Date: 01.04.2023

Changelog MODIFY-IDM Kitchen/Bath Version 3.0.1

This changelog describes all innovations, enhancements and corrections that are made available with the current version **MODIFY-IDM 3.0.1** (compared to version MODIFY-IDM 2.8.1) both in the XML schema and in the documentation.

Version MODIFY-IDM 3.0.1 is published on 01.04.2023 and becomes valid from 01.07.2023.

Content:

			Decision of: *(date)*	Type	Page
Released	**1.**	**Catalogue identification and versioning**			2
	1.1.	New element CATALOG_ID under CATALOG_IDENTIFICATION	2021-09-29	A	2
	1.2.	CATALOG_YEAR under CATALOG is deleted	2021-09-29	R	3
	1.3.	Move MAJOR_NO and FILE_RELEASE_DATE to CATALOG_IDENTIFICATION	2021-09-29	C	4
	1.4.	New element MINOR_NO under CATALOG_IDENTIFICATION	2021-09-29	A	5
	1.5.	CATALOG_NUMBER below CATALOG is deleted	2021-09-29	R	6
	2.	**Classification according to ECLASS**			7
	2.1.	New optional complex Type CLASSIFICATION under CATALOGUE	2023-March	A	7
	2.2.	New optional complex Type CLASSIFICATION under SERIES	2023-March	A	9
	2.3.	New optional complex Type CLASSIFICATION under ITEM	2023-March	A	11
	2.4.	New element ECLASS_VALUE under FEATURE	2023-March	A	13
	2.5.	New element ECLASS_VALUE under OPTION	2023-March	A	14
	3.	**Regular expression for key elements**			15
	3.1.	Changed pattern in the TYPE_NO element under ITEM	2022-03-22	C	15
	4.	**Textual changes in the documentation**			16
	4.1.	Extended descriptions of the variant types according to CSV documentation	2022-03-22	C	16
	4.2.	Explanatory example for multi-language catalogues (e.g. Switzerland) on the elements COUNTRY_ID and ISO_LANGUAGE_ID	2022-03-22	C	17
	4.3.	Note on handling special colours in the OPTION element	2022-03-22	C	18
	4.4.	Corrected description of the data type gYear	2023-03-27	F	19

A = Added = Adding new elements or attributes
C = Changed = Changes to existing elements, attributes or descriptive texts.
R = Removed = Deletion of elements or attributes
F = Fixed = Error corrections to existing elements, attributes or descriptive texts.

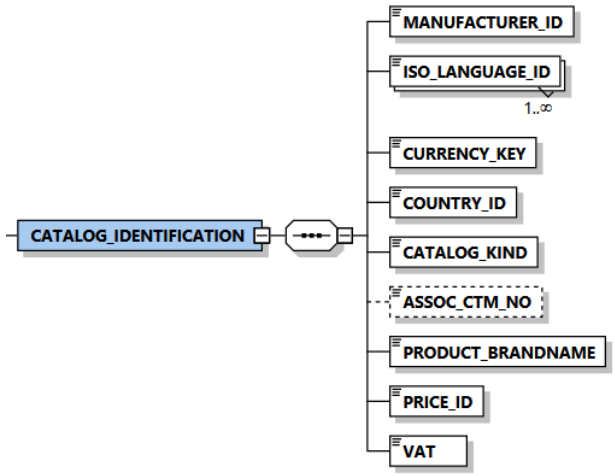
Released

Display of all changes recorded for version 3.0.1

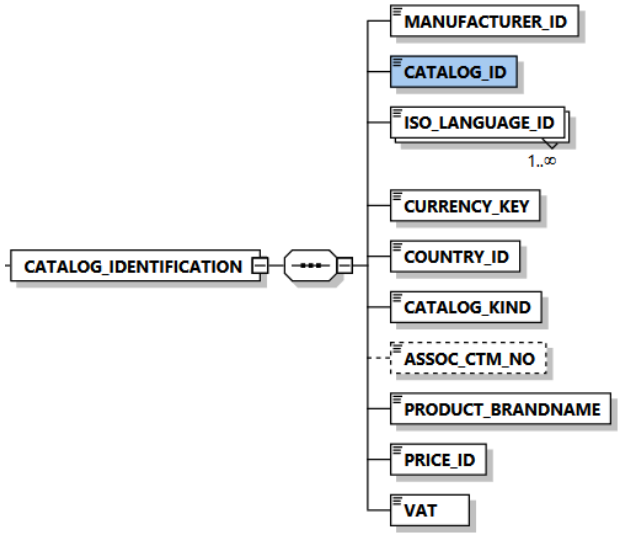
1. Catalogue identification and versioning

