand/sales partners can combine multiple

NODE	REQ.	TYPE	DESCRIPTION
			properties during mapping for this component. This is especially useful when transferring the material composition of textiles. You can provide a list of values which can be used to select individual materials via a TAG that shares the component's name.
BOOLEAN		XML	Brand/sales partners must provide the value "true" or "false" for the component. Use BOOLEAN if there are only two possible values for the COMPONENT.

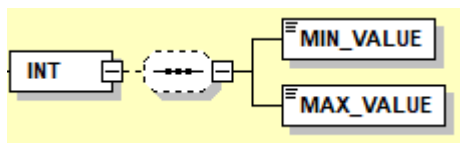
EXAMPLE

```

<COMPONENT>
  <KEY>customsNumber</KEY>
  <NAME>Tariff number CH</NAME>
  <DATA_TYPE>
    <INT/>
  </DATA_TYPE>
</COMPONENT>
<COMPONENT>
  <KEY>upper_material_clothing</KEY>
  <NAME>Upper Material</NAME>
  <DATA_TYPE>
    <COMPOSITION/>
  </DATA_TYPE>
</COMPONENT>

```

TBCATALOG / ... / COMPONENT / DATA_TYPE / INT



NODE	REQ.	TYPE	DESCRIPTION
INT	No	XML	Defines that the component's values must be integers. You can define a minimum and maximum value (they must also be integers).
MIN_VALUE	No	INT	Defines a minimum for the range of valid values. If you define a minimum value, you must also define a maximum value.
MAX_VALUE	No	INT	Defines a maximum for the range of valid values. If you define a maximum value, you must also define a minimum value.

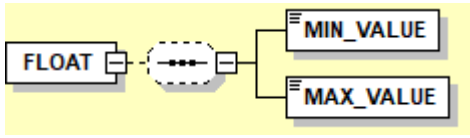
EXAMPLE

```

<COMPONENT>
  <KEY>customsNumber</KEY>
  <NAME>Tariff number CH</NAME>
  <DATA_TYPE>
    <INT>
      <MIN_VALUE>10000000000</MIN_VALUE>
      <MAX_VALUE>99999999999</MAX_VALUE>
    </INT>
  </DATA_TYPE>
</COMPONENT>

```

TBCATALOG / ... / COMPONENT / DATA_TYPE / FLOAT



NODE	REQ.	TYPE	DESCRIPTION
FLOAT	No	XML	Defines that the component's values must be floating-point numbers. You can define a minimum and maximum value.
MIN_VALUE	No	FLOAT	Defines a minimum for the range of valid values. If you define a minimum value, you must also define a maximum value. Use a point as the decimal separator.
MAX_VALUE	No	FLOAT	Defines a maximum for the range of valid values. If you define a maximum value, you must also define a minimum value. Use a point as the decimal separator.

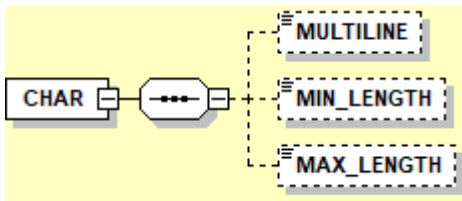
EXAMPLE

```

<COMPONENT>
  <KEY>sizeassignment</KEY>
  <NAME>Size assignment</NAME>
  <DATA_TYPE>
    <FLOAT>
      <MIN_VALUE>0,5</MIN_VALUE>
      <MAX_VALUE>999,5</MAX_VALUE>
    </FLOAT>
  </DATA_TYPE>
</COMPONENT>

```

TBCATALOG / ... / COMPONENT / DATA_TYPE / CHAR



NODE	REQ.	TYPE	DESCRIPTION
CHAR	No	XML	Defines that the component's values must be texts.
MULTILINE	No	BOOL	To allow text in multiple lines, provide the value "true" here.
MIN_LENGTH	No	INT	Minimum number of characters the component's values must contain.
MAX_LENGTH	No	INT	Maximum number of characters the component's values may contain.

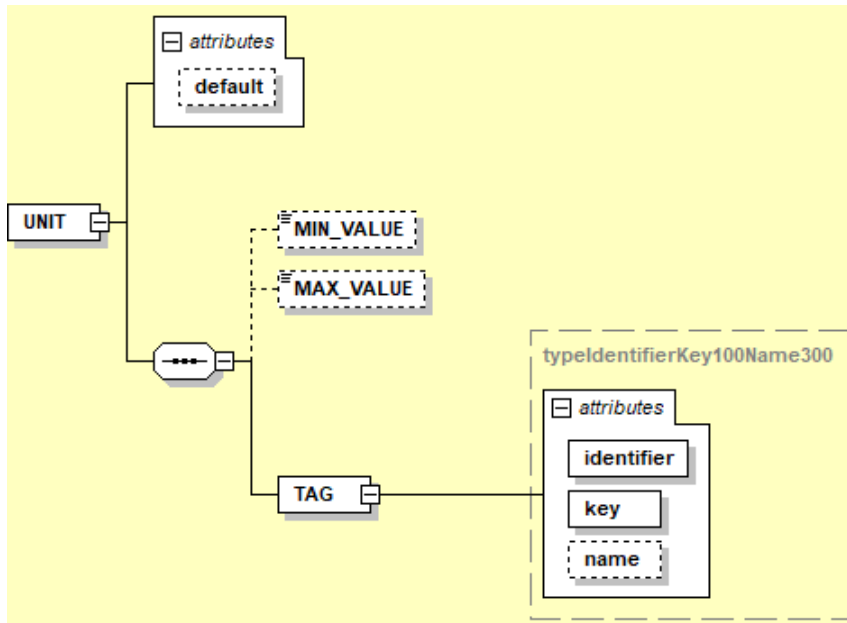
EXAMPLE

```

<COMPONENT>
  <KEY>doll_fashion_text</KEY>
  <NAME>Doll fashion description</NAME>
  <DATA_TYPE>
    <CHAR>
      <MULTILINE>true</MULTILINE>
      <MIN_LENGTH>0</MIN_LENGTH>
      <MAX_LENGTH>200</MAX_LENGTH>
    </CHAR>
  </DATA_TYPE>
</COMPONENT>

```


TBCATALOG / ... / COMPONENT / DATA_TYPE / UNIT



NODE	REQ.	TYPE	DESCRIPTION
UNIT	No	XML	Use the UNIT node to define a unit for the component's values.
UNIT[default]	No	CHAR (100)	Default value to be used for the component's unit (e.g. cm, inch, etc).
MIN_VALUE	No	INT	The component's minimum value.
MAX_VALUE	No	INT	The component's maximum value.
TAG	Yes	XML	To provide a list of possible values from which brand/sales partners can choose, you need to reference the TAG which contains the permitted values.

EXAMPLE

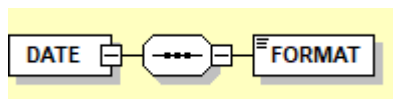
We want our partners to be able to choose the unit "cm" or "mm", the default choice being "cm". For that unit, they can choose values between 0 and 100.

```
<COMPONENT>
  <KEY>doll_fashion_length</KEY>
  <NAME>Doll fashion length</NAME>
  <DATA_TYPE>
    <UNIT default="cm">
      <MIN_VALUE>0</MIN_VALUE>
      <MAX_VALUE>100</MAX_VALUE>
      <TAG identifier="key" key="doll_fashion_length"/>
    </UNIT>
  </DATA_TYPE>
</COMPONENT>
```

The referenced tag (TBCATALOG / CLASSIFICATION / TAGS / TAG):

```
<TAG>
  <KEY>doll_fashion_length</KEY>
  <NAME>Doll fashion length</NAME>
  <VALUES>
    <VALUE>
      <KEY>mm</KEY>
      <NAME>mm</NAME>
    </VALUE>
    <VALUE>
      <KEY>cm</KEY>
      <NAME>cm</NAME>
    </VALUE>
  </VALUES>
</TAG>
```

TBCATALOG / ... / COMPONENT / DATA_TYPE / DATE



NODE	REQ.	TYPE	DESCRIPTION
DATE	No	XML	Include this node if the component is used to transfer date information. If you use this node, FORMAT is also mandatory.
FORMAT	Yes	CHAR (100)	Indicate the format "d.m.Y". This will result in values such as "24.12.2021".

