Technical Specification 101

ECLASS in BMEcat

Aug 15, 2023

Authors and contributors

Dr. Christian Block, ECLASS Head Office

Stefan Mülhens, AmpereSoft GmbH

Matthias Redecker, Weidmüller Interface GmbH & Co. KG

Dr. Matthias Richter, Paradine GmbH

Volker Römisch, Noxum GmbH

Frank Scherenschlich, Class.Ing Ingenieur-Partnerschaft

Josef Schmelter, PHOENIX CONTACT GmbH & Co. KG

Please send remarks to info@eclass.eu

Revision History

Date	Version	Changes
Aug 15, 2023	1.0	▪ Merge of available documents ▪ Include BMEcat 2005.2

This document replaces:

- https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-BMEcat-Guideline-1.2_v1_0.pdf
- https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-BMEcat-Guideline-2005_v1_1.pdf
- https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-BMEcat-Guideline-2005_1_v2_1.pdf

Table of contents

1 Introduction

The data exchange of product data is an important component in the communication with business partners. Standards play an important role for smooth exchange. These include catalog standards such as BMEcat and the master data standard ECLASS. The aim of this document is to define rules on how to apply ECLASS in the BMEcat exchange format. For this purpose, ECLASS in 1.1, as well as BMEcat in 1.2, are first briefly described. In 1.3 the scope of this document is clearly shown.

1.1 ECLASS

The objective of ECLASS is to simplify the electronic, cross-industry data exchange through the classification of standardized product descriptions. The central unique characteristic of ECLASS is the possibility of providing an unambiguous, language-neutral, machine-readable, and industry-independent description of products and services and their characteristics. Currently, there are about 45,000 product classes, that are organized into four levels. On the fourth level, a set of Properties is assigned. The set of Properties are generated as a set of the 19,000 well-defined Properties of the ECLASS Dictionary. A large part of the goods and services traded worldwide is represented in ECLASS. In addition to Classes and Properties, further element types and modelling features exist in the ECLASS Dictionary. The Dictionary, which is exchanged in different ways like CSV, XML or JSON is available in two representations ECLASS Basic and ECLASS Advanced.

ECLASS uses globally unique identifiers for every Structure Element included in the ECLASS Standard. This globally unique identifier is called IRDI (International Registration Data Identifier). These identifiers are used to ensure that the semantic of an element is unique in the overall system. The IRDI is based on the international standards ISO/IEC 11179-6, ISO 29002, and ISO 6532.

More information on ECLASS can be found on www.eclass.eu.

1.2 BMEcat

The ECLASS Standard itself defines the structure and semantics of products. To exchange product data a specific exchange format is needed. The BMEcat is such a format. BMEcat is a freely available and XML-based standard for electronic data transfer by electronic catalogues created and published by the BME (Bundesverband Materialwirtschaft, Einkauf und Logistik e. V., the German association for materials management, purchasing and logistics). In contrast to the ECLASS Dictionary which describes how things can be characterized at an abstract level, it is about actual instances of Application Classes that are described by distinct values in accordance with the Dictionary.

More information on BMEcat can be found at https://www.bme.de/services/bmecat/ .

1.3 Scope

The objective of this document is the definition of a common application of ECLASS content in the context of BMEcat catalogs. Although BMEcat is available in four different versions 1.2, 2005, 2005.1 and the last update 2005.2 this document is a generic specification. This document is a holistic view of all ECLASS modelling options (including both ECLASS representations Advanced and Basic) and their application in BMEcat 2005.2. If something is not possible or needs to be mapped differently, version dependent specifications are marked.

In chapter 2 general BMEcat specific rules are defined. Chapter 3 describes the concrete usage of ECLASS Structure Elements in BMEcat Transaction Type T_NEW_CATALOG, which is the focus of this document.

Out of scope are the transactions T_UPDATE_PRODUCTS and T_UPDATE_PRICES as well as other feature group systems, be they external or transported in the catalog. Furthermore, UDX (user defined extensions) are not described here.

The following basic assumptions are made in this document:

- The ECLASS Dictionary is available separately from the BMEcat catalog, i.e., ECLASS is not to be transported inside the catalog
- There is a process in place that guarantees that catalog requirements can be expressed in a machine tractable way
- Concept identifiers are used to identify elements from the Dictionary
- The Dictionary applies Dictionary change management
- It is recommended to use the version 2005.2 or to upgrade to the version

2 General Definitions to Use ECLASS in BMEcat

In this chapter general BMEcat specific rules and prerequisites are defined to use ECLASS as a Dictionary in BMEcat.

2.1 BMEcat Versions

This section describes the most important changes between the BMEcat versions. The official documentations and schema files can be found online on https://www.bme.de/services/bmecat/downloads_BMEcat. In the following XML schema and documentation URLs point to files on ECLASS servers. The schema and documentation files were copied to provide a consistent set for implementation. The original references are given in the footnotes.

2.1.1 BMEcat 1.2

BMEcat 1.2 is the initial version with a specific ECLASS integration. For more information see:

- Description: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_1.2/bmecatv12e.pdf[1]
- XSD: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_1.2/bmecat_12_xsd_all_in_one.zip[2]

2.1.2 BMEcat 2005

The next evolution step was the version 2005. For more information see:

- Description: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_2005/bmecat_2005_en.pdf[3]

[1] https://a.storyblok.com/f/104752/x/191681679f/bmecatv12e.pdf
[2] https://a.storyblok.com/f/104752/x/b7ed0193f6/bmecat_12_xsd_all_in_one.zip
[3] https://a.storyblok.com/f/104752/x/bef34fe816/bmecat_2005_en.pdf

- XSD: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_2005/bmecat_2005_xsd.zip[4]

2.1.3 BMEcat 2005.1

BMEcat 2005.1 was developed in a cooperation between BME e.V. and ECLASS e.V.. Goals for the BMEcat 2005.1 format were to improve the multilingual capabilities of BMEcat as well as to support products according to the ECLASS Advanced model. This version is fully backward compatible with BMEcat 2005, i.e., all changes in BMEcat 2005.1 are optional extensions to BMEcat 2005.

Regarding the use of ECLASS Advanced the following schema extensions were done in 2005.1:

- Support of nested structures in ECLASS (Blocks, Cardinality, Polymorphism)

To use nested structures the BMEcat 2005 schema of FEATURE was extended.

```
<xsd:element ref="FID" minOccurs="0"/>
<xsd:element ref="FPARENT_ID" minOccurs="0"/>
<xsd:element ref="FEATURE" minOccurs="0" maxOccurs="unbounded"/>
```

This allows two kinds of nesting. First, embed a FEATURE in a FEATURE, and secondly work with parent-child-references. Especially with regard to legacy systems, the parent-child-references has established itself as a de facto standard. Therefore, this approach is also used and recommended by ECLASS. The FID is a document dependent and freely assigned ID, which can be referenced in FPARENT_ID again. Although minOccurs of FPARENT_ID is null, the proposal is to set a “-1” for an unnested FEATURE without a parent.

- Support of ECLASS Aspects

To allow for direct representation of Aspects within BMEcat 2005.1 the element FEATURE_GROUP was added to the element PRODUCT_FEATURES as a container representing the Aspect, having an optional Preferred Name (FEATURE_GROUP_NAME), Definition (FEATURE_GROUP_DESCRIPTION) and IRDI (REFERENCE_FEATURE_GROUP_ID).

[4] https://a.storyblok.com/f/104752/x/6a10e4022d/bmecat_2005_xsd.zip

```
<xsd:element name="FEATURE_GROUP">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element ref="FEATURE_GROUP_NAME" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element ref="FEATURE_GROUP_DESCRIPTION" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element ref="REFERENCE_FEATURE_GROUP_ID"/>
         <xsd:element ref="FEATURE" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="featureGroupType" type="dtSTRING"/>
   </xsd:complexType>
</xsd:element>
```

- For the full XML schema see: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_2005.1/bmecat_2005_1-xsd.zip[5]
- For the full description see: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_2005.1/bmecat2005-1.pdf[6]

2.1.4 BMEcat 2005.2

BMEcat 2005.2 was developed in a cooperation between BME e.V. and ECLASS e.V.. Goals for the BMEcat 2005.2 format were to improve the transport of product data in a few details:

- Combination of FVALUE and VALUE_IDREF

In 2005.1 it was possible to transport several Values for a characteristic via FVALUE or VALUE_IDREF. However, it was not possible to combine FVALUE and VALUE_IDREF. That is the case if an ECLASS Property is defined with a Value List, but not all necessary Values are available. In ECLASS all Values are suggestions in general. Therefore the <xsd:choice> of FVALUE and VALUE_IDREF was extended backward compatible:

```
<xsd:choice>
   <xsd:choice maxOccurs="unbounded">
     <xsd:element ref="FVALUE" maxOccurs="unbounded"/>
     <xsd:element ref="VALUE_IDREF" maxOccurs="unbounded"/>
   </xsd:choice>
   <xsd:element ref="VARIANTS"/>
</xsd:choice>
```