1.1.	A	New element CATALOG_ID under CATALOG_IDENTIFICATION	Decision : 2021-09-29

2.8.1



3.0.1



The element CATALOG_ID under CATALOG_IDENTIFICATION is a mandatory field with max. 64 characters and uniquely identifies the catalogue together with the MANUFACTURER_ID.

Description in the documentation:

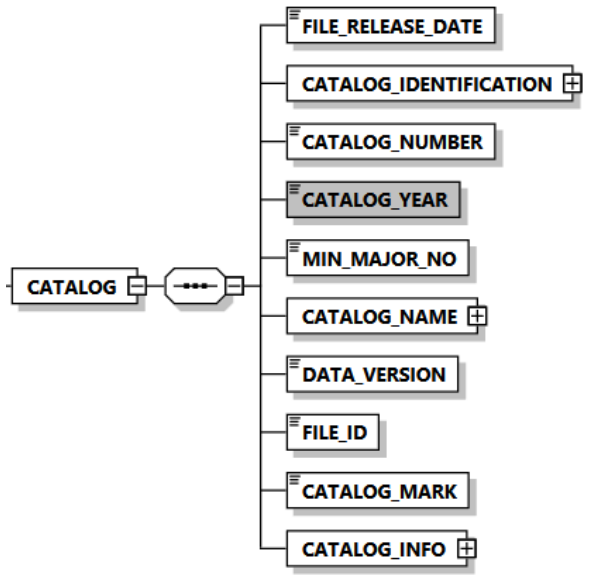
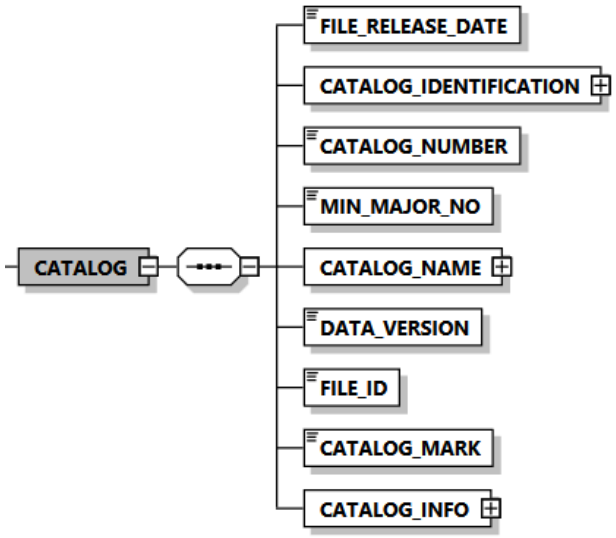
The manufacturer must enter a unique ID (e.g. GUID) that unequivocally identifies the catalogue here.

