

TB.CAT 1.2 - CLASSIFICATION PUSH



STRUCTURE DEFINITION OF THE CHANNEL

© 2025 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.

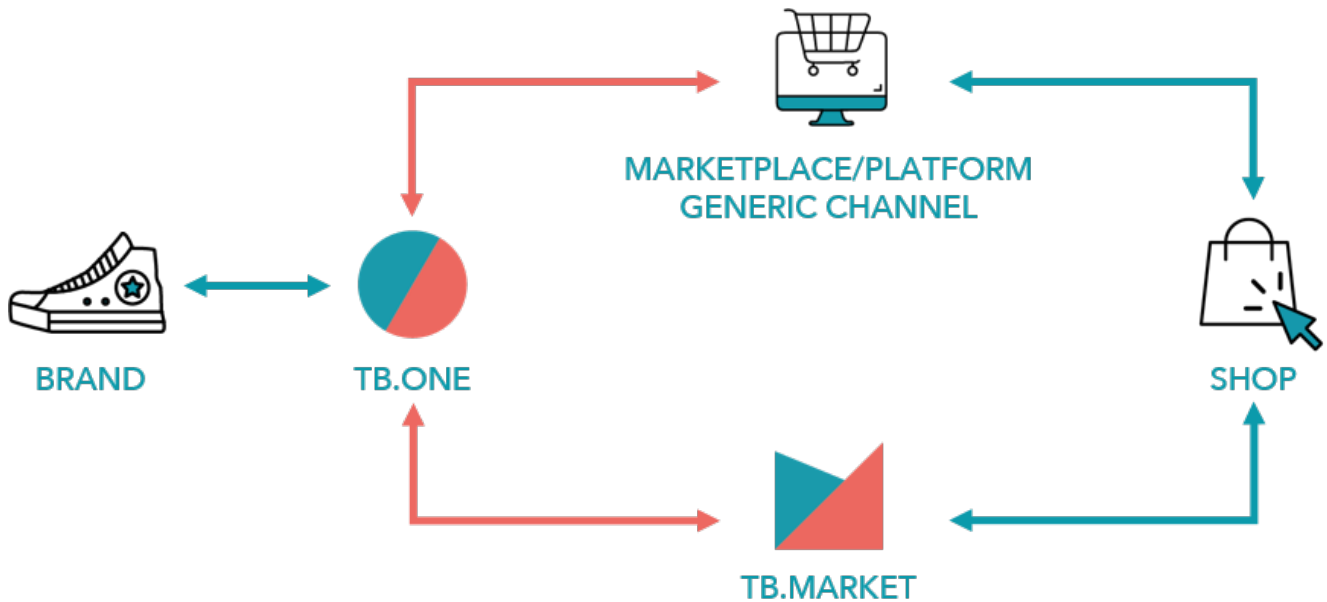
All other company names and logos and brand or product names are trademarks or registered trademarks of their respective owners.

TABLE OF CONTENTS

1. AIM AND PURPOSE OF THE DOCUMENT	5
2. DEFINITIONS AND CONVENTIONS	6
2.1 CHANNEL	6
2.2 STRUCTURE: CLASSIFICATION AND STRUCTURE_ITEMS	6
2.3 CONVENTIONS	6
3. DATA EXCHANGE VIA REST	7
3.1 CALL THE REST INTERFACE	7
3.2 REQUIRED FIELDS	8
3.3 RETURN VALUES	8
3.4 CLASSIFICATION PUSH	9
4. REFERENCING OF DATA	10
4.1 KEY	10
4.2 NAME	12
4.3 ID	14
5. STRUCTURE OF THE TB.CAT CLASSIFICATION	15
5.1 TBCATALOG / CLASSIFICATION	15
5.2 TBCATALOG / CLASSIFICATION / STRUCTURE / STRUCTURE_ITEM	17
5.3 TBCATALOG / ... / STRUCTURE_ITEM / ATTRIBUTE_SET	20
5.4 TBCATALOG / CLASSIFICATION / COMPONENTS	21
5.5 TBCATALOG / CLASSIFICATION / TAGS	22
5.6 TBCATALOG / CLASSIFICATION / BRANDS	24
5.7 TBCATALOG / CLASSIFICATION / LOGOS	25
5.8 TBCATALOG / CLASSIFICATION / ATTRIBUTE_SETS	27
5.9 TBCATALOG / ... / ATTRIBUTE_SETS / ATTRIBUTE_SET_ITEM	29
6. EXAMPLE CLASSIFICATION (FLAT STRUCTURE)	33

1. AIM AND PURPOSE OF THE DOCUMENT

This document is intended for operators of marketplaces, platforms and TB.Market and describes the transfer of categories and mandatory values to a connected TB.One.



2. DEFINITIONS AND CONVENTIONS

2.1 CHANNEL

As a channel we call each distribution channel; there is no distinction between marketplaces, platforms, and a generic channel (e.g. own shop). As a channel operator, you will receive a channel ID (Channel_ID) and a channel sign from Tradebyte through which you identify yourself when communicating. You will receive the channel ID and the channel sign from your Tradebyte contact.

2.2 STRUCTURE: CLASSIFICATION AND STRUCTURE_ITEMS

The connection of TB.One/TB.Market to a channel requires that you as a channel operator define a structure (e.g. categories or commodity groups) to which the offered products/articles must be assigned (by the data provider). This structure is created as part of a so-called classification which may not only contain the structure elements of your channel but also the definition of mandatory fields for components, attributes and brands. Within the classification you determine which mandatory information you expect from your suppliers, i.e. the connected TB.One clients. Mandatory fields are defined by means of attribute sets. These attribute sets can either be set globally across all categories or may be different for each category.

The provision to TB.Market or TB.One clients generally takes place in an XML file using the REST interface.