- Length specifications of FT_NAME and FNAME

[5] https://a.storyblok.com/f/104752/x/063fbced32/bmecat_2005_1-xsd.zip
[6] https://a.storyblok.com/f/104752/x/ff75282c52/bmecat2005-1.pdf

Inside of a FEATURE the elements FT_NAME or FNAME should be used to transport the Preferred Name of the ECLASS Property (compare 3.2). However, the ECLASS Preferred Name can have a length of 80 characters (https://eclass.eu/support/technical-specification/structure-and-elements/property#c2039). Therefore, the length restriction was extended to 80:

```
<xsd:element name="FNAME">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:restriction base="dtMLSTRING">
               <xsd:maxLength value="80"/>
            <xsd:minLength value="1"/>
         </xsd:restriction>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
```

- Length specifications of FVALUE

Until 2005.1 it was only possible to transport Values with a maximal length of 60 characters. This makes the definition of longer Values problematic. Among other things, URLs are only possible with some difficulty. ECLASS does not define any length restrictions in principle.

Furthermore, it was necessary to define a Value of a minimum of one character. It may be necessary to transport empty Values. E.g. with multilingual catalogs, Values must be put in relation, if there is in addition a multivalent evaluation. An example is a multicolored product (red, yellow, green). It was only possible to specify via sequences, a second language the Value "ROT" belongs to “red”. But it is problematic if a Value is missing in a language. Then an empty element must be transferred. This was not possible in 2005.1.

Therefore, the length restriction was removed:

```
<xsd:element name="FVALUE">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:restriction base="dtMLSTRING">
         </xsd:restriction>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
```

- MIME-Types

The BMEcat specification 2005.1 did not include all possible Mime-Types. For instance, “application/dxf” was not allowed. To support all known Media Types from https://www.iana.org/assignments/media-types/media-types.xhtml and to keep the schema backward compatible the Regular Expression for Mime Types was changed to:

```
<xsd:element name="MIME_TYPE">
   <xsd:simpleType>
      <xsd:restriction base="dtSTRING">
         <xsd:maxLength value="100"/>
         <xsd:minLength value="1"/>
         <xsd:pattern value="[-+.\w]+(\/[-+.\w]+)(;\s?[-+.\w]+=(([-+.\w]+)|(&quot;[-
+.\w]+&quot;)))*|^url$"/>
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>
```

- For the full XML schema see: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_2005.2/bmecat_2005-2.xsd[7]
- For the full description see: https://eclass.eu/fileadmin/static/documents/wiki/BMEcat/BMEcat_2005.2/bmecat_2005-2_eng.pdf[8]

2.2 Selection of BMEcat Version

As stated in the chapters before, the two representations ECLASS Basic and ECLASS Advanced as well as the four different BMEcat versions 1.2, 2005, 2005.1 and 2005.2 exist. However, not every ECLASS representation fits to each BMEcat version. Therefore, the following Table 1 and Table 2 define a possible mapping.

[7] https://a.storyblok.com/f/104752/x/13da94f38b/bmecat_2005-2.xsd
[8] https://a.storyblok.com/f/104752/x/4080aacefd/bmecat_2005-2_eng.pdf

Table 1: Mapping of ECLASS Representation and BMEcat Version

	BMEcat 1.2	BMEcat 2005	BMEcat 2005.1	BMEcat 2005.2
ECLASS Basic	x	x	x	X
ECLASS Advanced (from ECLASS 7)			x	X

Table 2: Mapping of BMEcat Version and ECLASS Feature

	BMEcat 1.2	BMEcat 2005	BMEcat 2005.1	BMEcat 2005.2
ECLASS Classification Class	x	x	x	X
ECLASS Application Class			x	X
ECLASS Property	x	x	x	x (improved support for the preferred name)
ECLASS Value	x	x	x	x (improved Value List support)
ECLASS Aspect			x	X
ECLASS Block			x	X
ECLASS Cardinality			x	X
ECLASS Polymorphism			x	X
Level Type*	(x)	(x)	x	x (incl. empty Values)
Axis Type*	(x)	(x)	x	X

* Even if a general application of LevelType and AxisType is possible in all four BMEcat versions, these special types are only applied in ECLASS Advanced.

2.3 Namespace

BMEcat is a standardized and XML-based exchange format for catalog data. To be able to identify the different BMEcat versions used in the XML files, the following namespace convention is defined.

2.3.1 BMEcat 1.2

It is expected that the following start tag / namespace is used for BMEcat 1.2:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BMECAT SYSTEM "bmecat_new_catalog_1_2.dtd">
<BMECAT version="1.2" xmlns="http://www.bmecat.org/bmecat/1.2/bmecat_new_catalog">
```

2.3.2 BMEcat 2005

It is expected that the following start tag / namespace is used for BMEcat 2005:

```
<?xml version="1.0" encoding="UTF-8"?>
<BMECAT version="2005" xmlns="http://www.bmecat.org/bmecat/2005/bmecat_2005">
```

2.3.3 BMEcat 2005.1

It is expected that the following start tag / namespace is used for BMEcat 2005.1:

```
<?xml version="1.0" encoding="UTF-8"?>
<BMECAT version="2005.1" xmlns="http://www.bmecat.org/bmecat/2005.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

2.3.4 BMEcat 2005.2

It is expected that the following start tag / namespace is used for BMEcat 2005.2:

```
<?xml version="1.0" encoding="UTF-8"?>
<BMECAT version="2005.2" xmlns="http://www.bmecat.org/bmecat/2005.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

2.4 HEADER Information

Please see BMEcat standard documentation for this XML element and its nested components. The documentations are linked in chapter 2.1.

2.5 Multiple Classification Systems

In a T_NEW_CATALOG a PRODUCT can have many PRODUCT_FEATURES. Each of these PRODUCT_FEATURES elements can be used to contain a reference to a different Classification System. However, it is not recommended to use several Classification Systems in one PRODUCT definition and one catalog.

Note: For BMEcat 1.2 the necessary XML elements are ARTICLE and ARTICLE_FEATURES (see BMEcat changelog).

Note: Due to legacy reasons ARTICLE and ARTICLE_FEATURES are also available in 2005 and up. However, it is recommended to use PRODUCT and PRODUCT_FEATURES.

2.6 Nationalization Conventions for FVALUE

In the BMEcat specification 2005.1 and 2005 the decimal separator character for Values in (as XPath) /BMECAT/T_NEW_CATALOG/PRODUCT/PRODUCT_FEATURES/FEATURE/FVALUE is included.

However, it is not clear from BMEcat 1.2 specification what decimal separator character should be used for numbers (e.g. valuation of real Properties). Therefore, the following number notation according to the BMEcat specification 2005.1 is defined for completeness:

Always use dot notation (e.g. 108.0 and not 108,0).

Example: Number Value

```
<FVALUE>9.5</FVALUE>
```

Note: Furthermore, according to the BMEcat documentation no separator is allowed to delimit 1000's digits.

2.7 Translatable strings

Since BMEcat 2005 <FVALUE> is a multilingual string. That means, that for the datatype STRING_TRANSLATABLE several language dependent Values can be transferred via BMEcat.

To transport multilanguage content in one file, all languages have to be named in the BMEcat header (as XPath) /BMECAT/HEADER/CATALOG/LANGUAGE.

The default language can be set for one language. The concept of the default language is used in the following context:

The default language is used for missing translations. The perspective is the content-generating system. This system has to use the default language for items that are not translated.

Example: Language definition in BMEcat header