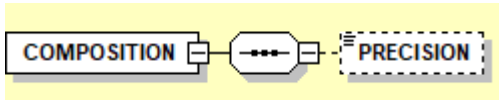
EXAMPLE

```

<COMPONENT>
  <KEY>launch_date</KEY>
  <NAME>Product launch</NAME>
  <DATA_TYPE>
    <DATE>
      <FORMAT>d.m.Y</FORMAT>
    </DATE>
  </DATA_TYPE>
</COMPONENT>

```

TBCATALOG / ... / COMPONENT / DATA_TYPE / COMPOSITION



NODE	REQ.	TYPE	DESCRIPTION
COMPOSITION	No	XML	If you include the node COMPOSITION in a component, the brand/sales partner and combine multiple properties for that component, which is especially useful for the material composition of textiles. You can provide a list of possible values for the COMPONENT. To do so, create a TAG with the same key as this COMPONENT. If you don't provide a list of values, partners will be able to enter their values in free text fields interpreted as percentages.
PRECISION	No	INT	Brands/Partners can enter percentages for the individual parts of a component with the node COMPOSITION. Use PRECISION to indicate the number of decimal places that can be used for these percentages.

EXAMPLE

In this example, users could provide any combination of the materials cotton, viscose, and silk for the component (along with their respective percentages). Partners could transfer component values such as "90.50% cotton, 9.50% viscose" or "50.00% cotton, 40.00% viscose, 10.00% silk".

```

<COMPONENT>
  <KEY>material_clothing</KEY>
  <NAME>Material</NAME>
  <DATA_TYPE>
    <COMPOSITION>
      <PRECISION>2</PRECISION>
    </COMPOSITION>
  </DATA_TYPE>
</COMPONENT>

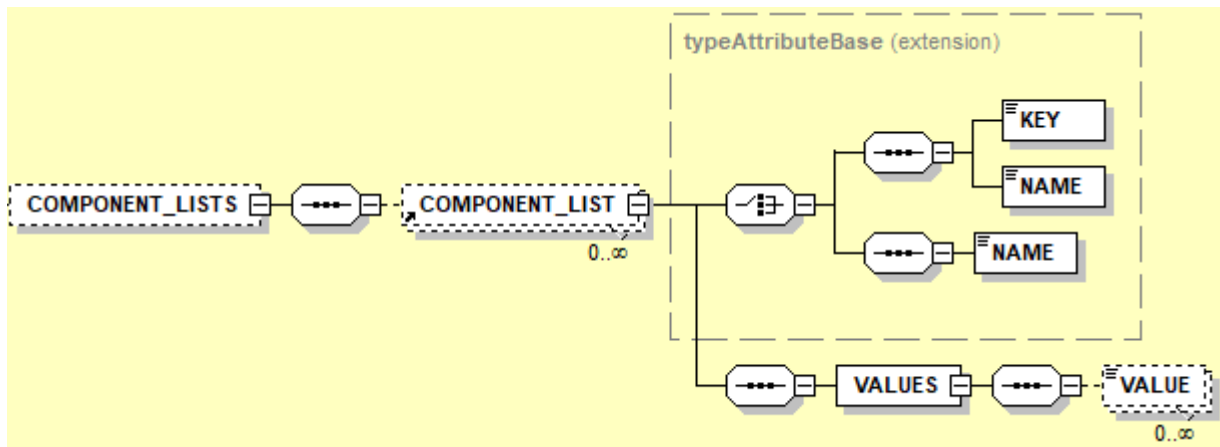
```

The referenced tag (TBCATALOG / CLASSIFICATION / TAGS / TAG):

```
<TAG>
  <KEY>material_clothing</KEY>
  <NAME>Material</NAME>
  <VALUES>
    <VALUE>
      <KEY>cotton</KEY>
      <NAME>cotton</NAME>
    </VALUE>
    <VALUE>
      <KEY>viscose</KEY>
      <NAME>viscose</NAME>
    </VALUE>
    <VALUE>
      <KEY>silk</KEY>
      <NAME>silk</NAME>
    </VALUE>
  </VALUES>
</TAG>
```

TBCATALOG / ... / COMPONENT / COMPONENT_LISTS

If you want to provide partners with a list of possible values to choose from for your components, you can do so by including COMPONENT_LISTS in your Classification.



If you include the node COMPONENT_LISTS you also need to provide at least one child-node COMPONENT_LIST with values.

NODE	REQ.	TYPE	DESCRIPTION
COMPONENT_LISTS	No	XML	The parent node for the component's component lists.
COMPONENT_LIST	Yes	XML	Contains the a component list. You can provide multiple lists.
VALUES	Yes	XML	The parent node for the component list's individual values.
VALUE	Yes	CHAR (256)	Each VALUE node contains one of the component list's values.
KEY-BASED IMPORT			
KEY	Yes	CHAR (255)	The component list's import key.
NAME	Yes	CHAR (300)	The component list's designation.
NAME-BASED IMPORT			

NODE	REQ.	TYPE	DESCRIPTION
NAME	Yes	CHAR (300)	The component list's designation.

EXAMPLE

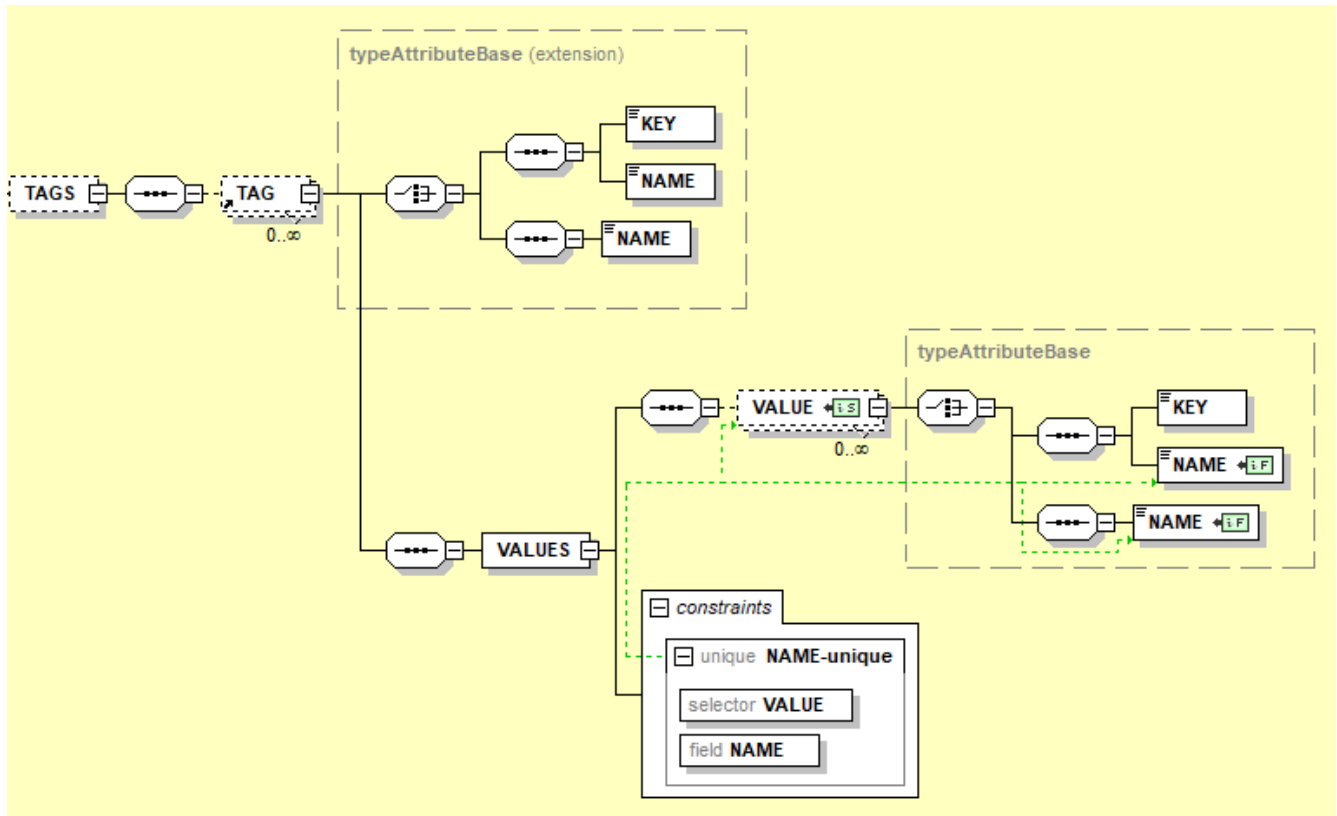
```

<COMPONENT>
  <KEY>filter_color</KEY>
  <NAME>Filter color</NAME>
  <COMPONENT_LISTS>
    <COMPONENT_LIST>
      <KEY>filter_color_adidas</KEY>
      <NAME>Filter color Adidas</NAME>
      <VALUES>
        <VALUE>Blue</VALUE>
        <VALUE>Red</VALUE>
        <VALUE>Green</VALUE>
      </VALUES>
    </COMPONENT_LIST>
    <COMPONENT_LIST>
      <KEY>filter_color_puma</KEY>
      <NAME>Filter color Puma</NAME>
      <VALUES>
        <VALUE>Navy</VALUE>
        <VALUE>Scarlet</VALUE>
        <VALUE>Lime</VALUE>
      </VALUES>
    </COMPONENT_LIST>
  </COMPONENT_LISTS>
</COMPONENT>