2.3 CONVENTIONS



Gray lines in the field descriptions may not be delivered. These are either system nodes, which are generated by TB.One, or currently unused nodes.

3. DATA EXCHANGE VIA REST

The data exchange between the channel and TB.One takes place by means of an XML file that contains your "classification". The transmission takes place via a REST interface. Whether you receive data from a TB.One or send it, you must validate it against the current TB.CAT data schema. Your Tradebyte representative will advise you about the model to be used. The validation against the current scheme must be checked at each sending/reception, so that any updates can be intercepted. You can find the schema files in our server at

https://api.core.tradebyte.com/schema/all_in_one/tb-cat_1_2_all_in_one.xsd

By means of the REST interface, you can transfer your XML file directly to your TB.Market/TB.One.

3.1 CALL THE REST INTERFACE

By means of the REST interface, you can transfer your classification file directly to your TB.Market/TB.One. After the successful PUT, the classification is valid immediately. Please note that due to new requirements/definitions in a classification (e.g. a new mandatory field) articles may no longer can be exported as long as this duty (value mapping) is not met by the data supplier. Basic structure of the REST URI:

https://rest.core.tradebyte.com/ACCOUNT-NR/TARGET/?channel=CHANNEL_ID

The REST interface call requires the authentication by user name and password.

3.2 REQUIRED FIELDS

FIELD	DESCRIPTION
USR/PWD	Username and password for the REST access
ACCOUNT-NR	Merchant number for which the classification is to apply.
TARGET	Specifies the resource group where the call aims to, in this case products (products)
CHANNEL_ID	The channel ID is required and will be provided by your Tradebyte contact. In the example calls the ID is referenced with CHID.

3.3 RETURN VALUES

Basically, a REST call always returns a message via XML. You either get the requested result or a status message, which is complemented by a status code in the HTTP header to indicate the successful or unsuccessful processing.

In addition to the respective content, you must take into account the http status code (header) response.

- 200 (all codes beginning with 2xx): call/processing successful.
- 400 (all codes starting with 4xx): call not successful or request incorrect.
- 500 (all codes starting with 5xx): processing error

3.4 CLASSIFICATION PUSH

METHOD	PUT	Send
TARGET	products	valid for the products
ACTION	classification	a classification file in TB.Cat XML format to TB. One that is valid for the specified channel and which serves the primary categorization of the products at product data creation. By means of the classification, categories and values are specified where the suppliers need to map their own values to.
CALLER	Channel	
EXAMPLE	...ACCOUNT-NR/products/classification/?channel=CHID Required XML file: see chapter 6³³.	
RESPONSE	200 OK	

4. REFERENCING OF DATA

The data transmitted can be exported with different types of referencing. The main distinction is the way in which this information can then be re-interpreted or are to be interpreted. Actually we distinguish two types of referencing within the classification:

4.1 KEY

The transferred information is exchanged based on a defined key. This is indicated by the given attribute **"identifier=key"**. The key to interpret is then defined as an attribute in the form "key=defined key". The actual content is transferred as an XML value if it is relevant to the given information (e.g. for components). This type of referencing is always used when the receiver of the data will assign and interpret the information accurately.

```

<TAGS>
  <TAG>
    <KEY>shippingcondition</KEY>
    <NAME>Versandzustand</NAME>
    <VALUES>
      <VALUE>
        <KEY>45</KEY>
        <NAME>montiert</NAME>
      </VALUE>
      <VALUE>
        <KEY>46</KEY>
        <NAME>teilmontiert</NAME>
      </VALUE>
      <VALUE>
        <KEY>47</KEY>
        <NAME>zerlegt</NAME>
      </VALUE>
    </VALUES>
  </TAG>
</TAGS>
...
<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true">
    <NAME>globales Mapping</NAME>
  </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>

```

```
<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="1">  
    <TAG identifier="key" key="shippingcondition"/>  
</ATTRIBUTE_SET_ITEM>  
</ATTRIBUTE_SET>  
</ATTRIBUTE_SETS>
```

4.2 NAME

The transferred information is provided based on the name (designation) which the sender defined. This is indicated by the given attribute "**identifier=name**". The identifier/name will be indicated with the attribute "**key=<Name/Designator>**". The actual content is transferred as an XML value if it is relevant to the given information. This type of referencing is used when the receiver receives data which can be assigned by means of the transferred area (component, attribute, and brand) but any information can/should not be interpreted accurately.

```
<TAGS>
  <TAG>
    <KEY>shippingcondition</KEY>
    <NAME>Versandzustand</NAME>
    <VALUES>
      <VALUE>
        <KEY>45</KEY>
        <NAME>montiert</NAME>
      </VALUE>
      <VALUE>
        <KEY>46</KEY>
        <NAME>teilmontiert</NAME>
      </VALUE>
      <VALUE>
        <KEY>47</KEY>
        <NAME>zerlegt</NAME>
      </VALUE>
    </VALUES>
  </TAG>
</TAGS>
...
<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true">
    <NAME>globales Mapping</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="1">
```

```
<TAG identifier="name" key="Versandzustand"/>  
</ATTRIBUTE_SET_ITEM>  
</ATTRIBUTE_SET>  
</ATTRIBUTE_SETS>
```

4.3 ID

The provided information is transmitted based on the database ID of the sender. This is indicated by the given attribute "**identifier=id**". The database ID is then indicated with the attribute "**key=<DB_ID>**". The actual content is transferred as an XML value if it is relevant to the given information.

In addition, you get an appropriate CLASSIFICATION part for all the nodes that are indicated as ID-based. This is included as a table of contents at the beginning of the TB.Cat.xml. This CLASSIFICATION lists the used database IDs together with the name defined by the sender for each node (components, attributes, etc.).

This type of referencing is used when the transmitter is the leader for the transfer of information. Normally, this can only be used when there is only one data provider, since the same information is communicated with different IDs for different sender.