```
<BMECAT …
   <HEADER>
      <!--…-->
      <CATALOG>
         <LANGUAGE default="true">eng</LANGUAGE>
         <LANGUAGE>deu</LANGUAGE>
         <LANGUAGE>fra</LANGUAGE>
         <LANGUAGE>kor</LANGUAGE>
      <!--…-->
      </CATALOG>
<!--…-->
```

An example of multilanguage readable free text Values can be found in 3.2.1.3.

Note: The transfer of multilanguage content is from the view of ECLASS only relevant for free-text Values. IRDI-based Values are transferred by VALUE_IDREF (see 3.2.3.1) and these items are not relevant in context of multiple languages.

Note: BMEcat 1.2 does not support the transport of multilanguage content in one file. In this case it is necessary to create a separate BMEcat file for each language.

2.8 IRDI vs. Item-Code

Independent of the BMEcat Version, in ECLASS every element has an IRDI https://eclass.eu/support/technical-specification/structure-and-elements/irdi . This IRDI contains a six-digit code, the Item-Code. Especially in the past only the Item-Code was used, because it is listed individually in the ECLASS Basic CSV and is understood as leading ID in legacy systems.

However, only the IRDI as a whole is a unique identifier. Therefore, the recommendation is to always use the full IRDI.

2.9 Units

Basically, the Unit field in context of the PRODUCT_FEATURES in BMEcat is an optional element and freely assignable. So, it can be left empty or standard Units like UN/ECE can be used. BMEcat in general uses UN/ECE Units.

Concerning ECLASS it is recommended to always use one Unit, namely the IRDI of the Unit, even if the FEATURE is evaluated according to ECLASS Standard.

See also 3.2.6.

2.10 Use of XML Comments

The last specifications defined the use of XML comments to make the XML more friendly for human readers. Due to the fact, that ECLASS, BMEcat and XML are defined for machine readability these comments are technically not necessary.

However, such comments are a good tool for the development process and for validation.

In principle, it is recommended to dispense with these comments.

3 ECLASS to BMEcat Mapping

BMEcat defines the three transaction types T_NEW_CATALOG, T_UPDATE_PRODUCTS and T_UPDATE_PRICES. Focus of this document is only the first one. Therefore, in this chapter the concrete usage of ECLASS Structure Elements in BMEcat transaction type T_NEW_CATALOG is described. Even more detailed this chapter focuses on (as XPath) /BMECAT/T_NEW_CATALOG/PRODUCT/PRODUCT_FEATURES.

Note: All examples in this chapter will use ECLASS 11.0.

3.1 Classification Class and Application Class

In a T_NEW_CATALOG a PRODUCT can have many PRODUCT_FEATURES. Each of these elements can be used to contain a reference to a different classification system (s. 2.5). Therefore, in the first step to start with the description of a product in BMEcat you have to specify the classification system in PRODUCT_FEATURES. This puts the following FEATURES (cf. upcoming chapters) into context.

To define the classification system, the following three BMEcat elements have to be defined:

- REFERENCE_FEATURE_SYSTEM_NAME

The Value of this element is the used ECLASS Release. The BMEcat specification defines a rule for formatting the content of REFERENCE_ FEATURE_SYSTEM_NAME:

The name of the feature system (in capital letters) separated by a dash (minus sign) from the Major version and Minor version (separated by one dot) that is given. Moreover, the BMEcat specification has a list of known reference feature system names where the naming schema is explicitly mentioned. A list of all ECLASS Release Numbers can be found here: https://eclass.eu/support/content-creation/release-process/release-numbers-and-versioning

Example: “ECLASS 11.0” has to be written as “ECLASS-11.0”

- REFERENCE_FEATURE_GROUP_ID

The Value of this element indicates the ECLASS Classification. The Value is the ECLASS Coded Name of the classification as an eight-digit string without any separator. In combination with REFERENCE_FEATURE_SYSTEM_NAME it is a unique Value.

Example: “27141120”

- REFERENCE_FEATURE_GROUP_ID2

From BMEcat 2005.1 the value of this element should be the IRDI of the ECLASS Application Class to determine if the product description is according to ECLASS Advanced or ECLASS Basic.

Note 1: In BMEcat 1.2 REFERENCE_FEATURE_GROUP_ID2 is not available and not necessary.

Note 2: In BMEcat 2005 the REFERENCE_FEATURE_GROUP_ID2 is not necessary.

Example: A defined classification system

```
<PRODUCT_FEATURES>
   <REFERENCE_FEATURE_SYSTEM_NAME>ECLASS-11.0<REFERENCE_FEATURE_SYSTEM_NAME>
   <REFERENCE_FEATURE_GROUP_ID>27141151<REFERENCE_FEATURE_GROUP_ID>
   <REFERENCE_FEATURE_GROUP_ID2>0173-1---ADVANCED_1_1#01-
ADO312#009<REFERENCE_FEATURE_GROUP_ID2>
   <FEATURE>
   …
```

3.2 Properties and Values

Once the classification system has been defined, the product and its characteristics must be described. For this purpose, the FEATURE element is used in BMEcat. In general, there are two different ways to provide the data. First, describe a FEATURE without FTEMPLATE and secondly with FTEMPLATE. Both ways are described below and both variants can be used alternatively. However, it is recommended to use FTEMPLATE.

ECLASS defines a large set of Properties per Application Class. In some cases, it is not necessary or even not possible to define a Value for a FEATURE in BMEcat. In such cases it is generally proposed to skip the specific Property and not to create a FEATURE without any Value information.

Note: Until BMEcat 2005.1 including a Value must not be empty or null. That FVALUE in the XML Schema is a String with a minimum length of one character and since e.g., a 0 can represent a concrete Value or a space can be misunderstood by special parsers. However, if it is necessary to exchange a FEATURE without a Value or if it is technically mandatory to set a Value, the following rule applies: ECLASS defines a special string. ECLASS defines an Empty-Tag-String “ECLASS_NULL“. For BMEcat 2005.2 the value can be empty. Therefore, in BMEcat 2005.2 the Empty-Tag-String should not be used. See also 2.1.4 for improvements in 2005.2.

Note: In BMEcat 1.2 FTEMPLATE is not available. Therefore, only the first following variant is possible in this case.

- FEATURE without FTEMPLATE

For the transfer of product characteristics according an ECLASS Properties the FT_IDREF in FEATURE is used. The Value of FT_IDREF is the IRDI of the ECLASS Property.

Note: In BMEcat 1.2 FT_IDREF is not available. Therefore, FNAME has to be used.

Optionally the element FDESCR can be used to transfer the Preferred Name of the Property.

Example: FEATURE without FTEMPLATE

```
<FEATURE>
   <FT_IDREF>0173-1#02-BAA018#007</FT_IDREF>
   <FDESCR>length</FDESCR>
   <FVALUE>55.1</FVALUE>
   <FUNIT>0173-1#05-AAA480#003</FUNIT>
</FEATURE>
```

- FEATURE with FTEMPLATE

As alternative of using FT_IDREF from above BMEcat allows the usage of FTEMPLATE. In FTEMPLATE the FT_ID and the FT_NAME are encapsulated. FT_ID is evaluated with the IRDI again and FT_NAME carries the preferred name. FT_NAME instead of FT_DESCR is used because FT_NAME is a mandatory element.

Example: FEATURE with FTEMPLATE

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAA018#007</FT_ID>
      <FT_NAME>length</FT_NAME>
   </FTEMPLATE>
   <FVALUE>55.0</FVALUE>
   <FUNIT>0173-1#05-AAA480#003</FUNIT>
</FEATURE>
```

Note: In BMEcat 1.2 FTEMPLATE is not available.

3.2.1 Properties with Free Valuation

The next step after creating a semantic unique FEATURE is the definition of its Value. With “Free Valuation” this document means readable free text Values for which the range of Values is not restricted by ECLASS Value Lists. In general, the element FVALUE has to be used for this purpose.

Furthermore, ECLASS supports different data types. The full list can be found on https://eclass.eu/support/technical-specification/structure-and-elements/property#c2041 . In the following chapters for each data type one example is given for the transfer of one free Value.

3.2.1.1 Data type Boolean

The data type Boolean is treated specially in ECLASS. For the Values true and false a unique ECLASS Value List is defined. Please see: 3.2.3.2 for more details.

Example: FEATURE with data type Boolean

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN493#004</FT_ID>
      <FT_NAME lang="deu">Befestigung auf einem anderen Bauteil möglich</FT_NAME>
      <FT_NAME lang="eng">fixing on another component possible</FT_NAME>
   </FTEMPLATE>
   <VALUE_IDREF>0173-1#07-CAA016#001</VALUE_IDREF>
</FEATURE>
```

Note: Regarding multilingualism see also 2.7 and 3.2.1.3.

3.2.1.2 Data type String

The data type String is used for not translatable texts. In contrast proprietary values also suggested texts from Value Lists (see 3.2.3) can be used.

Note: The lang-attribute in VALUE should not be used.

Example: FEATURE with data type String

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAO677#002</FT_ID>
      <FT_NAME>Manufacturer name</FT_NAME>
   </FTEMPLATE>
   <VALUE>Sample company</VALUE>
</FEATURE>
```

3.2.1.3 Data type String Translatable

A characteristic from type String translatable is a more complex type. See chapter 2.7 for more details. It is always a combination of the real value and a corresponding language.

Note: There can be two variants: Univalent or multivalent. In the case of a univalent property definition, only one value can be set per language. In the case of a multivalent definition, multiple values can be specified for each language.