```

TBCATALOG / CLASSIFICATION / TAGS

With tags you can provide lists of possible values where each value has its own key and name (as opposed to component lists where only the list itself has an identifier). You may also need tags for the data types [UNIT](#) and [COMPOSITION](#).



NODE	REQ.	TYPE	DESCRIPTION
TAGS	No	XML	Parent node for defining tags.
TAG	No	XML	Each tag is specified in an individual TAG node.
VALUES	Yes	XML	Contains the tag's list of possible values.
VALUE	No	XML	Each of the tag's values has its own VALUE node.
KEY-BASED IMPORT (TAG and VALUE)			
KEY	Yes	CHAR(50)	The import key. Each TAG and VALUE has its own KEY node.
NAME	Yes	CHAR(200)	The designation. Each TAG and VALUE has its own NAME node.

NODE	REQ.	TYPE	DESCRIPTION
NAME-BASED IMPORT (TAG and VALUE)			
NAME	Yes	CHAR(200)	The designation. Each TAG and VALUE has its own NAME node.

EXAMPLE (key-based)

```

<TAGS>
  <TAG>
    <KEY>gender</KEY>
    <NAME>Gender</NAME>
    <VALUES>
      <VALUE>
        <KEY>men</KEY>
        <NAME>Men</NAME>
      </VALUE>
      <VALUE>
        <KEY>women</KEY>
        <NAME>Women</NAME>
      </VALUE>
    </VALUES>
  </TAG>
</TAGS>

```

EXAMPLE (name-based)

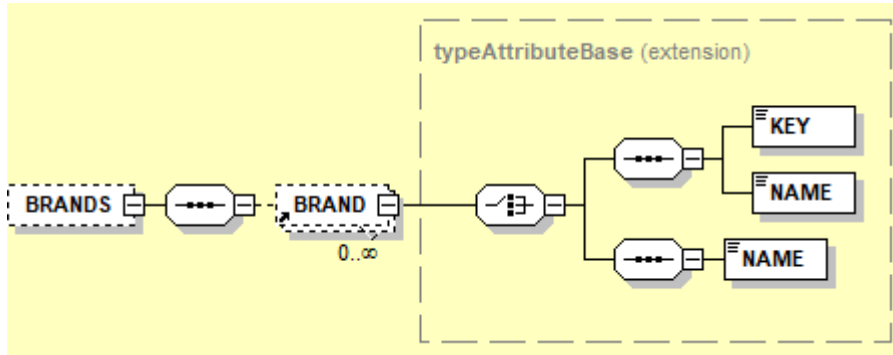
```

<TAGS>
  <TAG>
    <NAME>Gender</NAME>
    <VALUES>
      <VALUE>
        <NAME>Men</NAME>
      </VALUE>
      <VALUE>
        <NAME>Women</NAME>
      </VALUE>
    </VALUES>
  </TAG>
</TAGS>

```

TBCATALOG / CLASSIFICATION / BRANDS

If you provide a list of brands in your Classification, TB.One users can only use brands from that list when mapping for your channel. Providing a brand list is optional.



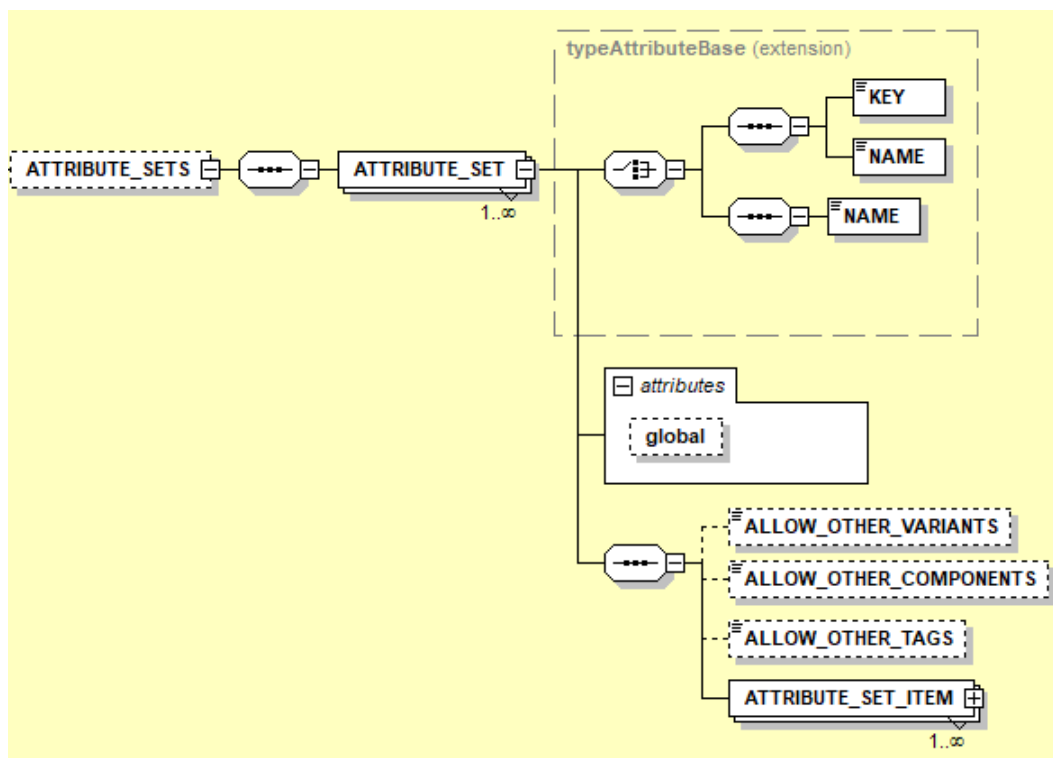
NODE	REQ.	TYPE	DESCRIPTION
BRANDS	No	XML	Parent node for defining brands.
BRAND	No	XML	Each brand is specified in an individual BRAND node.
KEY-BASED IMPORT			
KEY	Yes	CHAR (50)	The brand import key.
NAME	Yes	CHAR (200)	The brand designation.
NAME-BASED IMPORT			
NAME	Yes	CHAR (200)	The brand import key.

EXAMPLE (key-based)

```

<BRANDS>
  <BRAND>
    <KEY>JSF</KEY>
    <NAME>Jane Sinclair Fashion</NAME>
  </BRAND>
</BRANDS>
  
```

With attribute sets you can define recommended and required fields globally or for one of your Classification's categories (STRUCTURE ITEM).



NODE	REQ.	TYPE	DESCRIPTION
ATTRIBUTE_SETS	No	XML	Root node for attribute sets.
ATTRIBUTE_SET	No	XML	Defines an individual attribute set.
ATTRIBUTE_SET [global]	No	BOOL	Set to "true" if you want to use the attribute set in all categories you have defined in the STRUCTURE .
ALLOW_OTHER_VARIANTS	No	BOOL	Set to "false" if you only want to permit the variants you have defined in the respective list.
ALLOW_OTHER_COMPONENTS	No	BOOL	Set to "false" if you only want to permit the components you have defined.
ALLOW_OTHER_TAGS	No	BOOL	Set to "false" if you only want to permit the tags you have defined.