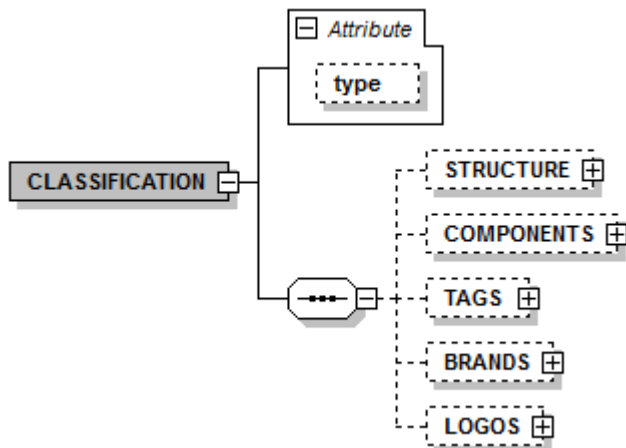
```
<PRODUCTDATA type="full">
  <PRODUCT>
    ...
    <P_BRAND identifier="id" key="577">Superjackenmeister</P_BRAND>
    <P_COMPONENTDATA>
      <P_COMPONENT identifier="id" key="4">
        <VALUE xml:lang="x-default">100% Jackenmaterial</VALUE>
      </P_COMPONENT>
    </P_COMPONENTDATA>
    <P_TAGS>
      <P_TAG identifier="id" key="55" name="Geschlecht">
        <VALUES>
          <VALUE identifier="id" key="1953">Herren</VALUE>
        </VALUES>
      </P_TAG>
    ...
  </PRODUCT>
</PRODUCTDATA>
```

5. STRUCTURE OF THE TB.CAT CLASSIFICATION

5.1 TBCATALOG / CLASSIFICATION



The classification part optionally contains lists with categories, components, attributes, brands and logos. This part of the TB.CAT serves the definition of the desired (mandatory) values by you.



NODE	MUST	DATA TYPE	DESCRIPTION
CLASSIFICATION[type]	N	ENUM	Type of the Classification feed Possible values are: merge = existing and provided values are summarized; not provided values are ignored. This is the default setting replace = not supplied values are deleted
STRUCTURE ¹⁷	N	XML	Categories
COMPONENTS ²¹	N	XML	Properties with continuous text values
TAGS ²²	N	XML	"Machine" evaluable properties with values from a value list such as filter attributes in online shops
BRANDS ²⁴	N	XML	Brand list
LOGOS ²⁵	N	XML	Logo/Icon list

NODE	MUST	DATA TYPE	DESCRIPTION
ATTRIBUTE_SETS ²⁷	N	XML	Definition of mandatory values
SUB_SUPPLIERS	N	XML	Only relevant in TB.CAT Export - indicates within a TB.Market export a reference to the connected TB.One clients/suppliers

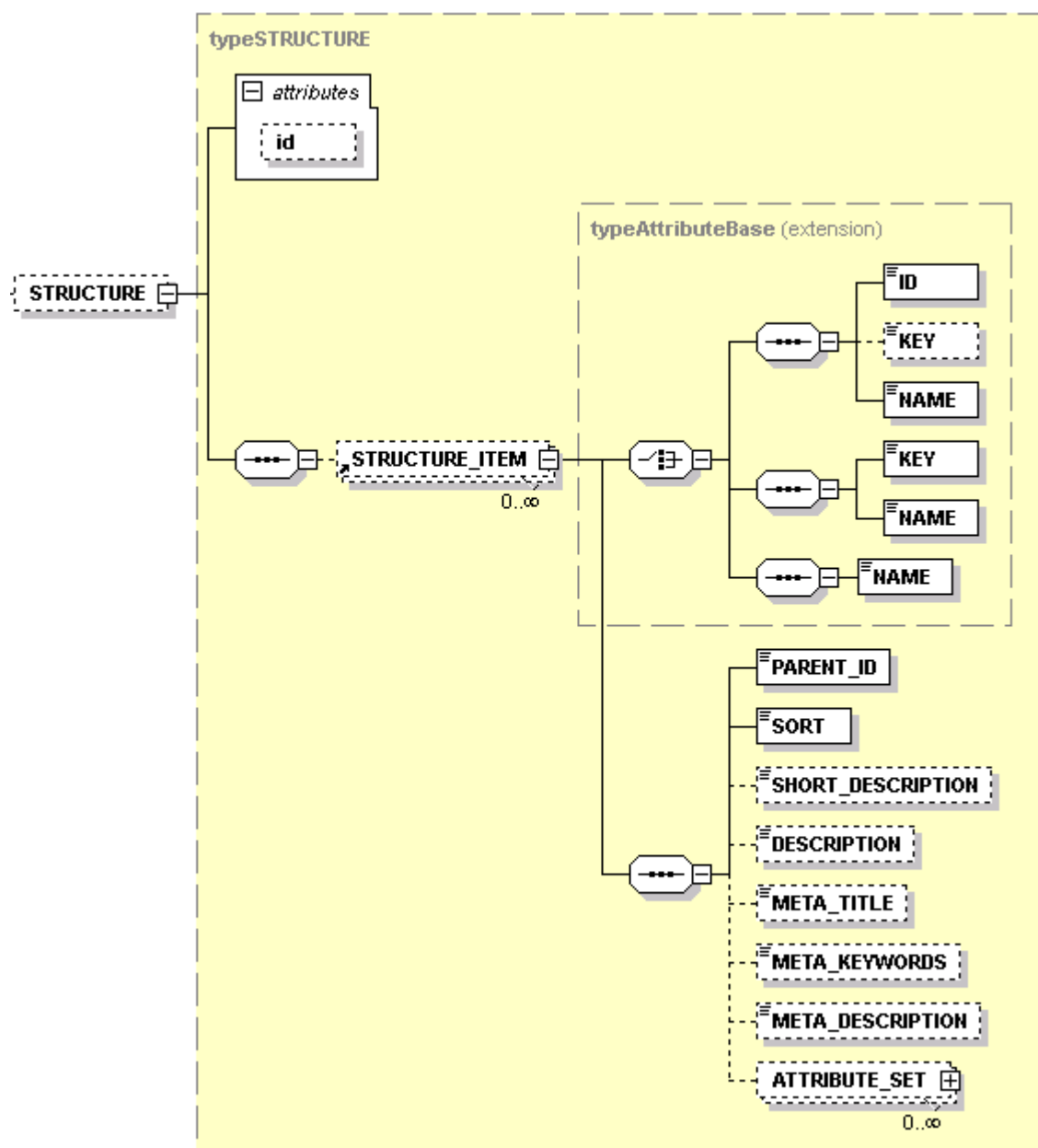
Sample

```

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