A manufacturer catalogue must always have the same ID for the same dealership.

```
<xs:element name="CATALOG_ID">
    <xs:annotation>...</xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <xs:maxLength value="64"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

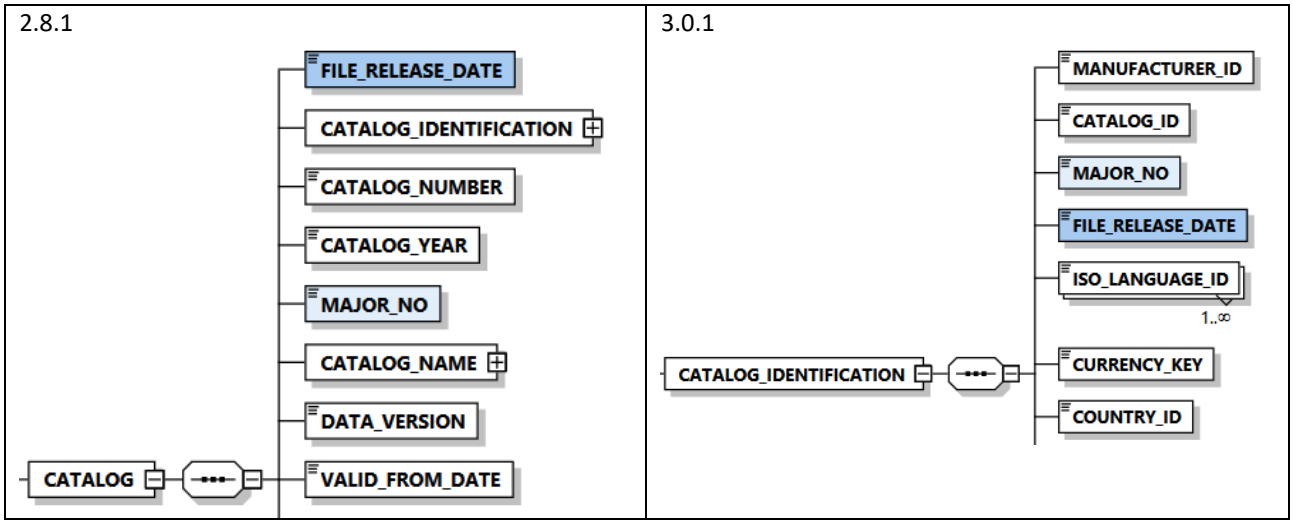
A = Added = Adding new elements or attributes
C = Changed = Changes to existing elements, attributes or descriptive texts.
R = Removed = Deletion of elements or attributes
F = Fixed = Error corrections to existing elements, attributes or descriptive texts.

1.2. R CATALOG_YEAR under CATALOG is deleted Decision : 2021-09-29

2.8.1	3.0.1
	

The element CATALOG_YEAR under CATALOG is deleted. It is replaced by the element MAJOR_NO under CATALOG_IDENTIFICATION.

1.3. C Move MAJOR_NO and FILE_RELEASE_DATE to CATALOG_IDENTIFICATION — Decision : 2021-09-29



The elements MAJOR_NO and FILE_RELEASE_DATE version the catalogue uniquely together with the new element MINOR_NO.

The FILE_RELEASE_DATE remains unchanged.

MAJOR_NO:

The existing element MAJOR_NO is changed in its function. It now indicates the year of entry of the main catalogue.

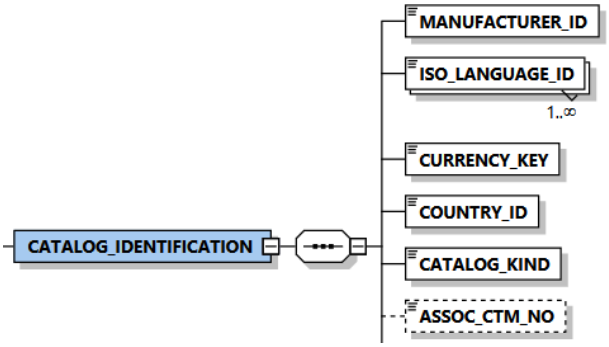
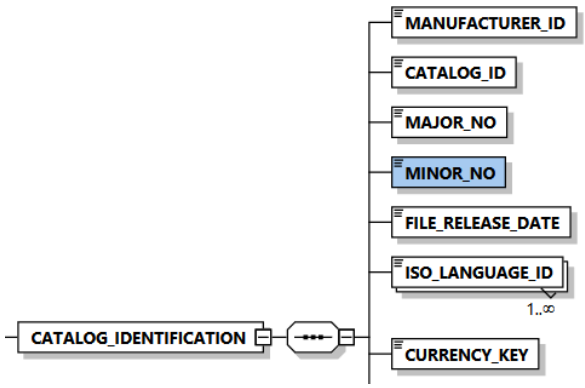
Description in the documentation:

The catalogue version number MAJOR_NO indicates the year in which the main catalogue was created. The complete specification of the catalogue version in which a dataset is delivered consists of MAJOR and MINOR.