NODE	REQ.	TYPE	DESCRIPTION
ATTRIBUTE_SET_ITEM	Yes	XML	Defines which variant, component, or tag is part of the attribute set and which rules apply. An ATTRIBUTE_SET may contain multiple ATTRIBUTE_SET_ITEMS.
KEY-BASED IMPORT			
KEY	Yes	CHAR(50)	The attribute set's import key.
NAME	Yes	CHAR (200)	The attribute set's designation.
NAME-BASED IMPORT			
NAME	Yes	CHAR (200)	The attribute set's import key.

EXAMPLE (key-based)

```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true"/>
    <KEY>ASG1</KEY>
    <NAME>Global attribute set 1</NAME>
    <ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>
    <ATTRIBUTE_SET_ITEM>
      <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
        <COMPONENT identifier="key" key="Farbe"/>
      </ATTRIBUTE_SET_ITEM>
    </ATTRIBUTE_SET>
  </ATTRIBUTE_SETS>

```

EXAMPLE (name-based)

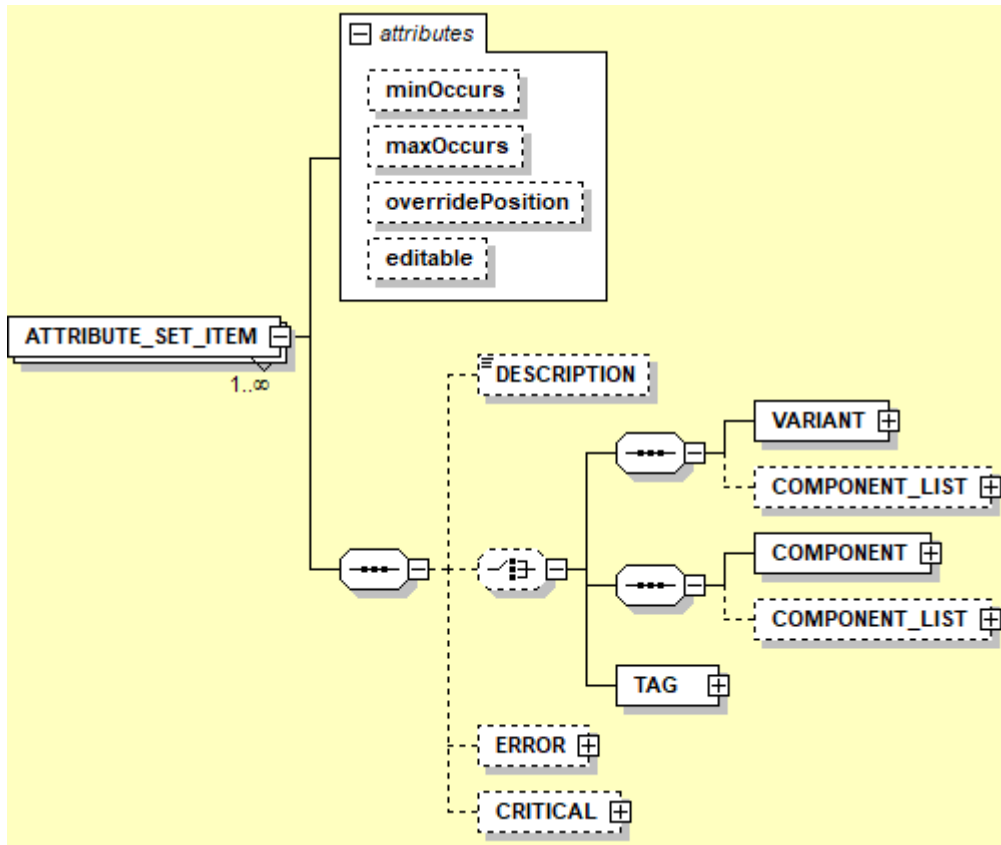
```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true"/>
    <NAME>Global attribute set 1</NAME>
    <ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>
    <ATTRIBUTE_SET_ITEM>
      <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
        <COMPONENT identifier="name" key="Farbe"/>
      </ATTRIBUTE_SET_ITEM>
    </ATTRIBUTE_SET>
  </ATTRIBUTE_SETS>

```

TBCATALOG / ... / ATTRIBUTE_SETS / ATTRIBUTE_SET_ITEM

ATTRIBUTE_SET_ITEM defines which variant, component, or tag is part of the attribute set and which rules apply.



NODE	REQ.	TYPE	DESCRIPTION
ATTRIBUTE_SET_ITEM	Yes	XML	One element of the attribute set.
ATTRIBUTE_SET_ITEM [minOccurs]	No	INT	Defines whether the attribute set item is mandatory or optional. "0" = optional "1" = mandatory If a brand/partner does not provide a piece of mandatory information, the affected products will not be exported to your system and an error message will be displayed to them.
ATTRIBUTE_SET_ITEM [maxOccurs]	No	INT	Defines, how often the attribute set item may be used for a product.

NODE	REQ.	TYPE	DESCRIPTION
			Indicate the number of possible assignments as a <i>positive integer</i> . If an unlimited amount of assignments is possible, use <i>"unbounded"</i> instead.
ATTRIBUTE_SET_ITEM [overridePosition]	No	STRING	Defines, to which of a product's elements in your shop (the parent product itself or its variants, e.g. size variants for fashion) the property from this attribute set item is assigned. Possible Values: ▪ "no": Export as indicated by the brand/merchant ▪ "article": Export at article level ▪ "product": Export at product level
ATTRIBUTE_SET_ITEM [editable]	No	BOOL	Defines whether a partner's choice is restricted to the values you offer via the COMPONENT_LIST, or if they can enter values of their own. Possible values: ▪ "0": partners have to choose from your COMPONENT_LIST. ▪ "1": partners may choose from your COMPONENT_LIST but may enter values of their own via a free text field instead.
DESCRIPTION	No	TEXT	Here, you can enter a text that explains to your partners what the intended use of this attribute set item is. The description will be shown to partners in TB.One's user interface during the value mapping process for this attribute set item.
VARIANT / COMPONENT / TAG	No	XML	References the variant, component, or tag described by the attribute set.
ERROR	No	XML	Defines the error level triggered when a partner does not fulfill the attribute set item's requirements and configures an error message.
CRITICAL	No	XML	ATTRIBUTE_SET_ITEMS marked as critical will display a pop-up-warning, when partners attempt

NODE	REQ.	TYPE	DESCRIPTION
			to change the mapping after the first export to your channel. This is especially useful if partners should not change the mapping.

EXAMPLE (key-based)

```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET>
    <KEY>flowers</KEY>
    <NAME>Flowers</NAME>
    <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
    <ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>
    <ALLOW_OTHER_TAGS>true</ALLOW_OTHER_TAGS>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="unbounded">
      <DESCRIPTION>Actual color may diverge</DESCRIPTION>
      <COMPONENT identifier="key" key="color"/>
      <COMPONENT_LIST identifier="key" key="filter_color_adidas"/>
    </ATTRIBUTE_SET_ITEM>
  </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>