```


Categories have to be maintained in a flat hierarchy.



NODE	MUST	DATA TYPE	DESCRIPTION
STRUCTURE [id]	N	INT	Variant of the category tree (currently not used)
STRUCTURE_ITEM	N	XML	Category node
PARENT_ID	Y	INT	Reference to the ID of the parent node (Main Category: value = 0). Always pass "0" for a flat hierarchy.
SORT	Y	INT	Sort value of the category within the associated parent node. Without sorting values, TB.One sorts the categories alphabetically.
SHORT_DESCRIPTION	N	TEXT	Brief description of the category node
DESCRIPTION	N	TEXT	Description of the category node
META_TITLE	N	CHAR(255)	Meta title of the category node
META_KEYWORDS	N	CHAR(255)	Meta Keywords of the category node
META_DESCRIPTION	N	CHAR(255)	Meta-Description of the category node
ATTRIBUTE SET ²⁹	N	XML	Assignment of attribute sets to this category node. The definition of the attribute sets take place in the node "Attribute Sets".
IMPORT VIA ID			
ID	Y	INT	Unique ID of the category node
KEY	N	CHAR(50)	Import key of the category node
NAME	Y	CHAR(200)	Name of the category node

IMPORT VIA KEY

KEY	Y	CHAR(50)	Import key of the category node
NAME	Y	CHAR(200)	Name of the category node

IMPORT VIA NAME

NAME	Y	CHAR(200)	Name of the category node
------	---	-----------	---------------------------

Sample XML (key-based, flat)

```
<STRUCTURE>
  <STRUCTURE_ITEM>
    <KEY>bush</KEY>
    <NAME>Büsche</NAME>
    <PARENT_ID>0</PARENT_ID>
    <SORT>10</SORT>
  </STRUCTURE_ITEM>
  <STRUCTURE_ITEM>
    <KEY>4712</KEY>
    <NAME>Büsche | Beeren</NAME>
    <PARENT_ID>0</PARENT_ID>
    <SORT>20</SORT>
  </STRUCTURE_ITEM>
</STRUCTURE>
```

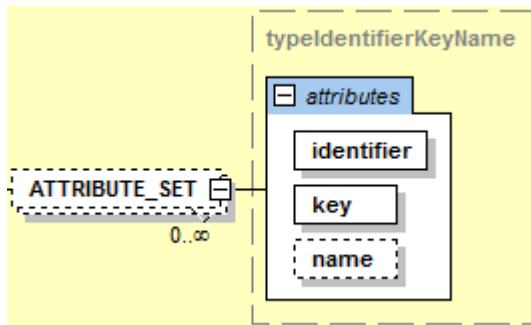
The result of the category list under "Channels > Channel name > Categories":

Categories					
	Channel ID	Designation	Assignability	Date	Products
			-- all --		-- all --
☐	2223	Beauty Frauen Düfte (Hauptebene)			0
☐	2225	Beauty Frauen Düfte Deo			0
☐	2226	Beauty Frauen Düfte Körperspray			0

5.3 TBCATALOG / ... / STRUCTURE_ITEM / ATTRIBUTE_SET



By means of the node [TBCATALOG / CLASSIFICATION / ATTRIBUTE_SETS](#) ²⁹ you can define mandatory and desired fields for your categories. Here they are assigned to the current structure node.



NODE	MUST	DATA TYPE	DESCRIPTION
ATTRIBUTE_SET	N		Attribute definition for a category node
ATTRIBUTE_SET[identifier]	Y	ENUM	Identifier for referencing ¹⁰
ATTRIBUTE_SET[key]	Y	CHAR(50)	Value stored in the identifier assignment statement
ATTRIBUTE_SET[name]	N	CHAR(100)	Name stored in the identifier assignment statement

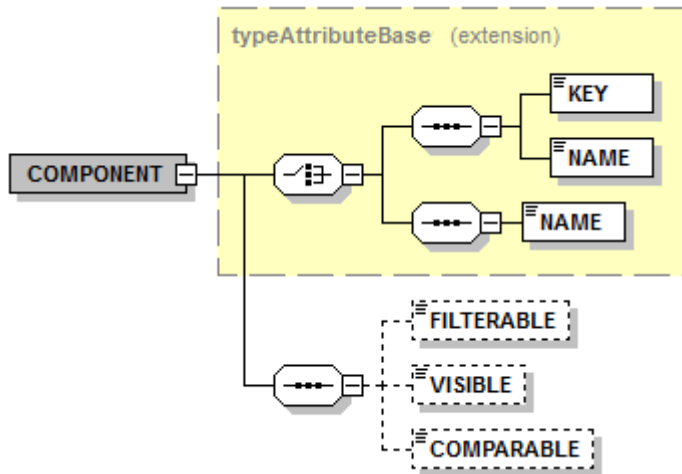
Sample XML (key-based)

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

Sample XML (name-based)