Example: FEATURE with data type String translatable & univalent

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAU734#001</FT_ID>
      <FT_NAME>Manufacturer product description</FT_NAME>
   </FTEMPLATE>
   <FVALUE lang="eng">Zack marker strip</FVALUE>
   <FVALUE lang="ces">Označovací štítek</FVALUE>
   <FVALUE lang="deu">Zackband</FVALUE>
   <FVALUE lang="dan">Mærkestrimler</FVALUE>
   <FVALUE lang="spa">Tira Zack</FVALUE>
   <FVALUE lang="fra">Repérage ZB</FVALUE>
   <FVALUE lang="hun">Jelölőcsík</FVALUE>
   <FVALUE lang="ita">Nastro Zack</FVALUE>
   <FVALUE lang="jpn">クイックマーカー</FVALUE>
   <FVALUE lang="kor">Zack 마커 스트립</FVALUE>
   <FVALUE lang="nld">Zackband</FVALUE>
   <FVALUE lang="nor">Merkeremse</FVALUE>
   <FVALUE lang="pol">Taśma oznaczników ponacinana</FVALUE>
   <FVALUE lang="por">Tira Zack</FVALUE>
   <FVALUE lang="rus">Планка Zack</FVALUE>
   <FVALUE lang="swe">Märkband</FVALUE>
   <FVALUE lang="tur">Etiket şeridi</FVALUE>
   <FVALUE lang="zho">快速标记条</FVALUE>
</FEATURE>
```

Example: FEATURE with data type String translatable & multivalent

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAF658#002</FT_ID>
      <FT_NAME>Material of chain</FT_NAME>
   </FTEMPLATE>
   <FVALUE lang="eng">steel</FVALUE>
   <FVALUE lang="deu">Stahl</FVALUE>
   <FVALUE lang="eng">plastic</FVALUE>
   <FVALUE lang="deu">Kunststoff</FVALUE>
   <FVALUE_TYPE>set</FVALUE_TYPE>
</FEATURE>
```

Note: See 3.2.2 for multivalent incl. FVALUE_TYPE in this example.

3.2.1.4 Data type Integer (count)

Example: FEATURE with data type Integer (count)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAP403#001</FT_ID>
      <FT_NAME>Number of contact points per level</FT_NAME>
   </FTEMPLATE>
   <FVALUE>2</FVALUE>
</FEATURE>
```

3.2.1.5 Data type Integer (measure)

The data type Integer (measure) also requires a Unit. For more information see 3.2.6.

Example: FEATURE with data type Integer (measure)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAA452#006</FT_ID>
      <FT_NAME>Operating pressure</FT_NAME>
   </FTEMPLATE>
   <FVALUE>20</FVALUE>
   <FUNIT>0173-1#05-AAA015#003</FUNIT>
</FEATURE>
```

3.2.1.6 Data type Integer (currency)

The data type Integer (currency) also requires a currency. For more information see 3.2.7.

Since there is no Property with this data type Integer (currency) in ECLASS, no example can be given.

3.2.1.7 Data type Real (count)

Example: FEATURE with data type Real (count)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN420#001</FT_ID>
      <FT_NAME>Power factor (cos phi)</FT_NAME>
   </FTEMPLATE>
   <FVALUE>0.98</FVALUE>
</FEATURE>
```

3.2.1.8 Data type Real (measure)

The data type Real (measure) also requires a Unit. For more information see 3.2.6.

Example: FEATURE with data type Real (measure)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAH005#005</FT_ID>
      <FT_NAME>rated voltage</FT_NAME>
   </FTEMPLATE>
   <FVALUE>400.0</FVALUE>
   <FUNIT>0173-1#05-AAA153#003</FUNIT>
</FEATURE>
```

3.2.1.9 Data type Real (currency)

The data type Real (currency) also requires a currency. For more information see 3.2.7.

Example: FEATURE with data type Real (currency)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAV928#001</FT_ID>
      <FT_NAME>cost of failure</FT_NAME>
   </FTEMPLATE>
   <FVALUE>99.75</FVALUE>
   <FUNIT>0173-1#08-AAA000#001</FUNIT>
</FEATURE>
```

3.2.1.10 Data type Rational (count)

The data type Rational is used to represent rational numbers (e.g. 1/3 or -11/17) without rounding as a fraction and consists of three Values. See also 3.2.2.

The first Value is a full integer, the second Value is the numerator and the third Value is the denominator.

Note: Three values must always be specified. If the full integer is not given, e.g. for 1/3, the first value of the triple is 0.

Example: FEATURE with data type Rational (count)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAZ199#001</FT_ID>
      <FT_NAME>factor</FT_NAME>
   </FTEMPLATE>
   <FVALUE>1</FVALUE>
   <FVALUE>1</FVALUE>
   <FVALUE>3</FVALUE>
</FEATURE>
```

3.2.1.11 Data type Rational (measure)

For an explanation see 3.2.1.10.

The data type Rational (measure) also requires a Unit. For more information see 3.2.6.

Example: FEATURE with data type Rational (measure)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAV059#002</FT_ID>
      <FT_NAME>pulse length threshold for low alarm (selected)</FT_NAME>
   </FTEMPLATE>
   <FVALUE>5</FVALUE>
   <FVALUE>6</FVALUE>
   <FVALUE>8</FVALUE>
   <FUNIT>0173-1#05-AAA114#003</FUNIT>
</FEATURE>
```

3.2.1.12 Data type Time

Since there is no Property with this data type Time in ECLASS, no example can be given. However, the Value should have the format according to ISO 8601:2004.

3.2.1.13 Data type Timestamp

The Value should have the format according to ISO 8601:2004.

Example: FEATURE with data type Timestamp (Zulu time; GMT/UTC)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-ABC500#001</FT_ID>
      <FT_NAME>time of measurement (inlet pressure)</FT_NAME>
   </FTEMPLATE>
   <FVALUE>1979-01-15T12:45:00Z</FVALUE>
</FEATURE>
```

Example: FEATURE with data type Timestamp (incl. timezone UTC+1)

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-ABC500#001</FT_ID>
      <FT_NAME>time of measurement (inlet pressure)</FT_NAME>
   </FTEMPLATE>
   <FVALUE>1979-01-15T12:45:00+01:00</FVALUE>
</FEATURE>
```

Note: These are possible values. All possibilities according to ISO 8601:2004 are given.

3.2.1.14 Data type Date

The Value should have the format according to ISO 8601:2004.

Example: FEATURE with data type Date

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAR972#002</FT_ID>
      <FT_NAME>Date of manufacture</FT_NAME>
   </FTEMPLATE>
   <FVALUE>2022-03-14</FVALUE>
</FEATURE>
```

Note: This is a possible value. All possibilities according to ISO 8601:2004 are given.

3.2.1.15 Data type URL

Since there is no Property with this data type URL in ECLASS 11.0, an example with an IRDI of ECLASS 12.0 is given.

Example: FEATURE with data type URL

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-ABA669#001</FT_ID>
      <FT_NAME>URI of manufacturer</FT_NAME>
   </FTEMPLATE>
   <FVALUE> https://www.phoenixcontact.com/</FVALUE>
</FEATURE>
```

3.2.2 Multivalent Properties

In general, all ECLASS Properties, except Boolean and Polymorphic Properties (see 3.3.4), are multivalent from scratch. This means, that product descriptions can have more than one Value for these Properties. To evaluate a BMEcat FEATURE with more than one Value the FVALUE element can be used several times. In this case the BMEcat element FVALUE_TYPE has to be used with “set”. However, FVALUE_TYPE “set” can only be used with more than one value.

Example: Transferring more than one free text Value

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAC289#004</FT_ID>
      <FT_NAME>Color</FT_NAME>
   </FTEMPLATE>
   <FVALUE>green</FVALUE>
   <FVALUE>white</FVALUE>
   <FVALUE_TYPE>set</FVALUE_TYPE>
</FEATURE>
```

Note: This concept is independent from the data type.

Furthermore, a combination of multilanguage and multivalent is possible. The following example shows such a data set.