```

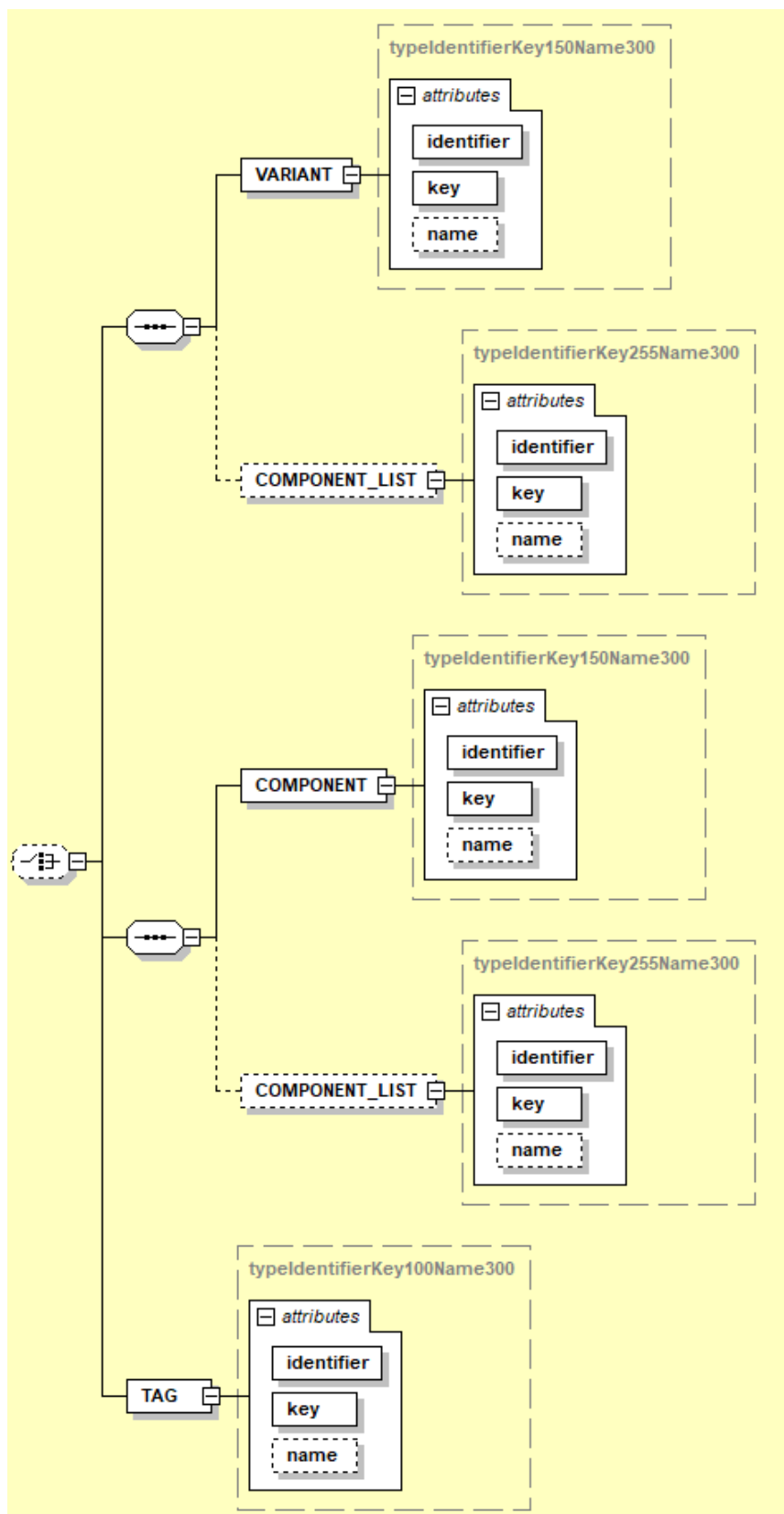
EXAMPLE (name-based)

```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET>
    <NAME>Flowers</NAME>
    <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
    <ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>
    <ALLOW_OTHER_TAGS>true</ALLOW_OTHER_TAGS>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="unbounded">
      <DESCRIPTION>Actual color may diverge</DESCRIPTION>
      <COMPONENT identifier="name" key="Farbe"/>
      <COMPONENT_LIST identifier="name" key="Filter color Adidas"/>
    </ATTRIBUTE_SET_ITEM>
  </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>

```

TBCATALOG / ... / ATTRIBUTE_SET_ITEM / (VARIANT/COMPONENT/TAG)

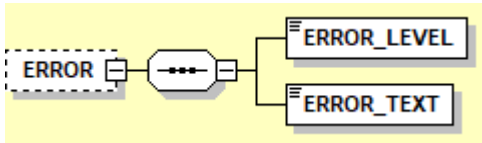


NODE	REQ.	TYPE	DESCRIPTION
ATTRIBUTE_SET_ITEM	Yes	XML	One element of an attribute set.
VARIANT	No	XML	Defines the property indicated in this node as variant-forming for the category (STRUCTURE_ITEM) to which this attribute set is assigned (or as variant-forming for all categories, if "ATTRIBUTE_SET[global]" = true). Use <i>identifier</i> and <i>key</i> to indicate the component used to form the variant (tags cannot be used to form variants).
COMPONENT	No	XML	Use <i>identifier</i> and <i>key</i> to indicate a component. That component will be added to the list of properties for the category (STRUCTURE_ITEM) to which this attribute set is assigned.
COMPONENT_LIST	No	XML	Use this node to add a list of possible values to this ATTRIBUTE_SET_ITEM's variant or component. The <i>identifier/key</i> needed here is defined in the respective list's node in COMPONENT_LISTS.
TAG	No	XML	Use <i>identifier</i> and <i>key</i> to indicate a tag. That tag will be added to the list of properties for the category (STRUCTURE_ITEM) to which this attribute set is assigned. Since tags themselves already contain a list of possible values, no COMPONENT_LIST is necessary.
VARIANT/ COMPONENT/ TAG[identifier]	Yes	ENUM	Indicator for the type of referencing used.
VARIANT/ COMPONENT/ TAG[key]	Yes	CHAR (50)	Value for the referencing type provided in <i>identifier</i> .
VARIANT/ COMPONENT/ TAG[name]	No	CHAR (100)	Value for the referencing type provided in <i>identifier</i> .

EXAMPLE

```
<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true">
    <KEY>ASG1</KEY>
    <NAME>Global attribute set 1</NAME>
    <ALLOW_OTHER_VARIANTS>false</ALLOW_OTHER_VARIANTS>
    <ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>
    <ALLOW_OTHER_TAGS>true</ALLOW_OTHER_TAGS>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
      <COMPONENT identifier="key" key="material composition"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
      <VARIANT identifier="key" key="size"/>
      <COMPONENT_LIST identifier="key" key="size"/>
    </ATTRIBUTE_SET_ITEM>
  </ATTRIBUTE_SET>
  <ATTRIBUTE_SET>
    <KEY>target_group</KEY>
    <NAME>Target group</NAME>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
      <COMPONENT identifier="key" key="color"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
      <TAG identifier="key" key="target_group"/>
    </ATTRIBUTE_SET_ITEM>
  </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>
```

TBCATALOG / ... / ATTRIBUTE_SET_ITEM / ERROR



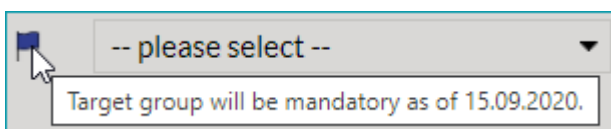
By adding the node ERROR to an attribute set item and assigning the respective attribute set to a category you can display notices and warnings about mappings to the brand/sales partners.

Via the ERROR_LEVEL you can define whether the ATTRIBUTE_SET_ITEMS requirements create a notice or a warning.

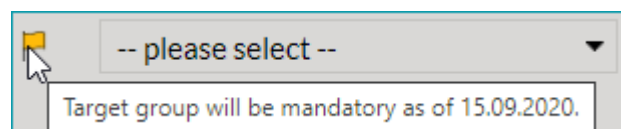
NODE	REQ.	TYPE	DESCRIPTION
ERROR	No	XML	Adds a message and an error level for the ATTRIBUTE_SET_ITEM to the brand/partner's value mapping menu.
ERROR_LEVEL	Yes	STRING	The ERROR_LEVEL defines which error type (Warning/Notice) is shown to the partner in TB.One's if the requirements from the ATTRIBUTE_SET_ITEM are not met in the mapping. For both types the export to Marketplaces does not exclude products/articles if mapping is not defined as required Possible values: "NOTICE": Adds a blue info icon in the value mapping menu. If partners do not fulfill the requirement, they will see a "hint" in the SolutionCenter's error list. "WARNING": Adds a yellow warning icon in the value mapping menu. If partners do not fulfill the requirement, they will see a "warning" in the SolutionCenter's error list.
ERROR_TEXT	Yes	STRING	This text determines, what message is shown to partners when they hover their mouse over the ERROR_LEVEL icon in the value mapping.