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

5.4 TBCATALOG / CLASSIFICATION / COMPONENTS



NODE	MUST	DATA TYPE	DESCRIPTION
COMPONENTS	N	XML	Component node
COMPONENT	N	XML	Individual component
COMPONENT[filterable]	N	BOOLEAN	For the use of the component in a generic channel you can specify whether the component is to serve as a filter or comparison criterion or whether it is visible. This is optional. Is displayed in TB.One/TB.Market, but only passed on, so that for example the web shop can further process the information
COMPONENT[visible]	N	BOOLEAN	
COMPONENT[comparable]	N	BOOLEAN	

IMPORT VIA KEY

KEY	Y	CHAR(50)	Import key of the node
NAME	Y	CHAR(200)	Name of the node

IMPORT VIA NAME

Name	Y	CHAR(200)	Name of the node
------	---	-----------	------------------

Sample XML (name-based)

```

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

  </COMPONENT>/>
</COMPONENTS>

```

Sample XML (key-based)

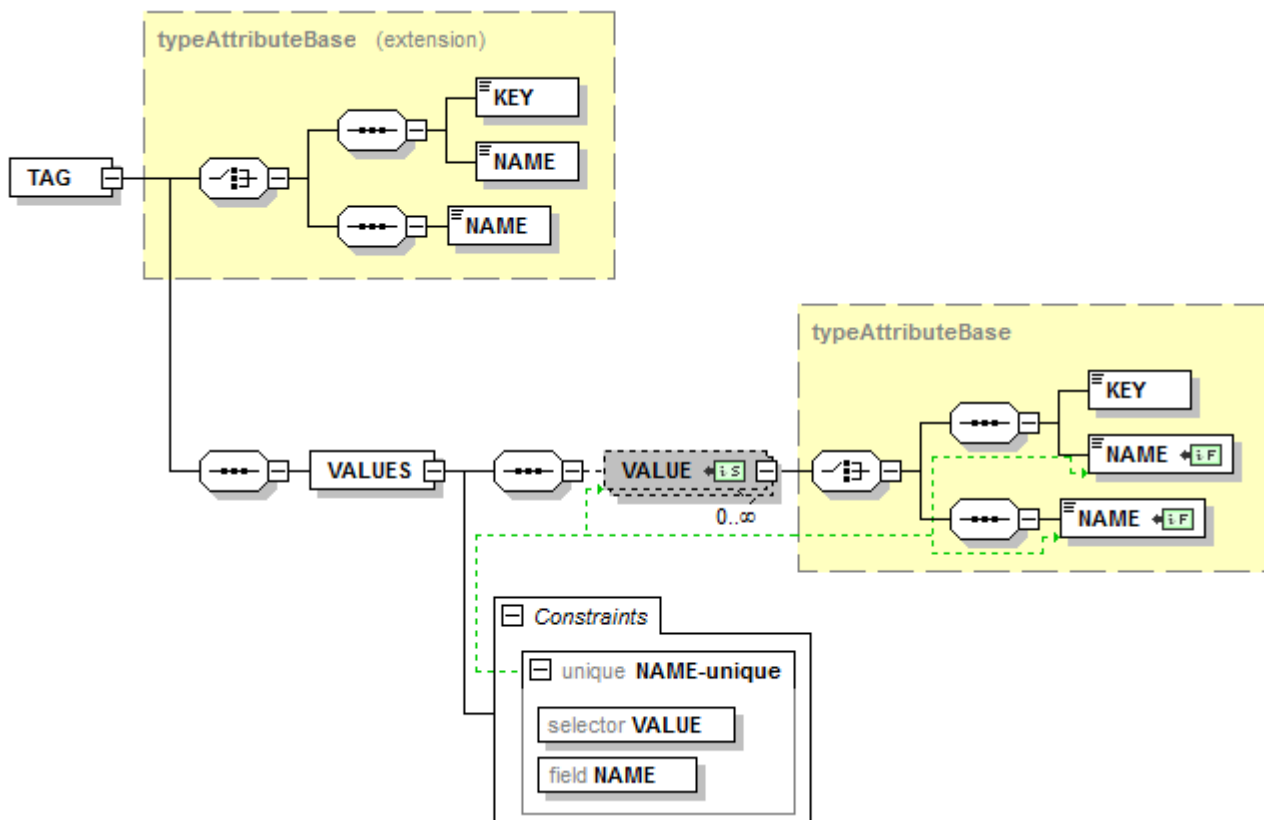
```

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

  </COMPONENT>/>
</COMPONENTS>

```

5.5 TBCATALOG / CLASSIFICATION / TAGS



NODE	MUST	DATA TYPE	DESCRIPTION
TAGS	N	XML	Definition of attributes
TAG	N	XML	Definition of one attribute
VALUES	Y	XML	Definition of a values list for the attribute
VALUE	N	XML	Single value

IMPORT VIA KEY

KEY	Y	CHAR(50)	Import key of the node
NAME	Y	CHAR(200)	Name of the node (unique)

IMPORT VIA NAME

NAME	Y	CHAR(200)	Name of the node (unique)
------	---	-----------	---------------------------

Sample XML (key-based)

```

<TAGS>
  <TAG>
    <KEY>gender</KEY>
    <NAME>Zielgruppe</NAME>
    <VALUES>
      <VALUE>
        <KEY>male</KEY>
        <NAME>Mann</NAME>
      </VALUE>
      <VALUE>
        <KEY>female</KEY>
        <NAME>Frau</NAME>
      </VALUE>
    </VALUES>
  </TAG>

```

</TAGS>

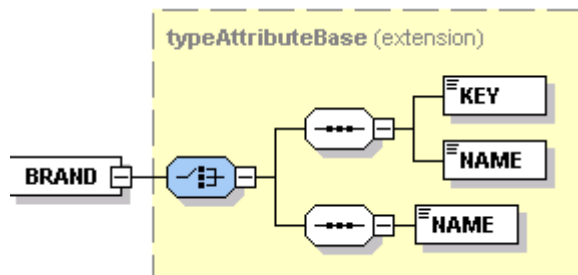
Sample XML (name-based)

```
<TAGS>
  <TAG>
    <NAME>Geschlecht</NAME>
    <VALUES>
      <VALUE>
        <NAME>Mann</NAME>
      </VALUE>
    </VALUES>
  </TAG>
</TAGS>
```

5.6 TBCATALOG / CLASSIFICATION / BRANDS



If you define a brand list within the Classification, a TB.One user can only use brands of your brand list for value mapping. The statement of brands is optional.