```
<xs:element name="MAJOR_NO">
    <xs:annotation>...</xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:gYear"/>
    </xs:simpleType>
</xs:element>
<xs:element name="MINOR_NO">...</xs:element>
<xs:element name="FILE_RELEASE_DATE" type="xs:dateTime">
    <xs:annotation>...</xs:annotation>
</xs:element>
```

A = Added = Adding new elements or attributes
C = Changed = Changes to existing elements, attributes or descriptive texts.
R = Removed = Deletion of elements or attributes
F = Fixed = Error corrections to existing elements, attributes or descriptive texts.

1.4. A New element MINOR_NO under CATALOG_IDENTIFICATION Decision : 2021-09-29

2.8.1	3.0.1
	

The element MINOR_NO is a mandatory field and contains consecutive numbers as a sub-version number to each MAJOR_NO.

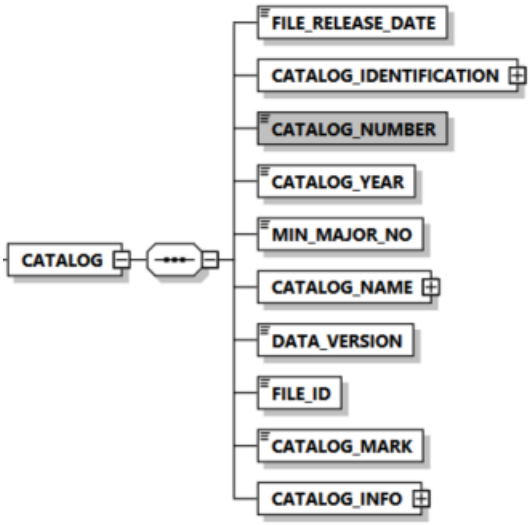
Description in the documentation:

The sub-catalog version number MINOR_NO is given in consecutive numbers during the year or during the period of validity of the main catalog. The complete specification of the catalog version in which a dataset is delivered consists of MAJOR and MINOR. If the catalog version MAJOR is changed, the sub-catalog version is reset to 0.

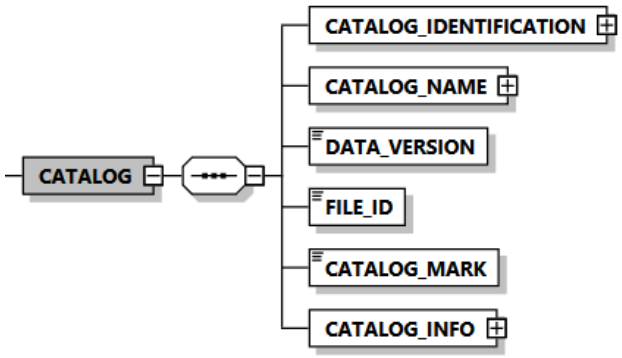
```
<xs:element name="MINOR_NO">
    <xs:annotation>...</xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:nonNegativeInteger">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="999"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

1.5. R CATALOG_NUMBER below CATALOG is deleted Decision : 2021-09-29

2.8.1



3.0.1

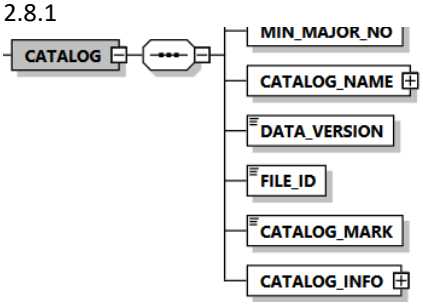


The element CATALOG_NUMBER under CATALOG is deleted.

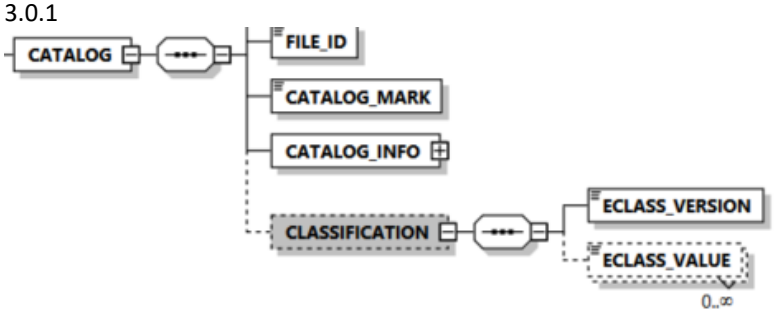
2. Classification according to ECLASS

2.1. A New optional complex Type CLASSIFICATION under CATALOGUE Decision : 2023-March

2.8.1



3.0.1



As the only classification scheme in the future, ECLASS will be maintained at various nodes in the IDM format. At the catalogue level, the ECLASS version valid for the entire catalogue is stored, as well as ECLASS properties that apply to all items in the catalogue.