NOTICE/WARNING IN THE VALUE MAPPING MENU

Notice:



Warning:

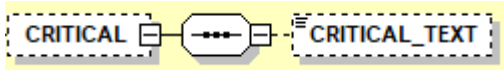


EXAMPLE

```
<ATTRIBUTE_SET global="true">
  <KEY>TargetGr</KEY>
  <NAME>Attribute Set Target Group</NAME>
  <ATTRIBUTE_SET_ITEM minOccurs="0">
    <COMPONENT identifier="key" key="TargetGr"/>
    <ERROR>
      <ERROR_LEVEL>WARNING</ERROR_LEVEL>
      <ERROR_TEXT>Target group will be mandatory as of 15.09.2020</ERROR_TEXT>
    </ERROR>
  </ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
<ATTRIBUTE_SET global="true">
  <KEY>AttributeSet1</KEY>
  <NAME>Attribute Set Season</NAME>
  <ATTRIBUTE_SET_ITEM minOccurs="1">
    <COMPONENT identifier="key" key="AttributeSet1"/>
    <ERROR>
      <ERROR_LEVEL>WARNING</ERROR_LEVEL>
      <ERROR_TEXT>Season information is mandatory</ERROR_TEXT>
    </ERROR>
  </ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
<ATTRIBUTE_SET global="true">
  <KEY>size</KEY>
  <NAME>Size</NAME>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1">
    <COMPONENT identifier="key" key="size"/>
    <COMPONENT_LIST identifier="key" key="size"/>
    <ERROR>
      <ERROR_LEVEL>NOTICE</ERROR_LEVEL>
      <ERROR_TEXT>Indicate sizes where necessary</ERROR_TEXT>
    </ERROR>
  </ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
```

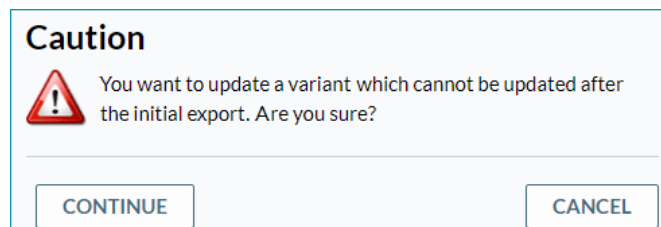
TBCATALOG / ... / ATTRIBUTE_SET_ITEM / CRITICAL

ATTRIBUTE_SET_ITEMS marked as critical will display a pop-up-warning, whenever partners attempt to change the mapping. This is especially useful if partners should not change the mapping after their first export.



NODE	REQ.	TYPE	DESCRIPTION
CRITICAL_TEXT	No	CHAR (255)	Defines text that will be displayed whenever a partner attempts to change the mapping of a value controlled by this ATTRIBUTE_SET_ITEM.

The critical text will be displayed in a pop-up message:

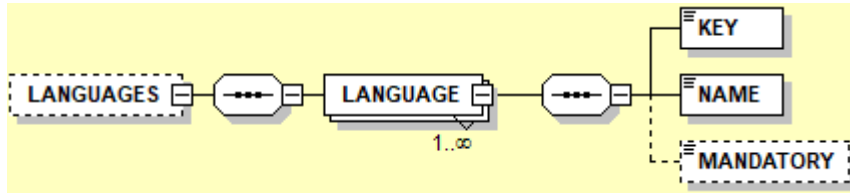


EXAMPLE

```
<ATTRIBUTE_SET global="true">
  <KEY>size</KEY>
  <NAME>Size</NAME>
  <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
    <COMPONENT identifier="key" key="size"/>
    <COMPONENT_LIST identifier="key" key="size"/>
    <CRITICAL_TEXT>You want to update a variant which cannot be updated after the
    initial export. Are you sure?</CRITICAL TEXT>
  </ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
```

TBCATALOG / CLASSIFICATION / LANGUAGES

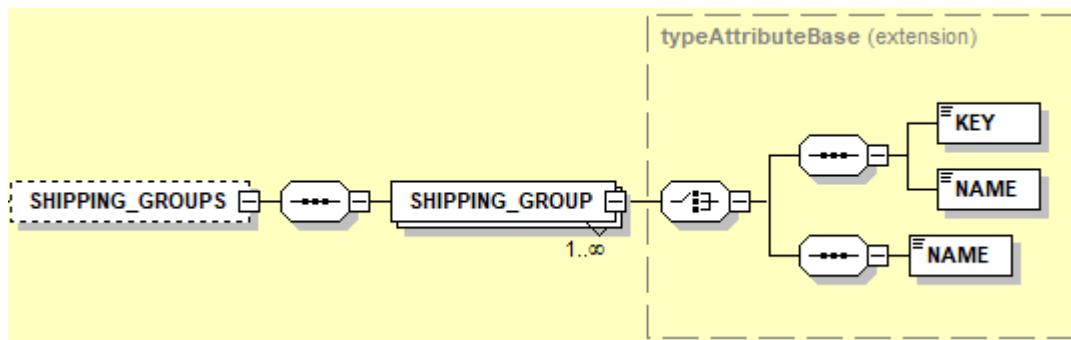
Here, you can define mandatory and optional languages for the product texts displayed in your shop. If you don't include a MANDATORY node with the value "true", the language will be optional.



NODE	REQ.	TYPE	DESCRIPTION
LANGUAGE	No	XML	Root node for a language.
KEY	Yes	LANG	Use language identifiers as defined by RFC 3066 for the keys ("de-DE", "en-US", "fr-FR", "nl-NL", etc).
NAME	Yes	CHAR (300)	The language's name.
MANDATORY	No	BOOL	Set the node MANDATORY to "true" if partners have to provide a version of their texts in the respective language.

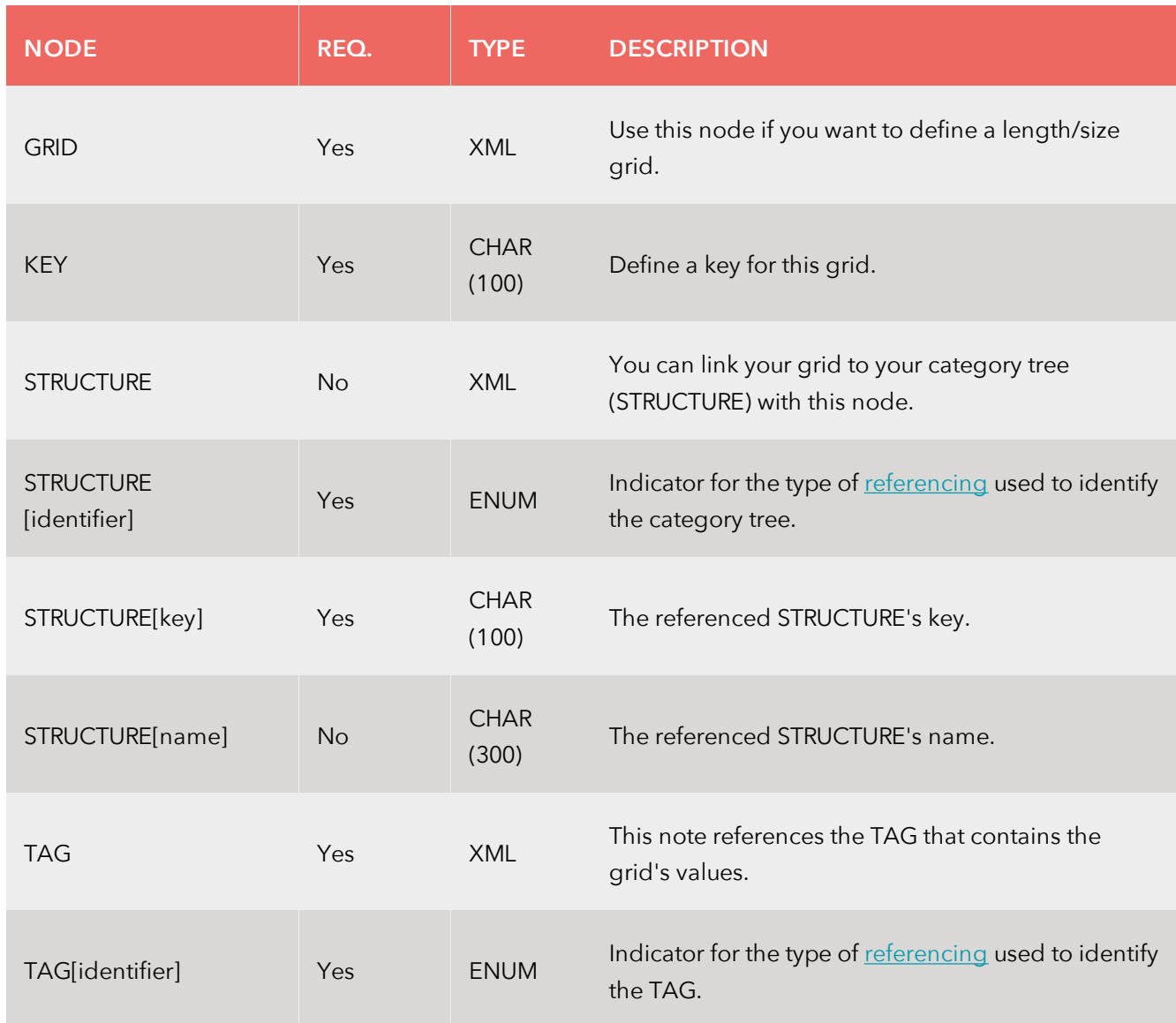
TBCATALOG / CLASSIFICATION / SHIPPING_GROUPS

Use the node SHIPPING_GROUPS if you want to define carriers partners have to use for their shipments.