NODE	MUST	DATA TYPE	DESCRIPTION
BRANDS	N	XML	Brand node
BRAND	N	XML	Definition of a brand
IMPORT VIA KEY			
KEY	Y	CHAR(50)	Import key of the node

IMPORT VIA KEY

NAME	Y	CHAR(200)	Name of the node
------	---	-----------	------------------

IMPORT VIA NAME

NAME	Y	CHAR(200)	Name of the node
------	---	-----------	------------------

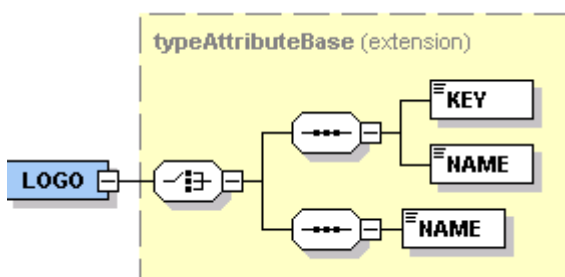
Sample XML (key-based)

```
<BRANDS>
  <BRAND>
    <KEY>supjam</KEY>
    <NAME>Superjackenmeister</NAME>
  </BRAND>
</BRANDS>
```

5.7 TBCATALOG / CLASSIFICATION / LOGOS



Logos are usually used to represent standardized product features, such as care symbols (30° wash), certifications (FSC) or features (Bluetooth). The provision in the classification part is not necessary.



NODE	MUST	DATA TYPE	DESCRIPTION
LOGOS	N	XML	Logo node
LOGO	N	XML	Definition of a logo

IMPORT VIA KEY

KEY	Y	CHAR(50)	Import key of the node
NAME	Y	CHAR(200)	Name of the node

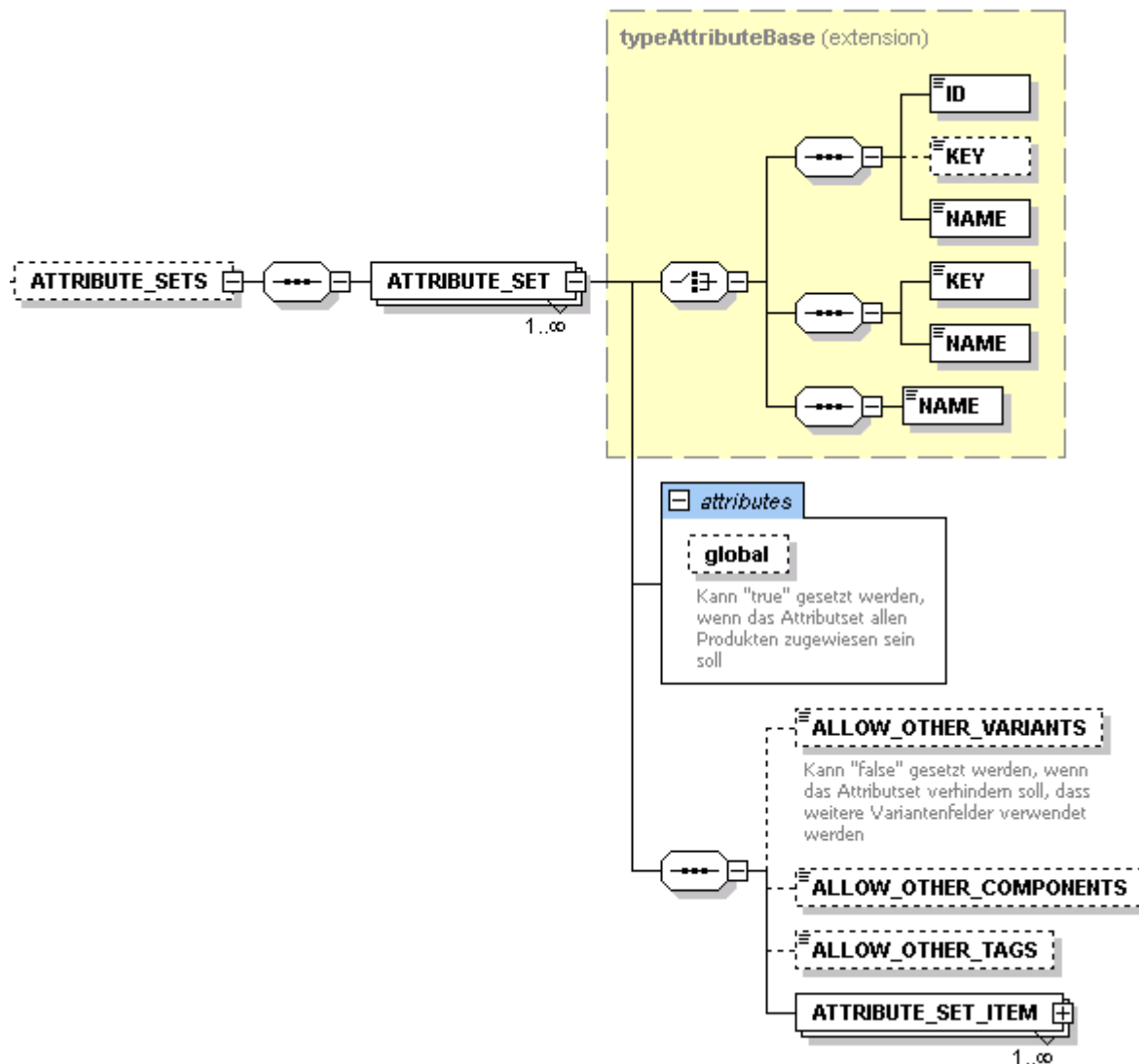
IMPORT VIA NAME

NAME	Y	CHAR(200)	Name of the node
------	---	-----------	------------------

5.8 TBCATALOG / CLASSIFICATION / ATTRIBUTE_SETS



By means of attribute sets, you can define mandatory and desire fields within your categories.