CLASSIFICATION:

The optional element CLASSIFICATION below CATALOG is a complexType.

Description in the documentation:

This element is used to store the classification information at catalog level.

If the catalog contains only one classification, the ECLASS_VERSION must be specified.

Information at lower levels for the same properties overwrites the value specified at catalog level.

ECLASS_VERSION:

The element ECLASS_VERSION below CLASSIFICATION is of type integer and mandatory.

Description in the documentation:

The classification version is stored in this element.

Only the major no. of the version is specified.

ECLASS_VALUE:

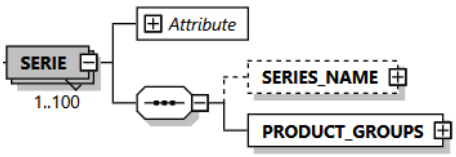
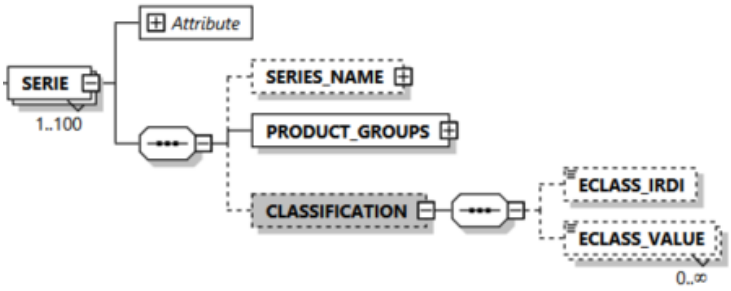
The optional element ECLASS_VALUE below CLASSIFICATION is of type string and can be created as often as desired.

Description in the documentation:

The values of the ECLASS property are stored in this element. For this purpose, the path along the multi-level property structure is specified. The underscore (_) serves as a separator.

```
<xs:element name="CLASSIFICATION" minOccurs="0">
    <xs:annotation>...</xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ECLASS_VERSION">
                <xs:annotation>...</xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:pattern value=""/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="ECLASS_VALUE" type="xs:string" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>...</xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

2.2. A New optional complex Type CLASSIFICATION under SERIES Decision : 2023-March

2.8.1	3.0.1
	

As the only classification scheme in the future, ECLASS is maintained at various nodes in the IDM format. The ECLASS class and any number of ECLASS properties, which apply to all items in the series, are stored on the serie.

CLASSIFICATION:

The optional element CLASSIFICATION below SERIES is a complexType.

Description in the documentation:

This element can be used to store classification information at series level.

Specifications at lower levels for the class or the same properties overwrites the value specified at series level.

ECLASS_IRDI:

The optional element ECLASS_IRDI below CLASSIFICATION is of type string and optional. The pattern restricts to the IRDI values possible for classes.

Description in the documentation:

This element stores the IRDI of the 4th level of the hierarchical ECLASS class structure.

ECLASS_VALUE:

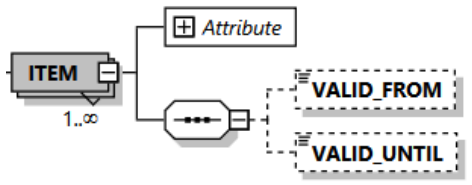
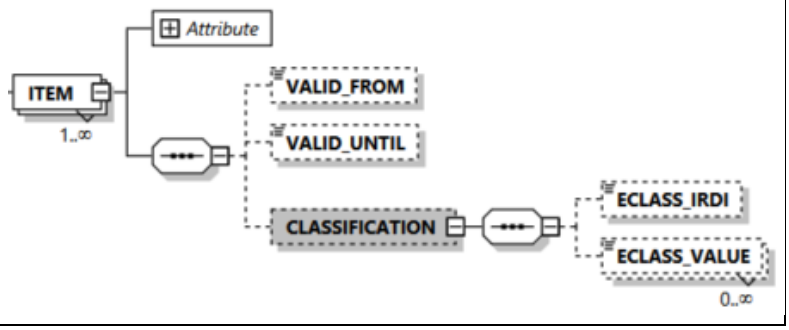
The optional element ECLASS_VALUE below CLASSIFICATION is of type string and can be created as often as desired.

Description in the documentation:

The values of the ECLASS property are stored in this element. For this purpose, the path along the multi-level property structure is specified. The underscore (_) serves as a separator.