Example: Multivalent and multilanguage readable free text Values

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAC289#004</FT_ID>
      <FT_NAME>Color</FT_NAME>
   </FTEMPLATE>
   <FVALUE lang="eng">green</FVALUE>
   <FVALUE lang="deu">grün</FVALUE>
   <FVALUE lang="fra">verte</FVALUE>
   <FVALUE lang="eng">white</FVALUE>
   <FVALUE lang="deu">weiß</FVALUE>
   <FVALUE lang="fra">blanch</FVALUE>
   <FVALUE_TYPE>set</FVALUE_TYPE>
</FEATURE>
```

Note: For the case multivalent (x Values) and multilanguage (n languages per Value) the sets of n FVALUES for each language have to be repeated x times for each Value. If a Value in a specific language is not available, the Empty-Tag-String (see. 3.2) has to be transferred.

Note: The BMEcat XML-Import- and -Export-Tools have to ensure that the serialized and interpreted order is correct.

3.2.3 Values from Value Lists

In addition to the free Value assignment described above, ECLASS contains specific Value Lists. ECLASS distinguishes explicit (3.2.3.1) and implicit (3.2.3.2) Value Lists. Furthermore, in most cases these Value Lists are a suggestion for possible Values (3.2.3.3). That means, that the user can choose to use one of the suggestions or use a free Value.

3.2.3.1 Explicit Values

Explicit Value Lists contain Values that represent commonly used concepts, like numbers (e.g. 10,25) which do not need definitions in dictionaries. Such explicit Values are expressed directly as FVALUE.

Note: FVALUE is without lang-attribute.

Example: Transfer one explicit Value

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN486#003</FT_ID>
      <FT_NAME>signal level</FT_NAME>
   </FTEMPLATE>
   <FVALUE>0 ... 10 V</FVALUE>
</FEATURE>
```

Example: Transfer more than one explicit Value

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN486#003</FT_ID>
      <FT_NAME>signal level</FT_NAME>
   </FTEMPLATE>
   <FVALUE>0 ... 10 V</FVALUE>
   <FVALUE>0 ... 5 V</FVALUE>
   <FVALUE>-10 ... +10 V</FVALUE>
   <FVALUE>-5 ... +5 V</FVALUE>
   <FVALUE_TYPE>set</FVALUE_TYPE>
</FEATURE>
```

Note: Concerning the Property definition, for explicit Value it is also possible to extend own Values. In such cases FVALUE contains further Values, which are not in ECLASS Dictionary.

Note: If the data type of the Property is String translatable then the lang attribute is necessary. If the language of FVALUE is the default language of the BMEcat document, then the lang-attribute is not necessary. See also 3.2.1.3.

Note: See 3.2.2 for FVALUE_TYPE in this example.

3.2.3.2 Implicit Values

Implicit Value Lists contain permissible Value(s) with a specific meaning. These meanings are encapsulated in separate ECLASS Values with a unique IRDI. Therefore, these Values are also called coded Values or ID Values. Such ECLASS Values will be transferred and expressed in BMEcat by using their IRDI in VALUE_IDREF.

Note: BMEcat 1.2 does support VALUE_IDREF for ID-based Values, so FVALUE has to be used in this case.

FVALUE_DETAILS can be used in the standard way to describe the Value more deeply. Basically, following the BMEcat specification, this element should only be filled if the Value differs from the ECLASS Dictionary. It is optional, but if an application wants to exchange the field, the recommendation is to transport the Preferred Name of the Value. However, this is only recommended for one Value. Or the same number of languages must then be exchanged in defined order for x Values. Here also the ECLASS-Empty-Tag has to be used.

Note: In BMEcat 1.2 FVALUE_DETAILS can be used only once in context of a feature, so it is a recommendation to use it only with single Value features and according to the BMEcat specification.

Example: Transfer one implicit Value

```
<FEATURE>
  <FTEMPLATE>
    <FT_ID>0173-1#02-BAA351#014</FT_ID>
    <FT_NAME>Farbe</FT_NAME>
  </FTEMPLATE>
  <VALUE_IDREF>0173-1#07-AAA875#004</VALUE_IDREF>
  <FVALUE_DETAILS>Grey</FVALUE_DETAILS>
</FEATURE>
```

Example: Transfer more than one implicit Value

```
<FEATURE>
  <FTEMPLATE>
    <FT_ID>0173-1#02-BAG640#008</FT_ID>
    <FT_NAME>Assembly type</FT_NAME>
  </FTEMPLATE>
  <VALUE_IDREF>0173-1#07-BAA576#004</VALUE_IDREF>
  <VALUE_IDREF>0173-1#07-AAM168#005</VALUE_IDREF>
  <FVALUE_TYPE>set</FVALUE_TYPE>
</FEATURE>
```

Note: According to 2.8 the IRDI of an implicit Value should always be used instead of any other code.

Note: See 3.2.2 for FVALUE_TYPE in this example.

3.2.3.3 Combination of Implicit and Free Text Values

Due to the fact, that in general a ECLASS Value List is a suggestion, also further Values can be used. Thus, a combination of Values from Value Lists and free Values is necessary.

In that case as many as needed implicit Values from the Value List are integrated via Value IRDI in VALUE_IDREF and additional Values are added via FVALUE.

Example: Transferring multiple implicit and free text Values

```
<FEATURE>
  <FTEMPLATE>
    <FT_ID>0173-1#02-BAB392#014</FT_ID>
    <FT_NAME>certificate/approval</FT_NAME>
  </FTEMPLATE>
  <FVALUE lang="eng">IECEE CB Scheme</FVALUE>
  <FVALUE lang="eng">UL Listed</FVALUE>
  <FVALUE lang="eng">cUL Listed</FVALUE>
  <FVALUE lang="eng">UL Listed</FVALUE>
  <FVALUE lang="eng">IECEE CB Scheme</FVALUE>
  <FVALUE lang="eng">cUL Listed</FVALUE>
  <VALUE_IDREF>0173-1#07-ABC243#001</VALUE_IDREF>
  <FVALUE lang="eng">KC</FVALUE>
  <FVALUE_TYPE>set</FVALUE_TYPE>
</FEATURE>
```

Note: Not possible for 2005.1 and older.

Note: The recommendation for 2005.1 and older is to use the preferred name via FVALUE for existing values instead of the IRDI to be schema compliant.

Note: See 3.2.2 for FVALUE_TYPE in this example.

3.2.4 Level Type

Note: This is only relevant for ECLASS advanced and BMEcat 2005.1 and newer

The Level Type specifies the usual qualities of the characteristic. It can be a minimal, a maximal, a typical or a nominal Value (min, max, typ, nom). Starting from ECLASS 7.0 Level Types are supported in ECLASS Advanced. For detailed information see https://eclass.eu/support/technical-specification/structure-and-elements/level-type.

In BMEcat always a set of four Values or a multiple of four Values has to be transported. The order of the Values is fix. ECLASS defines the Level Type order as follows:

1. MIN
2. MAX
3. NOM
4. TYP

The necessary Values depend on the ECLASS Property. If a Value is not necessary or not available, the ECLASS-Empty-Tag (see 3.2) has to be used.

Example: Level Type with MIN and MAX

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN513#003</FT_ID>
      <FT_NAME>multi-wired conductor cross-section (mm²)</FT_NAME>
   </FTEMPLATE>
   <FVALUE>0.2</FVALUE>
   <FVALUE>1.5</FVALUE>
   <FVALUE></FVALUE>
   <FVALUE></FVALUE>
</FEATURE>
```

Note: In 2005.1 and older "ECLASS_NULL" has to be used for empty FVALUE according to 3.2.

Note: The BMEcat specification defines the element FVALUE_TYPE (see also 3.2.2). Here "range" could be used as well. However, ECLASS describes the Level Type concept in detail. Therefore, FVALUE_TYPE should not be used here.

3.2.5 Axis Type

Note: This is only relevant for ECLASS Advanced and BMEcat 2005.1 and newer.

The Axis Type describes a placement in three-dimensional space in terms of a point and an axis direction. There are three different Axis Types available. Axis Type 1D, Axis Type 2D and Axis Type 3D. At the moment ECLASS only uses Axis Type 1D. For detailed information see https://eclass.eu/support/technical-specification/structure-and-elements/axis-1d . In BMEcat the Axis Type 1D always has six values.

Note: For this data type the Unit is optional. In ECLASS the length Unit is always "mm" and the rotation is given in "°". If a Unit is set in BMEcat, only the length Unit and not the angle Unit is exported at the level of the Property itself. The angle Unit is expected to always be the default Unit.

Example: Axis Type 1D

```
<FEATURE>
  <FTEMPLATE>
    <FT_ID>0173-1#02-AAN501#002</FT_ID>
    <FT_NAME>Arrangement (in mm)</FT_NAME>
  </FTEMPLATE>
  <FVALUE>4.9</FVALUE>
  <FVALUE>37.811065</FVALUE>
  <FVALUE>6.924349</FVALUE>
  <FVALUE>-125</FVALUE>
  <FVALUE>0</FVALUE>
  <FVALUE>0</FVALUE>
</FEATURE>
```

Example: Axis Type 2D

Since there is no Property with this data type Axis Type 2D in ECLASS, no example can be given.

Example: Axis Type 3D

Since there is no Property with this data type Axis Type 3D in ECLASS, no example can be given.