NODE	MUST	DATA TYPE	DESCRIPTION
ATTRIBUTE_SETS	N	XML	Root node for attribute sets
ATTRIBUTE_SET	N	XML	Attribute set definition
ATTRIBUTE_SET[global]	N	BOOLEAN	If you set the attribute set to "global", it is applied to all defined category nodes.

NODE	MUST	DATA TYPE	DESCRIPTION
ALLOW_OTHER_VARIANTS	N	BOOLEAN	If you wish to suppress any other values (variants, components, attributes) in addition to the values defined by you, you set this value to "false".
ALLOW_OTHER_COMPONENTS	N	BOOLEAN	
ALLOW_OTHER_TAGS	N	BOOLEAN	
ATTRIBUTE_SET_ITEM ²⁹	Y	XML	Element of the attribute set

IMPORT VIA ID

ID	J	INT	Unique ID of the node
KEY	N	CHAR(50)	Import key of the node
NAME	J	CHAR(200)	Name of the node

IMPORT VIA KEY

KEY	Y	CHAR(50)	Import key of the node
NAME	Y	CHAR(200)	Name of the node

IMPORT VIA NAME

NAME	Y	CHAR(200)	Name of the node
------	---	-----------	------------------

Sample XML

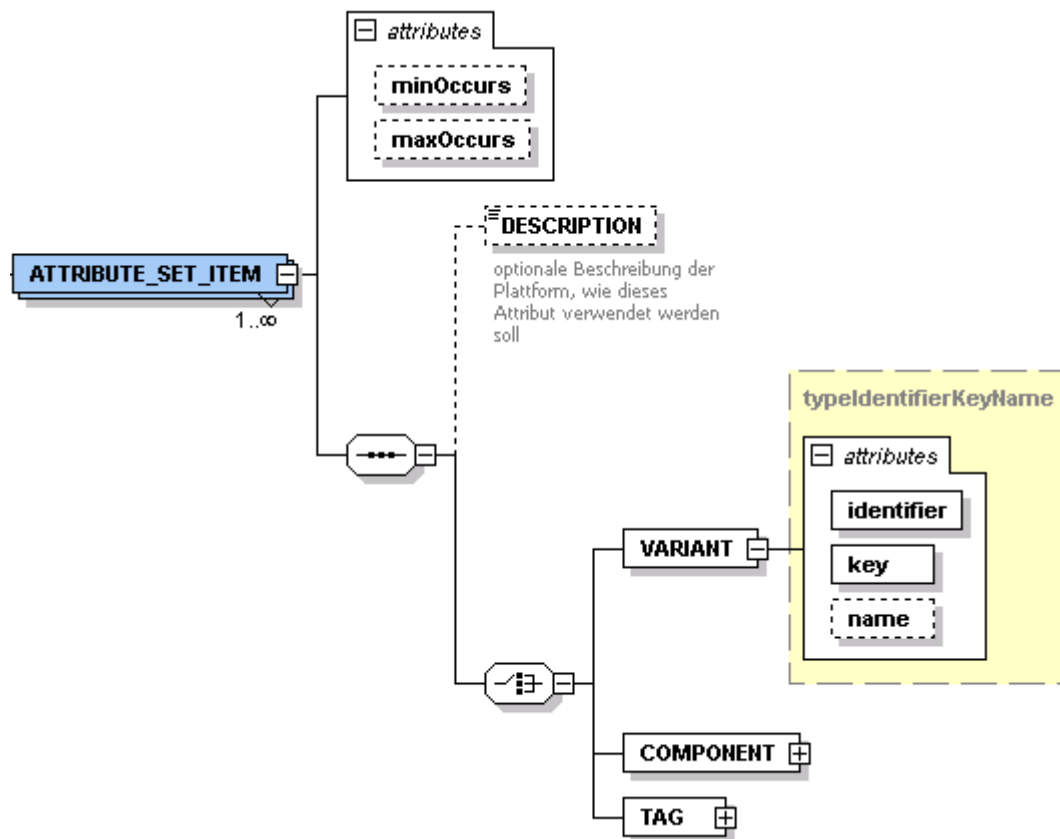
```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true" />
  <ATTRIBUTE_SET global="false" />
  <ATTRIBUTE_SET />
</ATTRIBUTE_SETS>

```

<!--corresponds to "global=false" -->

5.9 TBCATALOG / ... / ATTRIBUTE_SETS / ATTRIBUTE_SET_ITEM



NODE	MUST	DATA TYPE	DESCRIPTION									
ATTRIBUTE_SET_ITEM	Y	XML	Element of an attribute set									
ATTRIBUTE_SET_ITEM [min/maxOccurs]	N	INT	Defines the frequency of occurrence of the element defined below. Standard for both elements "1", i.e. the element is a mandatory input. <table><tr><th>MIN</th><th>MAX</th><th>DESCRIPTION</th></tr><tr><td>0</td><td>1</td><td>Need not be set but must not be used more than once</td></tr><tr><td>1</td><td>1</td><td>Mandatory value</td></tr></table>	MIN	MAX	DESCRIPTION	0	1	Need not be set but must not be used more than once	1	1	Mandatory value
MIN	MAX	DESCRIPTION										
0	1	Need not be set but must not be used more than once										
1	1	Mandatory value										