```
<xs:element name="CLASSIFICATION" minOccurs="0">
    <xs:annotation>...</xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ECLASS_IRDI" minOccurs="0">
                <xs:annotation>...</xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="0173-1#01-[A-Z]{3}[0-9]{3}#[0-9]{3}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="ECLASS_VALUE" type="xs:string" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>...</xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

2.3. A New elements under CLASSIFICATION below ITEM Decision : 2023-March

2.8.1	3.0.1
	

As the only classification scheme in the future, ECLASS will be maintained at various nodes in IDM format. The ECLASS class and any number of ECLASS properties are stored on the item.

CLASSIFICATION:

The optional element CLASSIFICATION below ITEM is a complexType.

Description in the documentation:

In this element, the classification information and categorisations are stored at item level.

Information on lower levels for the same ECLASS properties overwrites the value specified at item level.

ECLASS_IRDI:

The optional element ECLASS_IRDI below CLASSIFICATION is of type string. The pattern restricts to the IRDI values possible for classes.

Description in the documentation:

This element stores the IRDI of the 4th level of the hierarchical ECLASS class structure.

ECLASS_VALUE:

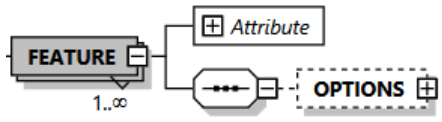
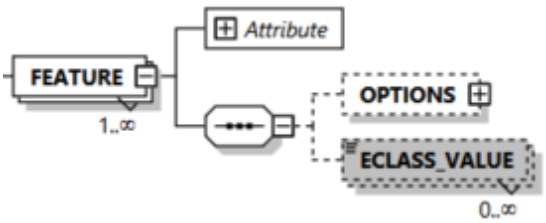
The optional element ECLASS_VALUE below CLASSIFICATION is of type string and can be created as often as desired.

Description in the documentation:

The values of the ECLASS property are stored in this element. For this purpose, the path along the multi-level property structure is specified. The underscore (_) serves as a separator.

```
<xs:element name="CLASSIFICATION" minOccurs="0">
    <xs:annotation>...</xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ECLASS_IRDI" minOccurs="0">
                <xs:annotation>...</xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="0173-1#01-[A-Z]{3}[0-9]{3}#[0-9]{3}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="ECLASS_VALUE" type="xs:string" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>...</xs:annotation>
            </xs:element>
            <xs:element name="OTHER_CATEGORISATION" minOccurs="0" maxOccurs="5">...</xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

2.4. A New element ECLASS_VALUE under FEATURE Decision : 2023-March

2.8.1	3.0.1
	

As the only classification scheme in the future, ECLASS is maintained at various nodes in IDM format. Any number of ECLASS properties are stored at the feature.

ECLASS_VALUE:

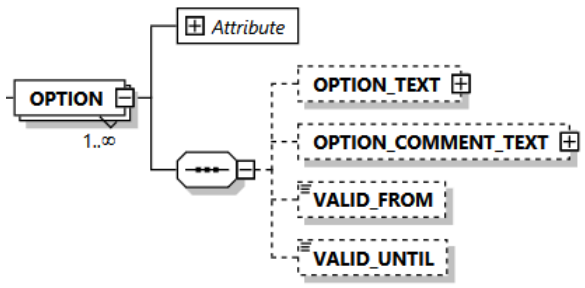
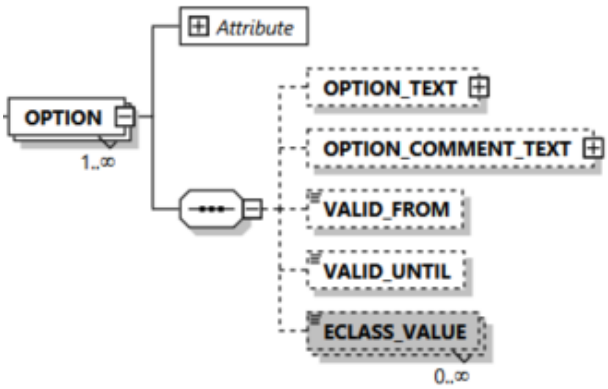
The optional element ECLASS_VALUE below CLASSIFICATION is of type string and can be created as often as desired.

Description in the documentation:

The values of the ECLASS property are stored in this element. For this purpose, the path along the multi-level property structure is specified. The underscore (_) serves as a separator.