Note: The BMEcat XML-Import- and -Export-Tools have to ensure that the serialized and interpreted order is correct.

3.2.6 Units of Measure

The ECLASS Unit and quantity system is harmonized with DIN and via DIN with UN/ECE. ECLASS delivers a Unit file including all the harmonized Units. These Units have to be used in context of ECLASS content. In these cases, the IRDI of the ECLASS Unit is used in FUNIT (s.a. 2.9).

For examples, please see the embedded examples in 3.2.1.

Furthermore, according to ECLASS it is also possible to transfer a different Unit out of the defined quantity. Then an IRDI of the selected Unit has to be used inside of FUNIT.

3.2.7 Currency

The data types INTEGER_CURRENCY and REAL_CURRENCY also need a currency to the Value. In ECLASS the Properties with these data types reference a currency from the ECLASS currencies (e.g. 0173-1#08-AAA000#001 for Euro). It is proposed to handle the IRDI of the currency from the ECLASS Dictionary like Units. If no currency is specified, the default currency defined in the BMEcat header applies.

3.3 Advanced Structures

Note: Not for BMEcat 2005 and 1.2

BMEcat 2005.1 and newer is designed for the transport of all modelling features in ECLASS Advanced. In context of the provided advanced constructs like Blocks, Aspects, Cardinality, Polymorphism, etc. it is necessary to define the complete structure of Attributes in the BMEcat.

3.3.1 Block

In ECLASS a Block is a reusable set of Properties that is linked to a Class like an Application Class, Aspect or is nested in another Block via a specific Reference Property. To valuate a Block the corresponding Reference Property has to be defined as FEATURE in BMEcat.

In addition to the IRDI in FT_ID a document internal unique ID has to be defined in FID for the FEATURE representing the Reference Property. All nested Properties represented as other FEATUREs inside this Block have to use this FID in their FPARENT_ID. The FID is a free Value. FPARENT_ID points to an ID in FID of a super element. If the Property is on root level FPARENT_ID is "-1".

To sum it up FID contains the ID of the element and FPARENT_ID contains the ID of the parent element. With these two elements it is possible to create a hierarchical structure of Blocks and Features.

Only the Reference Property and its IRDI in FTEMPLATE is used as FEATURE. Due to the rules defined in the BMEcat Schema the FVALUE or VALUE_IDREF must be set. Therefore, the VALUE_IDREF is used and has to carry the IRDI of the Block.

The following examples represent the following ECLASS Structure from Figure 1.

structure					**description**		
L1	L2	L3	L4	L5	element name	element type	IRDI
CC-4					35-06-01-01 Sheet metal (ferrous metal)	classification class	0173-1#01-AGV467#003
	AC-A				Sheet metal (ferrous metal)	application class (advanced)	0173-1---ADVANCED_1_1#01-AGO036#003
		PR-R			Semifinished products weight sheet metal	reference property (to block)	0173-1#02-AAZ558#001
			BL		Semifinished products weight sheet metal	block	0173-1#01-AGG393#001
				PR	Square meter weight	property	0173-1#02-AAZ532#001
				PR	Unit weight	property	0173-1#02-AAZ533#001

Figure 1: Nested Blocks

Example: Reference Property "Semifinished products weight sheet metal" incl. Block

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAZ558#001</FT_ID>
      <FT_NAME>Semifinished products weight sheet metal</FT_NAME>
   </FTEMPLATE>
   <VALUE_IDREF>0173-1#01-AGG393#001</VALUE_IDREF>
   <FID>4321</FID>
   <FPARENT_ID>-1</FPARENT_ID>
</FEATURE>
```

Example: Property in Block

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAZ532#001</FT_ID>
      <FT_NAME>Square meter weight</FT_NAME>
   </FTEMPLATE>
   <FVALUE>10.0</FVALUE>
   <FUNIT>0173-1#05-AAA423#003</FUNIT>
   <FPARENT_ID>4321</FPARENT_ID>
</FEATURE>
```

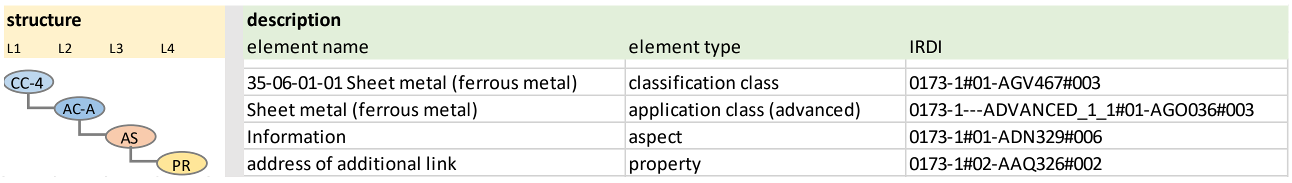
3.3.2 Aspect

Aspects contain structures describing a certain view of a product. Aspects are directly linked to an Application Class without a Reference Property. To bring an Aspect into application inside a BMEcat the new Element FEATURE_GROUP (see 2.1.3) was defined.

In general, the FEATURE list of a PRODUCT_FEATURES is extended by elements of FEATURE_GROUP where REFERENCE_FEATURE_GROUP_ID inside the FEATURE_GROUP carries the IRDI of the Aspect. The Aspect internal Properties are again represented as FEATURES following the same rules from above.

The Aspect as FEATURE_GROUP has no further hierarchy. Therefore, no FID or FPARENT_ID is defined.

Note: If a FEATURE inside the Aspect is on root level FPARENT_ID is "-1" again.



structure				description		
L1	L2	L3	L4	element name	element type	IRDI
CC-4				35-06-01-01 Sheet metal (ferrous metal)	classification class	0173-1#01-AGV467#003
	AC-A			Sheet metal (ferrous metal)	application class (advanced)	0173-1---ADVANCED_1_1#01-AGO036#003
		AS		Information	aspect	0173-1#01-ADN329#006
			PR	address of additional link	property	0173-1#02-AAQ326#002

Figure 2: Example for an Aspect in ECLASS

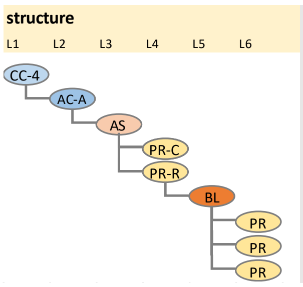
Example: ECLASS Aspect in BMEcat including one FEATURE

```
<FEATURE_GROUP>
   <REFERENCE_FEATURE_GROUP_ID>0173-1#01-ADN329#006</REFERENCE_FEATURE_GROUP_ID>
   <FEATURE>
      <FTEMPLATE>
         <FT_ID>0173-1#02-AAQ326#002</FT_ID>
         <FT_NAME>address of additional link</FT_NAME>
      </FTEMPLATE>
      <FVALUE>http://eclass.eu/</FVALUE>
   </FEATURE>
</FEATURE_GROUP>
```

3.3.3 Cardinality

One of the main elements of ECLASS Advanced is the usage of Cardinality of Blocks. The modelling of a Cardinality in ECLASS consists of a counter Property and a Reference Property. In application (here in BMEcat) the Reference Property in general is handled like any other Block (s.a. 3.3.1) and the counter Property is handled like any other Property as FEATURE. However, the FEATURE representing the counter Property defines how many times the Block is "instantiated" in BMEcat.

<u>Note:</u> If it is not necessary to transfer a cardinal Block, then the counter ("number of …") Property can be excluded from the FEATURE list. Furthermore, the Value "0" can be delivered. See general definition for FEATURES without Values in 3.2.



description		
element name	element type	IRDI
35-06-01-01 Sheet metal (ferrous metal)	classification class	0173-1#01-AGV467#003
Sheet metal (ferrous metal)	application class (advanced)	0173-1---ADVANCED_1_1#01-AGO036#003
Add on Documentation	aspect	0173-1#01-ADN464#009
number of documents	cardinality property	0173-1#02-AAN469#003
Additional information	reference property (to block)	0173-1#02-AAQ680#009
Additional information	block	0173-1#01-ADN356#009
version	property	0173-1#02-AAP003#002
type of document	property	0173-1#02-AAM660#005
description	property	0173-1#02-AAN466#002