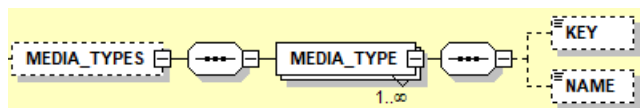
NODE	REQ.	TYPE	DESCRIPTION
SHIPPING_GROUP	Yes	XML	Use a separate SHIPPING_GROUP node for each carrier you want to define.
KEY	Yes	CHAR (100)	Provide an import key for the carrier.
NAME	Yes	CHAR (300)	Use this node to enter the carrier's name.

You may use this node to set up length/size grids to which partners can assign their sizes.



NODE	REQ.	TYPE	DESCRIPTION
TAG[key]	Yes	CHAR (100)	The referenced TAG's key.
TAG[name]	No	CHAR (300)	The referenced TAG's name.

TBCATALOG / CLASSIFICATION / MEDIA_TYPES



NODE	REQ.	TYPE	DESCRIPTION
MEDIA_TYPE	Yes	XML	Indicates a media type to which partners have to assign their media (model image, product on white background, etc).
KEY	No	CHAR (100)	Define a key for this media type.
NAME	No	CHAR (300)	Assign a self-explanatory name to the media type so partners know how to map their media.

6. EXAMPLE CLASSIFICATION

```
<?xml version="1.0" encoding="UTF-8"?>
<TBCATALOG xsi:noNamespaceSchemaLocation="https://api.trade-
server.net/schema/all_in_one/tb-cat_1_3_all_in_one.xsd" version="1.3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <CLASSIFICATION type="replace" key="Template4">
    <STRUCTURE identifier="key" key="Template4" name="Template4">
      <EXPORT_KEY>Template</EXPORT_KEY>
      <MIN_OCCURS>1</MIN_OCCURS>
      <MAX_OCCURS>1</MAX_OCCURS>
      <HAS_MAPPING>first</HAS_MAPPING>
      <STRUCTURE_ITEM>
        <KEY>Heels</KEY>
        <NAME>Heels</NAME>
        <PARENT_ID>0</PARENT_ID>
        <SORT>0</SORT>
        <ATTRIBUTE_SET identifier="key" key="Heels" name="Heels"/>
      </STRUCTURE_ITEM>
      <STRUCTURE_ITEM>
        <KEY>Shirts</KEY>
        <NAME>Shirts</NAME>
        <PARENT_ID>0</PARENT_ID>
        <SORT>0</SORT>
        <ATTRIBUTE_SET identifier="key" key="Shirts" name="Shirts"/>
      </STRUCTURE_ITEM>
    </STRUCTURE>
    <COMPONENTS>
      <COMPONENT>
        <KEY>manufacturer_size</KEY>
        <NAME>Manufacturer Size</NAME>
      </COMPONENT>
      <COMPONENT>
        <KEY>height_of_heel</KEY>
        <NAME>Height of heel (cm)</NAME>
        <DATA_TYPE>
          <INT>
            <MIN_VALUE>0</MIN_VALUE>
            <MAX_VALUE>100</MAX_VALUE>
          </INT>
        </DATA_TYPE>
      </COMPONENT>
      <COMPONENT>
        <KEY>gender</KEY>
        <NAME>Gender</NAME>
        <COMPONENT_LISTS>
```

```

        <COMPONENT_LIST>
            <KEY>gender</KEY>
            <NAME>Gender</NAME>
            <VALUES>
                <VALUE>male</VALUE>
                <VALUE>female</VALUE>
                <VALUE>unisex</VALUE>
            </VALUES>
        </COMPONENT_LIST>
    </COMPONENT_LISTS>
</COMPONENT>
<COMPONENT>
    <KEY>Material</KEY>
    <NAME>Material</NAME>
    <DATA_TYPE>
        <COMPOSITION/>
    </DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>integer-comp</KEY>
    <NAME>integer-comp</NAME>
    <DATA_TYPE>
        <INT>
            <MIN_VALUE>0</MIN_VALUE>
            <MAX_VALUE>100</MAX_VALUE>
        </INT>
    </DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>float-comp</KEY>
    <NAME>float-comp</NAME>
    <DATA_TYPE>
        <FLOAT>
            <MIN_VALUE>0</MIN_VALUE>
            <MAX_VALUE>100</MAX_VALUE>
        </FLOAT>
    </DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>char-comp</KEY>
    <NAME>char-comp</NAME>
    <DATA_TYPE>
        <CHAR>
            <MULTILINE>true</MULTILINE>
            <MIN_LENGTH>3</MIN_LENGTH>

```

```

        <MAX_LENGTH>55</MAX_LENGTH>
    </CHAR>
</DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>unit-comp</KEY>
    <NAME>unit-comp</NAME>
    <DATA_TYPE>
        <UNIT>
            <MIN_VALUE>0</MIN_VALUE>
            <MAX_VALUE>100</MAX_VALUE>
            <TAG identifier="key" key="units"></TAG>
        </UNIT>
    </DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>date-comp</KEY>
    <NAME>date-comp</NAME>
    <DATA_TYPE>
        <DATE>
            <FORMAT>d.m.Y</FORMAT>
        </DATE>
    </DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>composition-comp</KEY>
    <NAME>composition-comp</NAME>
    <DATA_TYPE>
        <COMPOSITION>
        </COMPOSITION>
    </DATA_TYPE>
</COMPONENT>
<COMPONENT>
    <KEY>boolean-comp</KEY>
    <NAME>boolean-comp</NAME>
    <DATA_TYPE>
        <BOOLEAN>
        </BOOLEAN>
    </DATA_TYPE>
</COMPONENT>
</COMPONENTS>
<TAGS>
    <TAG>
        <KEY>units</KEY>
        <NAME>Einheiten für Unit</NAME>
    </TAG>

```

```

<VALUES>
  <VALUE>
    <KEY>cm</KEY>
    <NAME>Zentimeter</NAME>
  </VALUE>
  <VALUE>
    <KEY>dm</KEY>
    <NAME>Dezimeter</NAME>
  </VALUE>
  <VALUE>
    <KEY>m</KEY>
    <NAME>Meter</NAME>
  </VALUE>
</VALUES>
</TAG>
<TAG>
  <KEY>Material</KEY>
  <NAME>Material</NAME>
  <VALUES>
    <VALUE>
      <KEY>cotton</KEY>
      <NAME>Cotton</NAME>
    </VALUE>
    <VALUE>
      <KEY>silk</KEY>
      <NAME>Silk</NAME>
    </VALUE>
    <VALUE>
      <KEY>nylon</KEY>
      <NAME>Nylon</NAME>
    </VALUE>
    <VALUE>
      <KEY>virgin_wool</KEY>
      <NAME>Virgin Wool</NAME>
    </VALUE>
  </VALUES>
</TAG>
<TAG>
  <KEY>color</KEY>
  <NAME>color</NAME>
  <VALUES>
    <VALUE>
      <KEY>black</KEY>
      <NAME>black</NAME>
    </VALUE>

```

```

    <VALUE>
      <KEY>white1</KEY>
      <NAME>White</NAME>
    </VALUE>
    <VALUE>
      <KEY>white2</KEY>
      <NAME>Frost White</NAME>
    </VALUE>
  </VALUES>
</TAG>
<TAG>
  <KEY>target_group</KEY>
  <NAME>Target Group</NAME>
  <VALUES>
    <VALUE>
      <KEY>Adult</KEY>
      <NAME>Adult</NAME>
    </VALUE>
    <VALUE>
      <KEY>Teen</KEY>
      <NAME>Teen</NAME>
    </VALUE>
    <VALUE>
      <KEY>Child</KEY>
      <NAME>Child</NAME>
    </VALUE>
  </VALUES>
</TAG>
<TAG>
  <KEY>sole</KEY>
  <NAME>Shoe Sole</NAME>
  <VALUES>
    <VALUE>
      <KEY>WOOD_CORK</KEY>
      <NAME>Cork</NAME>
    </VALUE>
    <VALUE>
      <KEY>RUBBER</KEY>
      <NAME>Rubber</NAME>
    </VALUE>
    <VALUE>
      <KEY>LATEX</KEY>
      <NAME>Latex</NAME>
    </VALUE>
    <VALUE>