```
<xs:element name="ECLASS_VALUE" type="xs:string" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>...</xs:annotation>
</xs:element>
```

2.5. A New element ECLASS_VALUE under OPTION Decision : 2023-March

2.8.1	3.0.1
	

As the only classification scheme in the future, ECLASS is maintained at various nodes in IDM format. Any number of ECLASS properties can be stored at the option.

ECLASS_VALUE:

The optional element ECLASS_VALUE below CLASSIFICATION is of type string and can be created as often as desired.

Description in the documentation:

The values of the ECLASS property are stored in this element. For this purpose, the path along the multi-level property structure is specified. The underscore (_) serves as a separator.

```
<xs:element name="ECLASS_VALUE" type="xs:string" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>...</xs:annotation>
</xs:element>
```

3. Regular expression for key elements

3.1.	C	Changed pattern in the TYPE_NO element under ITEM	Decision : 2022-03-22

2.8.1

```
<xs:element name="ITEM" maxOccurs="unbounded">
    <xs:annotation>...</xs:annotation>
    <xs:complexType>
        <xs:sequence>...</xs:sequence>
        <xs:attribute name="SERIE_NO" use="required">...</xs:attribute>
        <xs:attribute name="TYPE_NO" use="required">
            <xs:annotation>...</xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength value="1"/>
                    <xs:maxLength value="15"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
```

3.0.1