Figure 3: Example for a Cardinality in ECLASS

<u>Example:</u> Counter Property

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN469#003</FT_ID>
      <FT_NAME>number of documents</FT_NAME>
   </FTEMPLATE>
   <FVALUE>20</FVALUE>
   <FPARENT_ID>-1</FPARENT_ID>
</FEATURE>
```

For FVALUE = 1 the dependent Reference Property for the Block has to be represented in BMEcat like defined in 3.3.1. For FVALUE > 1 multiple Reference Properties as FEATURES for the Cardinal Blocks with unique FIDs are needed. The Cardinal Blocks are then embedded like in 3.3.1.

One challenge in the BMEcat representation of the cardinal product characteristics is the order of these characteristics. The order of the cardinal characteristics can have a specific meaning inside of the data according to ECLASS. For example, in the context of CAx product descriptions it is important to know which connection is described, the first or last or even one between.

Furthermore, it is important to ensure that several data representations and their interpretation of one product return the same value for a specific characteristic. The context is important. Examples for a unique data access are the "ECLASS Technical Specification 15 - URI Path" (see https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-Technical-Specification-15-URI-Path-V1.0.pdf) or "ECLASS Technical Specification 48 - Item Data Retrieval by RESTful Web API" (see https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS_Specification_for_ItemDataRetrieval_by_RESTful_Web_API_-_V1.0.0_2023-03-03.pdf)

Therefore, the BMEcat element FORDER has to be used for the cardinal FEATURE representation of the Reference Property to semantically identify the context in a unique manner via a counting integer.

Example: Cardinal Reference Property incl. Block

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAQ680#009</FT_ID>
      <FT_NAME>Additional information</FT_NAME>
   </FTEMPLATE>
   <VALUE_IDREF>0173-1#01-ADN356#009</VALUE_IDREF>
   <FORDER>1</FORDER>
   <FID>1234</FID>
   <FPARENT_ID>-1</FPARENT_ID>
</FEATURE>
```

Note: The creator and interpreter of the BMEcat has to ensure that the order of the cardinal Blocks in XML is according to FORDER.

Note: FORDER is only used in the case of cardinality.

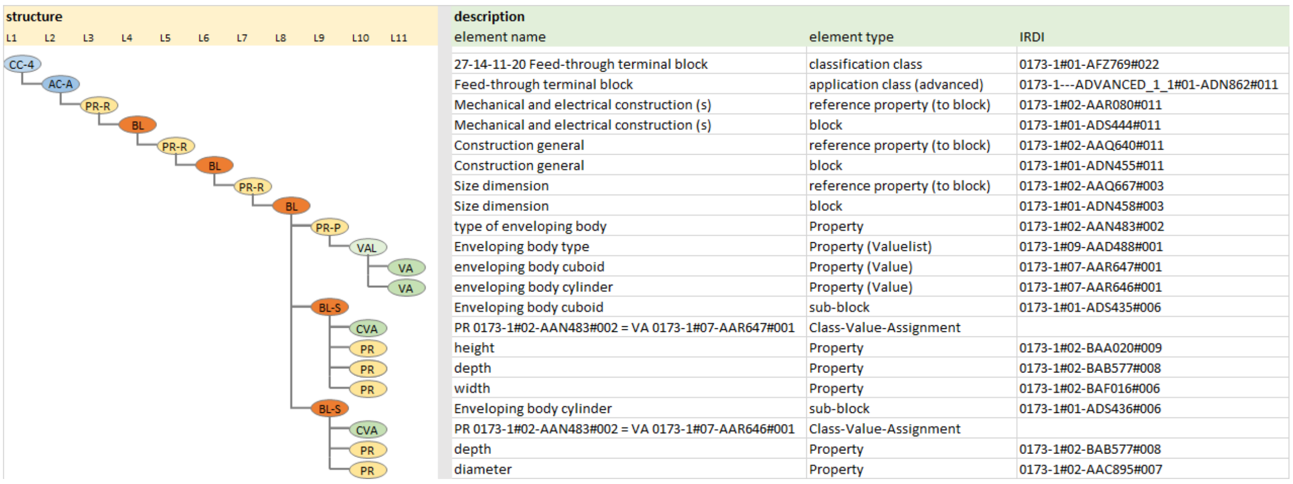
Example: Property in a Cardinal Block

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAP003#002</FT_ID>
      <FT_NAME>version</FT_NAME>
   </FTEMPLATE>
   <FVALUE>1.0.3</FVALUE>
   <FPARENT_ID>1234</FPARENT_ID>
</FEATURE>
```

Note: In BMEcat there is no link between Reference Property of the Cardinal Block and Counter. This knowledge comes from the ECLASS Dictionary.

3.3.4 Polymorphism

Polymorphism is an ECLASS modelling feature where a specialization of a Block and the selection of necessary Properties is done on instance level via a Class Value Assignment. A Polymorphism consists of a Reference Property, a selective Type Property and an inherited Block Hierarchy. In dependence of a selected Value out of a Value List and valuation of the selective Type Property, a specific Subclass has to be used and Properties have to be instanced as FEATUREs. The following Figure 4 shows an example of a Polymorphism in ECLASS.



structure	description		
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11	element name	element type	IRDI
CC-4	27-14-11-20 Feed-through terminal block	classification class	0173-1#01-AFZ769#022
AC-A	Feed-through terminal block	application class (advanced)	0173-1---ADVANCED_1_1#01-ADN862#011
PR-R	Mechanical and electrical construction (s)	reference property (to block)	0173-1#02-AAR080#011
BL	Mechanical and electrical construction (s)	block	0173-1#01-ADS444#011
PR-R	Construction general	reference property (to block)	0173-1#02-AAQ640#011
BL	Construction general	block	0173-1#01-ADN455#011
PR-R	Size dimension	reference property (to block)	0173-1#02-AAQ667#003
BL	Size dimension	block	0173-1#01-ADN458#003
PR-P	type of enveloping body	Property	0173-1#02-AAN483#002
VAL	Enveloping body type	Property (Valuelist)	0173-1#09-AAD488#001
VA	enveloping body cuboid	Property (Value)	0173-1#07-AAR647#001
VA	enveloping body cylinder	Property (Value)	0173-1#07-AAR646#001
BL-S	Enveloping body cuboid	sub-block	0173-1#01-ADS435#006
CVA	PR 0173-1#02-AAN483#002 = VA 0173-1#07-AAR647#001	Class-Value-Assignment	
PR	height	Property	0173-1#02-BAA020#009
PR	depth	Property	0173-1#02-BAB577#008
PR	width	Property	0173-1#02-BAF016#006
BL-S	Enveloping body cylinder	sub-block	0173-1#01-ADS436#006
CVA	PR 0173-1#02-AAN483#002 = VA 0173-1#07-AAR646#001	Class-Value-Assignment	
PR	depth	Property	0173-1#02-BAB577#008
PR	diameter	Property	0173-1#02-AAC895#007

Figure 4: Example of a Polymorphism

First, the Reference Property (e.g. "Size dimension") has to be transformed like any other Block described in 3.3.1. The VALUE_IDREF carries also the IRDI of the main Block (e.g. "Size dimension").

Example: Reference Property pointing to a Polymorphic Block "Size dimension"

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAQ667#003</FT_ID>
      <FT_NAME>Size dimension</FT_NAME>
   </FTEMPLATE>
   <VALUE_IDREF>0173-1#01-ADS435#006</VALUE_IDREF>
   <FID>480599771</FID>
   <FPARENT_ID>480599770</FPARENT_ID>
</FEATURE>
```

Note: According to https://eclass.eu/fileadmin/Redaktion/pdf-Dateien/Wiki/ECLASS-Technical-Specification-15-URI-Path-V1.0.pdf in Chapter 5.2 VALUE_IDREF is always the IRDI of the specialized Block. It does change with the selective Property.

Second, the selective Type Property (e.g. "type of enveloping body") has to be embedded in BMEcat as any other Property inside of a Block and with an implicit Value List (see. 3.2.3.2).