NODE	MUST	DATA TYPE	DESCRIPTION									
			<table><tr><th>MIN</th><th>MAX</th><th>DESCRIPTION</th></tr><tr><td>1</td><td>unbounded</td><td>Mandatory value that can be used multiple times</td></tr><tr><td>0</td><td>unbounded</td><td>Optional value that may be used multiple times</td></tr></table>	MIN	MAX	DESCRIPTION	1	unbounded	Mandatory value that can be used multiple times	0	unbounded	Optional value that may be used multiple times
MIN	MAX	DESCRIPTION										
1	unbounded	Mandatory value that can be used multiple times										
0	unbounded	Optional value that may be used multiple times										
ATTRIBUTE_SET_ITEM [overridePosition]	N	STRING	Determines at which element of the product a property (attributes/components) this attribute set is exported (product/article).Possible values are: <table><tr><th>VALUE</th><th>DESCRIPTION</th></tr><tr><td>no</td><td>Export as indicated by the brand/merchant</td></tr><tr><td>article</td><td>Export at article level</td></tr><tr><td>product</td><td>Export at product level</td></tr></table>	VALUE	DESCRIPTION	no	Export as indicated by the brand/merchant	article	Export at article level	product	Export at product level	
VALUE	DESCRIPTION											
no	Export as indicated by the brand/merchant											
article	Export at article level											
product	Export at product level											
DESCRIPTION	N	TEXT	Optional description of how this attribute shall be used. Shown in TB.One in the value mapping as hint.									
VARIANT/COMPONENT/TAG	Y	XML	Assignment of a variant dimension, component or attribute to the attribute set									

NODE	MUST	DATA TYPE	DESCRIPTION
COMPONENT_LIST	N	XML	For variants and components you may reference to a value list defined under COMPONENT_LISTS which is valid within this attribute set.
VARIANT[identifier] COMPONENT[identifier] TAG[identifier]	Y	ENUM	Identifier for referencing ¹⁰ (key, name)
VARIANT[key] COMPONENT[key] TAG[key]	Y	CHAR(50)	Value stored in the identifier assignment statement
VARIANT[name] COMPONENT[name] TAG[name]	N	CHAR(100)	Name stored in the identifier assignment statement

Sample XML (key-based)

```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET>
    <NAME>Blumenmischung</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">
      <COMPONENT identifier="key" key="farbe"/>
    </ATTRIBUTE_SET_ITEM>
  </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>

```

Sample XML (name-based)

```

<ATTRIBUTE_SETS>
  <ATTRIBUTE_SET global="true">
    <NAME>Blumenmischung</NAME>
    <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
  </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>

```

```
<ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>

<ALLOW_OTHER_TAGS>true</ALLOW_OTHER_TAGS>

<ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="unbounded">

    <DESCRIPTION>Die ERP Farbe kann von der tatsächlichen Farbe
abweichen.</DESCRIPTION>

    <COMPONENT identifier="name" key="ERP Farbe"/>

    <COMPONENT_LIST identifier="name" key="ERP Farben">

</ATTRIBUTE_SET_ITEM>

</ATTRIBUTE_SET>

</ATTRIBUTE_SETS>
```

6. EXAMPLE CLASSIFICATION (FLAT STRUCTURE)

```
<?xml version="1.0" encoding="UTF-8"?>
<TBCATALOG>
  <CLASSIFICATION type="replace">
    <STRUCTURE>
      <STRUCTURE_ITEM>
        <ID>200</ID>
        <KEY>heshirt</KEY>
        <NAME>Herren | Shirts & Pullover | Shirts</NAME>
        <PARENT_ID>0</PARENT_ID>
        <SORT>0</SORT>
        <ATTRIBUTE_SET identifier="name" key="Farbe-IntGroesse"/>
      </STRUCTURE_ITEM>
      <STRUCTURE_ITEM>
        <ID>210</ID>
        <KEY>hepull</KEY>
        <NAME>Herren | Shirts & Pullover | Pullover</NAME>
        <PARENT_ID>0</PARENT_ID>
        <SORT>0</SORT>
        <ATTRIBUTE_SET identifier="name" key="Farbe-IntGroesse"/>
      </STRUCTURE_ITEM>
    </STRUCTURE>
    <COMPONENTS>
      <COMPONENT>
        <KEY>color</KEY>
        <NAME>Farbe</NAME>
      </COMPONENT>
    </COMPONENTS>
    <TAGS>
      <TAG>
        <KEY>IntSize</KEY>
        <NAME>Internationale Größen</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>supjam</KEY>
        <NAME>Superjackenmeister</NAME>
    </BRAND>
    <BRAND>
        <KEY>hofre</KEY>
        <NAME>Hosenfreund</NAME>
    </BRAND>
</BRANDS>
<ATTRIBUTE_SETS>
    <ATTRIBUTE_SET global="true">
        <NAME>Farbe-IntGroesse</NAME>
        <ALLOW_OTHER_VARIANTS>true</ALLOW_OTHER_VARIANTS>
        <ALLOW_OTHER_COMPONENTS>true</ALLOW_OTHER_COMPONENTS>
        <ALLOW_OTHER_TAGS>true</ALLOW_OTHER_TAGS>
    </ATTRIBUTE_SET>
</ATTRIBUTE_SETS>

```

```
<ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
    <TAG identifier="name" key="Internationale Größen"/>
</ATTRIBUTE_SET_ITEM>
<ATTRIBUTE_SET_ITEM minOccurs="1" maxOccurs="1">
    <COMPONENT identifier="key" key="color"/>
</ATTRIBUTE_SET_ITEM>
</ATTRIBUTE_SET>
</ATTRIBUTE_SETS>
</CLASSIFICATION>
</TBCATALOG>
```