```
<xs:element name="ITEM" maxOccurs="unbounded">
    <xs:annotation>...</xs:annotation>
    <xs:complexType>
        <xs:sequence>...</xs:sequence>
        <xs:attribute name="SERIE_NO" use="required">...</xs:attribute>
        <xs:attribute name="TYPE_NO" use="required">
            <xs:annotation>...</xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:minLength value="1"/>
                    <xs:maxLength value="15"/>
                    <xs:pattern value="\S(.{0,28}\S)?"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
```

To minimise erroneous type numbers, the new pattern in the TYPE_NO element under ITEM excludes spaces at the beginning and end.

4. Textual changes in the documentation

4.1.	C	Extended descriptions of the variant types according to CSV documentation	Decision : 2022-03-22

2.8.1

Variant type	Variant text	Execution type	Comment
105	Front combination	K	If only one key is used for different front information.
201	Carcase combination	K	If only one key is used for several – interior /exterior - carcase colours.
303	Handle combination	K	If only one key is given for several handle conditions.
999	Fixed material assignment	F	

3.0.1

Variant type	Variant text	Execution type	Comment
105	Front combination	K	If only one key is given for several front information. If variant type 105 is filled, it overrides variant types 100-104 to the extent that they are only for internal use by the software companies (e.g. internal colour control). These are then set as invisible header variants ("U") in the element HEADER_POS_VARIATION_TYPE. For external use as a selection for the user (e.g. query as header and order information), only the elements under EDI_INFO are used.
201	Carcase combination	K	If there is only one key for several carcase colours - inside/outside. If variant type 201 is filled, it overrides variant types 200, 202-205 to the extent that they are only for internal use by the software companies (e.g. internal colour control). These are then set as invisible header variants ("U") in the element HEADER_POS_VARIATION_TYPE. For external use as a selection for the user (e.g. query as header and order information), only the elements under EDI_INFO are used.
303	Handle combination	K	If only one key is given for several handle conditions. If variant type 303 is filled, it overrides variant types 300-302 to the extent that they are only for internal use by the software companies (e.g. internal colour control). These are then set as invisible header variants ("U") in the element HEADER_POS_VARIATION_TYPE. For external use as a selection for the user (e.g. query as header and order information), only the elements under EDI_INFO are used.
999	Fixed material assignment	F	This variant type is defined globally, but may only be used in the material properties.

It was noticed that the descriptions of 4 variant types were only partially transferred during the transformation of the CSV into the XML documentation. Missing information has now been added again.

A = Added = Adding new elements or attributes
C = Changed = Changes to existing elements, attributes or descriptive texts.
R = Removed = Deletion of elements or attributes
F = Fixed = Error corrections to existing elements, attributes or descriptive texts.

4.2.	C	Explanatory example for multi-language catalogues (e.g. Switzerland) on the elements COUNTRY_ID and ISO_LANGUAGE_ID	Decision : 2022-03-22

ISA_LANGUAGE_ID:

2.8.1 The language code defines the language in which the texts in the data pool were created. The ISO language code must be used. Multiple language codes indicate that the catalogue is used in multiple languages. The catalogue texts must only use the language codes entered in these elements.
3.0.1 The language key defines the language in which the texts of the dataset were created. The ISO language key is to be used for this purpose. In the case of multilingual catalogues, all languages occurring in the catalogue must be entered here. Only the language keys entered in these elements may be used for the texts of the catalogue. For example, in a Swiss catalogue, the ISO_LANGUAGE_ID's DE (German), FR (French) and IT (Italian) can be stored here.

COUNTRY_ID:

2.8.1 This element is used to specify the correct country allocation where a language is spoken in multiple countries and the identical currency is specified. This requires entry of the ISO country code.
3.0.1 The ISO country code of the target market is entered in this element. For countries with several languages but the same currency, make sure to use the correct ISO country code. For example, a Swiss catalogue may contain the 3 ISO_LANGUAGE_ID's DE (German), FR (French) and IT (Italian). However, the COUNTRY_ID must be indicated as CH (Switzerland).

To avoid misleading information in the COUNTY_ID and ISO_LANGUAGE_ID elements, an explanatory example has been added to the description of both elements.

4.3. C Note on handling special colours in the OPTION element Decision : 2022-03-22

2.8.1 This element defines the features of a feature class. All features of the complete database are defined in this element. In case of a finish option (cf. table standard variations, flag "A") neither texture (TEXTUR)nor colour values (COLOR) are permitted. The assignment of brightness and transparency is optional. In case they are defined they have priority over the colour variations.
3.0.1 In this element, the options for a feature are defined and all options to be modified in the dataset are stored. If these are special colours, separate OPTIONs must be created for each colour scheme (e.g. RAL, NCS, etc...). (see also: Best Practices/Special Colours)

In order to avoid incorrect handling of spot colours in the future, the description in the OPTION element has been supplemented and a best practice has been added to the documentation.

Excerpt Best Practice:

Special colours

Not all possible spot colours can be created individually in an IDM catalogue.

Therefore, it is necessary to define a structure that enables the planning systems to show the applicable spot colours.

For this purpose, a variant is to be created for each standardised colour scheme, e.g. RAL, NCS, SIKKENS, etc.

Example 1:

The supplier offers a desired colour in the RAL and NCS colour schemes for various design elements.

Correct:

```
<OPTION FEATURE_NO="101" OPTION_KEY="NCS">
  <OPTION_TEXT>
    <LANGUAGE ISO_LANGUAGE_ID="DE">
      <TEXT>
        NCS-Lackierung
  <OPTION_COMMENT_TEXT>
  <MINIMUM_DELIVERY_TIME>
<OPTION FEATURE_NO="101" OPTION_KEY="RAL">
  <OPTION_TEXT>
    <LANGUAGE ISO_LANGUAGE_ID="DE">
      <TEXT>
        RAL-Lackierung
  <OPTION_COMMENT_TEXT>
  <MINIMUM_DELIVERY_TIME>
```

Wrong:

```
<OPTION FEATURE_NO="501" OPTION_KEY="MSLS">
  <OPTION_TEXT>
    <LANGUAGE ISO_LANGUAGE_ID="DE">
      <TEXT>
        RAL/NCS-Farben
```

[...]

4.4. F Corrected description of the data type gYear Decision : 2023-03-27

2.8.1

Data types and value ranges

[...]

gYear	In the IDM, the data type dateTime is used in conformity with the XML schemas as follows: YYYY

3.0.1

Data types and value ranges

[...]

gYear	In the IDM, the data type gYear is used in conformity with the XML schemas as follows: YYYY

Contact

Data Competence Center e. V.

Goebenstraße 4-10

32052 Herford

Info: www.dcc-moebel.org

Dr.-Ing. Olaf Plümer

E-mail: pluemer@dcc-moebel.org

Phone +49 52 21 / 12 65-37