```

```

        <KEY>LEATHER</KEY>
        <NAME>Leather</NAME>
    </VALUE>
</VALUES>
</TAG>
<TAG>
    <KEY>design</KEY>
    <NAME>Design</NAME>
    <VALUES>
        <VALUE>
            <KEY>dotted</KEY>
            <NAME>Dotted</NAME>
        </VALUE>
        <VALUE>
            <KEY>checkered</KEY>
            <NAME>Checkered</NAME>
        </VALUE>
    </VALUES>
</TAG>
<TAG>
    <KEY>fit</KEY>
    <NAME>Fit</NAME>
    <VALUES>
        <VALUE>
            <KEY>Boyfriend fit</KEY>
            <NAME>Boyfriend fit</NAME>
        </VALUE>
        <VALUE>
            <KEY>Regular fit</KEY>
            <NAME>Regular fit</NAME>
        </VALUE>
        <VALUE>
            <KEY>Skinny fit</KEY>
            <NAME>Skinny fit</NAME>
        </VALUE>
        <VALUE>
            <KEY>Slim fit</KEY>
            <NAME>Slim fit</NAME>
        </VALUE>
    </VALUES>
</TAG>
<TAG>
    <KEY>season</KEY>
    <NAME>Season</NAME>
    <VALUES>

```



```

        <VALUE>
            <KEY>NOOS</KEY>
            <NAME>Never out of stock</NAME>
        </VALUE>
    <VALUE>
        <KEY>Spring_20</KEY>
        <NAME>Spring 2020</NAME>
    </VALUE>
    <VALUE>
        <KEY>Winter_20</KEY>
        <NAME>Winter 2020</NAME>
    </VALUE>
</VALUES>
</TAG>
<TAG>
    <KEY>sizegrid</KEY>
    <NAME>Size-Grid</NAME>
    <VALUES>
        <VALUE>
            <KEY>size-schoes</KEY>
            <NAME>Shoe Sizes</NAME>
        </VALUE>
        <VALUE>
            <KEY>size-shirts</KEY>
            <NAME>Shirt Sizes</NAME>
        </VALUE>
    </VALUES>
</TAG>
<TAG>
    <KEY>size-schoes</KEY>
    <NAME>Shoe Sizes</NAME>
    <VALUES>
        <VALUE>
            <KEY>36</KEY>
            <NAME>36</NAME>
        </VALUE>
        <VALUE>
            <KEY>37</KEY>
            <NAME>37</NAME>
        </VALUE>
        <VALUE>
            <KEY>38</KEY>
            <NAME>38</NAME>
        </VALUE>
    <VALUE>

```

```

        <KEY>39</KEY>
        <NAME>39</NAME>
    </VALUE>
</VALUES>
</TAG>
<TAG>
    <KEY>size-shirts</KEY>
    <NAME>Shirt Sizes</NAME>
    <VALUES>
        <VALUE>
            <KEY>S</KEY>
            <NAME>S</NAME>
        </VALUE>
        <VALUE>
            <KEY>M</KEY>
            <NAME>M</NAME>
        </VALUE>
        <VALUE>
            <KEY>L</KEY>
            <NAME>L</NAME>
        </VALUE>
        <VALUE>
            <KEY>XL</KEY>
            <NAME>XL</NAME>
        </VALUE>
    </VALUES>
</TAG>
</TAGS>
<BRANDS>
    <BRAND>
        <KEY>Mizuno</KEY>
        <NAME>Mizuno</NAME>
    </BRAND>
    <BRAND>
        <KEY>Molten</KEY>
        <NAME>Molten</NAME>
    </BRAND>
<ATTRIBUTE_SETS>
    <ATTRIBUTE_SET global="true">
        <KEY>global</KEY>
        <NAME>global</NAME>
        <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
        <ALLOW_OTHER_COMPONENTS>>false</ALLOW_OTHER_COMPONENTS>
        <ALLOW_OTHER_TAGS>>false</ALLOW_OTHER_TAGS>
        <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">

```

```

    <TAG identifier="key" key="target_group" name="Target Group"/>
    <ERROR>
      <ERROR_LEVEL>WARNING</ERROR_LEVEL>
      <ERROR_TEXT>This Attribute will become mandatory on the 2020-09-15</ERROR_TEXT>
    </ERROR>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <TAG identifier="key" key="season" name="Season"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1" overridePosition="article">
    <TAG identifier="key" key="sizegrid" name="Size Grid"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="3" overridePosition="article">
    <TAG identifier="key" key="color" name="color"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="integer-comp" name="integer-comp"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="float-comp" name="float-comp"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="char-comp" name="char-comp"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="unit-comp" name="unit-comp"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="date-comp" name="date-comp"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="composition-comp" name="composition-comp"/>
  </ATTRIBUTE_SET_ITEM>
  <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="boolean-comp" name="boolean-comp"/>
  </ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
<ATTRIBUTE_SET global="false">
  <KEY>Heels</KEY>
  <NAME>Heels</NAME>
  <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
  <ALLOW_OTHER_COMPONENTS>false</ALLOW_OTHER_COMPONENTS>
  <ALLOW_OTHER_TAGS>false</ALLOW_OTHER_TAGS>
  <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1" overridePosition="article">

```

```

    <VARIANT identifier="key" key="manufacturer_size" name="Manufacturer
    size"/>
    <CRITICAL>
        <CRITICAL_TEXT>If this Attribute already got exported you are not
        allowed to change the values!</CRITICAL_TEXT>
    </CRITICAL>
</ATTRIBUTE_SET_ITEM>
<ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <TAG identifier="key" key="sole" name="Shoe Sole"/>
</ATTRIBUTE_SET_ITEM>
<ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
    <COMPONENT identifier="key" key="height_of_heel" name="Height of heel
    (cm)"/>
</ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
<ATTRIBUTE_SET global="false">
    <KEY>Shirts</KEY>
    <NAME>Shirts</NAME>
    <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
    <ALLOW_OTHER_COMPONENTS>false</ALLOW_OTHER_COMPONENTS>
    <ALLOW_OTHER_TAGS>false</ALLOW_OTHER_TAGS>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1" overridePosition="article">
        <VARIANT identifier="key" key="manufacturer_size" name="Manufacturer
        size"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1" overridePosition="article">
        <TAG identifier="key" key="Material" name="Material"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
        <TAG identifier="key" key="fit" name="Fit"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
        <TAG identifier="key" key="design" name="Design"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
        <COMPONENT identifier="key" key="gender" name="Gender"/>
        <COMPONENT_LIST identifier="key" key="gender" name="Gender"/>
    </ATTRIBUTE_SET_ITEM>
    <ATTRIBUTE_SET_ITEM minOccurs="0" maxOccurs="1" overridePosition="article">
        <COMPONENT identifier="key" key="Material" name="Material"/>
    </ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
</ATTRIBUTE_SETS>
<LANGUAGES>
    <LANGUAGE>
        <KEY>de-DE</KEY>

```

```
<NAME>Deutsch</NAME>
<MANDATORY>true</MANDATORY>
</LANGUAGE>
<LANGUAGE>
  <KEY>en-US</KEY>
  <NAME>English</NAME>
</LANGUAGE>
</LANGUAGES>
<SHIPPING_GROUPS>
  <SHIPPING_GROUP>
    <KEY>dhl</KEY>
    <NAME>DHL</NAME>
  </SHIPPING_GROUP>
  <SHIPPING_GROUP>
    <KEY>hermes</KEY>
    <NAME>Hermes</NAME>
  </SHIPPING_GROUP>
</SHIPPING_GROUPS>
<GRIDS>
  <GRID>
    <KEY>size</KEY>
    <STRUCTURE identifier="key" key="Template4"/>
    <TAG identifier="key" key="sizegrid"/>
  </GRID>
</GRIDS>
<MEDIA_TYPES>
  <MEDIA_TYPE>
    <KEY>mainpic</KEY>
    <NAME>Main Picture</NAME>
  </MEDIA_TYPE>
  <MEDIA_TYPE>
    <KEY>eek</KEY>
    <NAME>Energy Label</NAME>
  </MEDIA_TYPE>
</MEDIA_TYPES>
</CLASSIFICATION>
</TBCATALOG>
```