Example: Selective Type Property defining the Polymorphic Block for “enveloping body cuboid”

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-AAN483#002</FT_ID>
      <FT_NAME>type of enveloping body</FT_NAME>
   </FTEMPLATE>
   <VALUE_IDREF>0173-1#07-AAR647#001</VALUE_IDREF>
   <FVALUE_DETAILS>enveloping body cuboid</FVALUE_DETAILS>
   <FID>515657271</FID>
   <FPARENT_ID>480599771</FPARENT_ID>
</FEATURE>
```

Third, the Properties within the selected Block have to be evaluated. The necessary knowledge regarding the Class (Block) Value Assignment comes from the ECLASS Dictionary.

Example: Specific Properties in Polymorphic Block

Note: The described concept for the representation of a Polymorphism also applies for the "Advanced Polymorphism” defined in https://eclass.eu/support/content-creation/content-development-platform/polymorphism-help-page .

3.4 Additional Files

In addition to the pure XML exchange of product data further files like images or drawings can be linked. Basically, the section MIME_INFO in PRODUCT is used. However, two variants can be distinguished. The linking via URL and the exchange as container file (e.g. ZIP). Furthermore, ECLASS has also specified structures to which further data can be linked.

Note: A valid MIME_TYPE is necessary. To support as many as possible the BMEcat schema has been extended. See 2.1.4.

3.4.1 URL

The first approach to add further files to a BMEcat is the usage of URLs.

Example: Link a file via URL incl. multilanguage support

```
<MIME_INFO>
   <MIME>
      <MIME_TYPE>url</MIME_TYPE>
      <MIME_SOURCE
lang="deu">http://www.phoenixcontact.com/de/produkte/2799694</MIME_SOURCE>
      <MIME_SOURCE
lang="eng">http://www.phoenixcontact.com/gb/products/2799694</MIME_SOURCE>
      <MIME_SOURCE
lang="zho">http://www.phoenixcontact.com/gb/products/2799694</MIME_SOURCE>
      <MIME_DESCR lang="deu">Deeplink</MIME_DESCR>
      <MIME_DESCR lang="eng">Deeplink</MIME_DESCR>
      <MIME_DESCR lang="zho">Deeplink</MIME_DESCR>
      <MIME_ALT lang="deu">Internetadresse</MIME_ALT>
      <MIME_ALT lang="eng">Internetadresse</MIME_ALT>
      <MIME_ALT lang="zho">Internetadresse</MIME_ALT>
      <MIME_PURPOSE>others</MIME_PURPOSE>
   </MIME>
</MIME_INFO>
```

3.4.2 Exchange as Container File

The second approach is to exchange the BMEcat file including the additional files in one container file. For this approach the element HEADER/CATALOG/MIME_ROOT has to be defined (e.g. "./"). This MIME_ROOT defines the directory or URI to which the relative paths in MIME_SOURCE refer. MIME_SOURCE then defines the relative path and the filename. The path is then a concatenation of both strings.

Compare BMEcat specification from 2.1.3 on page 27 and 71 ff.

Example: Link a file in a container file

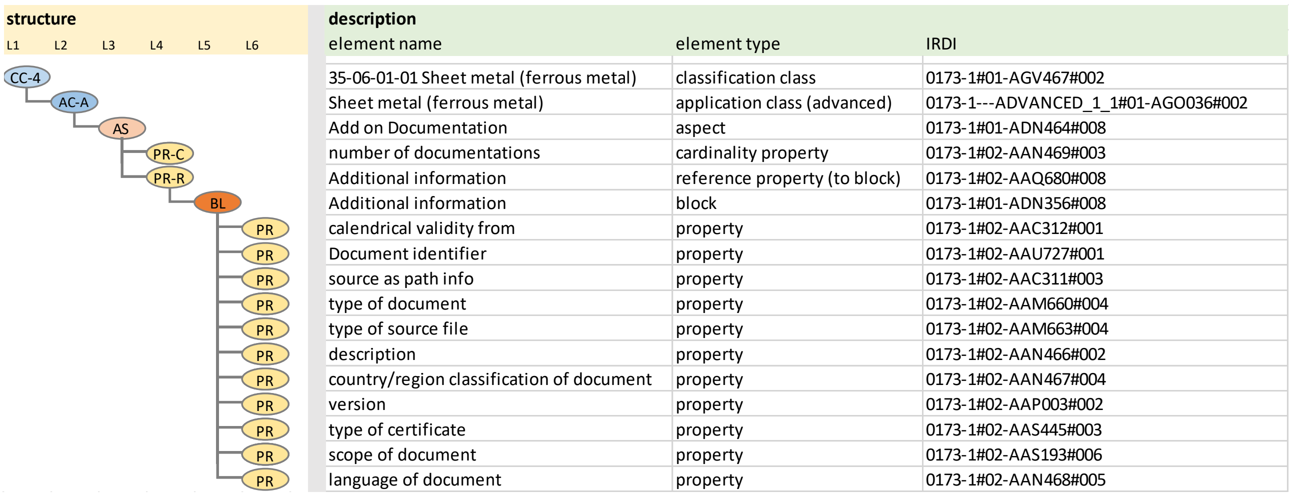
```
<!--…-->
<HEADER>
<!--…-->
   <CATALOG>
<!--…-->
      <MIME_ROOT>./</MIME_ROOT>
<!--…-->
<MIME_INFO>
   <MIME>
      <MIME_TYPE>image/jpeg</MIME_TYPE>
      <MIME_SOURCE>mimes/products/images/45700_3000_int_04.jpg</MIME_SOURCE>
      <MIME_ALT lang="deu">Artikeldarstellung</MIME_ALT>
      <MIME_ALT lang="eng">article view</MIME_ALT>
      <MIME_ALT lang="zho">article view</MIME_ALT>
      <MIME_PURPOSE>normal</MIME_PURPOSE>
   </MIME>
</MIME_INFO>
```

Table 3: Example of a ZIP file structure (with exemplary structure)

Level 1	Level 2	Level 3	Level 4	Level 5	Comment
catalog.zip					ZIP file with custom naming containing an XML file and files in an unrestricted folder structure
	BMEcat.xml				XML according to BMEcat schema
	./mimes				Custom folder
		/general			Custom folder
			/images		Custom folder
				/*name*.jpg	Custom file with individual name
		/products			Custom folder
			/cad		Custom folder
				/*drawing*.stp	Custom file with individual name
			/images		Custom folder
				/*name*.jpg	Custom file with individual name

Note: Such a ZIP-Container can have an individual number of levels. For instance, in this example the files could be on level 2 as well.

3.4.3 Aspect “Add on Documentation”



structure	description		
L1 L2 L3 L4 L5 L6	element name	element type	IRDI
CC-4	35-06-01-01 Sheet metal (ferrous metal)	classification class	0173-1#01-AGV467#002
AC-A	Sheet metal (ferrous metal)	application class (advanced)	0173-1---ADVANCED_1_1#01-AGO036#002
AS	Add on Documentation	aspect	0173-1#01-ADN464#008
PR-C	number of documentations	cardinality property	0173-1#02-AAN469#003
PR-R	Additional information	reference property (to block)	0173-1#02-AAQ680#008
BL	Additional information	block	0173-1#01-ADN356#008
PR	calendrical validity from	property	0173-1#02-AAC312#001
PR	Document identifier	property	0173-1#02-AAU727#001
PR	source as path info	property	0173-1#02-AAC311#003
PR	type of document	property	0173-1#02-AAM660#004
PR	type of source file	property	0173-1#02-AAM663#004
PR	description	property	0173-1#02-AAN466#002
PR	country/region classification of document	property	0173-1#02-AAN467#004
PR	version	property	0173-1#02-AAP003#002
PR	type of certificate	property	0173-1#02-AAS445#003
PR	scope of document	property	0173-1#02-AAS193#006
PR	language of document	property	0173-1#02-AAN468#005

Figure 5: Aspect “Add on Documentation”

A third way for additional files in BMEcat according to ECLASS is the usage of the Aspect “Add on Documentation”. In this case the Property “source as path info” is evaluated with the name of the file. This can be an URL or a relative path in the ZIP container again.

Example: Aspect “Add on Documentation”

```
<FEATURE_GROUP>
   <REFERENCE_FEATURE_GROUP_ID>0173-1#01-ADN464#008</REFERENCE_FEATURE_GROUP_ID>
<!--<FT_NAME>number of documentations</FT_NAME>-->
   <FEATURE>
      <FTEMPLATE>
         <FT_ID>0173-1#02-AAQ680#008</FT_ID>
         <FT_NAME>Additional Information</FT_NAME>
      </FTEMPLATE>
      <VALUE_IDREF>0173-1#01-ADN356#008</VALUE_IDREF>
      <FID>482014393</FID>
      <FPARENT_ID>-1</FPARENT_ID>
   </FEATURE>
   <FEATURE>
      <FTEMPLATE>
         <FT_ID>0173-1#02-AAC311#003</FT_ID>
         <FT_NAME>source as path info</FT_NAME>
      </FTEMPLATE>
      <FVALUE lang="eng">./images/58200_1000_int_04.jpg</FVALUE>
      <FVALUE_DETAILS>58200_1000_int_04.jpg</FVALUE_DETAILS>
      <FID>482014394</FID>
      <FPARENT_ID>482014393</FPARENT_ID>
   </FEATURE>
   <FEATURE>
      <FTEMPLATE>
         <FT_ID>0173-1#02-AAM663#004</FT_ID>
         <FT_NAME>type of source file</FT_NAME>
      </FTEMPLATE>
      <FVALUE lang="eng">image/jpg</FVALUE>
      <FVALUE_DETAILS>image/jpg</FVALUE_DETAILS>
      <FID>482014398</FID>
      <FPARENT_ID>482014393</FPARENT_ID>
   </FEATURE>
```