EtherCAT® Slave Information

Specification

Document: ETG.2000 S (R) V1.0.6

Nomenclature:	ETG-Number	ETG.2000
	Type	S (Standard)
	State	R (Release)
	Version	1.0.6

Created by:	ETG
Contact:	info@ethercat.org
Date:	2014-03-28

EtherCAT® Technology Group

LEGAL NOTICE

Trademarks and Patents

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics. In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Copyright

© EtherCAT Technology Group

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization is prohibited. Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

C	Conditional
CMD	Command
CoE	CAN Application Protocol over EtherCAT
DC	Distributed Clock
DPRAM	Dual-Ported RAM
ENI	EtherCAT Network Information (EtherCAT XML Master Configuration)
EoE	Ethernet over EtherCAT
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information (EtherCAT Devices Description)
ESM	EtherCAT State Machine
ETG	EtherCAT Technology Group
FMMU	Fieldbus Memory Management Unit
FoE	File Access over EtherCAT
FPRW	Configured Address Physical ReadWrite
I/O	Input/Output
IDN	Identification Number (Servo Profile Identifier)
IEC	International Electrotechnical Commission
LRD	Logical Read
LRW	Logical ReadWrite
LSB	Least Significant Bit
LWR	Logical Write
M	Mandatory
MAC	Media Access Controller
MI	(PHY) Management Interface
MII	Media Independent Interface
MSB	Most Significant Bit
ns	nanoseconds (10^{-9} seconds)
O	Optional
OD	Object Dictionary
PDO	Process Data Object
PreOp	Pre-Operational
SDO	Service Data Object
SII	Slave Information Interface
SM	SyncManager
SoE	Servo Drive Profile over EtherCAT
SOF	Start of Frame
SPI	Serial Peripheral Interface
WD	Watchdog
WKC	Working Counter
XML	eXtensible Markup Language

REFERENCES

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ETG Standards

[1] ETG.1000.2: Physical Layer service definition and protocol specification

[2] ETG.1000.3: Data Link Layer service definition

[3] ETG.1000.4: Data Link Layer protocol specification

[4] ETG.1000.5: Application Layer service definition

[5] ETG.1000.6: Application Layer protocol specification

[6] ETG.1004: EtherCAT Unit Specification

[7] ETG.1020: EtherCAT Guidelines and Protocol Enhancements

[8] ETG.1400: EtherCAT Technology Description

[9] ETG.5001: EtherCAT Modular Device Profile

Other References

--

1 Terms, Definitions and Word Usage

1.1 Terms and Definitions

The terms and definitions of ETG.1000 series shall be fully valid, unless otherwise stated.

1.2 Word usage: obsolete

Element/attributes which are described with “obsolete” shall not be used any more. A configuration tool shall use those elements as described behind the word “configtool”.

1.3 Word usage: for future use

Elements/attributes which are described with “For future use” shall not be used. A configuration tool shall skip those elements/attributes.

1.4 Word usage: shall, should, may, can

The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

2 Preface

For each EtherCAT Slave a device description, the so called EtherCAT Slave Information (ESI) has to be delivered. This is done in form of an XML file (eXtensible Markup Language). It describes EtherCAT specific as well as application specific features of the slave.

The ESI file is used by an EtherCAT configuration tool to generate the EtherCAT Network Information (ENI).

There is always one unique ESI file for a device. Revision changes on the device's hardware and/or software may have to be reflected in the ESI of this device (usually by the Revision Number).

This document specifies the structure and usage of the ESI file. The structure is represented by the schema files for generation and validation of the particular ESI file.

3 Document

3.1 Order of elements

The order of elements and attributes is given by the figures and tables whereas the topmost element within a figure or table shall be described first within the ESI file. It is important to use the elements in the right order. Top elements of a figure have to be used first, bottom elements last.

3.2 Explanation of Figures

In Figure 1 an example for `Choice` description is shown. Its meaning is that one (due to the "1" on the input of the `Choice` field) of the branches of the right side of the `Choice` field hast to be chosen, i.e. one of the following possibilities:

- One instance of SubElement_1
- Combination of one instance of SubElement_2a and one instance of SubElement2b
- One to four instances of SubElement_3 (which has "SampleType" as data type)

No other combinations are allowed in this case.

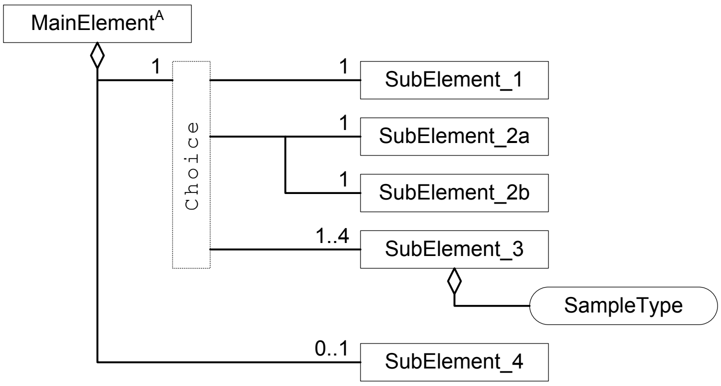

Figure 1: Sample of Choice description

Table 1 describes the terms of occurrence of the elements.

Table 1: Description of the terms of occurrence

Occurrence	Use	Description
0..1	O	Optional Minimum: zero Maximum: one
1	M	Mandatory exact one element allowed
0..∞	O	Optional Minimum: zero Maximum: infinite
1..∞	M	Mandatory Minimum: one Maximum: infinite
0..n	O	Optional Minimum: zero Maximum: n
1..n	M	Mandatory Minimum: one Maximum: n

3.3 Explanation of describing Tables

Table 2 describes the element description of the sample shown in Figure 1.

Table 2: Sample Description of Elements

Element	Data Type	Use	Description
MainElement^A	-	M	Main element Elements are shown as rectangles. The superscript A indicates that the element has attributes. When attributes are possible a table with further information follows.
MainElement:SubElement_1	xs:int	M^D	Sub elements are displayed with a colon. D: either SubElement_1, SubElement_2a and 2b, or SubElement_3 A superscript D indicates additional information in the Description field.
MainElement:SubElement_2a	xs:boolean	M^D	D: either SubElement_1, SubElement_2a and 2b, or SubElement_3
MainElement:SubElement_2b	xs:string	M^D	D: either SubElement_1, SubElement_2a and 2b, or SubElement_3
MainElement:SubElement_3	SampleType^T	M^D	D: either SubElement_1, SubElement_2a and 2b, or SubElement_3 Types are shown as rounded rectangles.All used types are described in clause 13.
MainElement:SubElement_4	xs:string	O^D	D: mandatory for devices supporting mailbox services. Optional elements may be omitted.

^{D} Additional information can be found in the “Description” field.
^{A} This element has an attribute described below.
^{T} See corresponding type descriptions in clause 13.

“Use” Column
The *Use* column reflects the property given by the schema but does not necessarily describe the usage of the element/attribute as it shall be used according to this specification. If the specification requires different usage than the schema does, this is indicated by a superscripted “D”. In this case the definition is found in the “Description” column.

Abbreviations:
M = Mandatory: element/attribute shall be present
O = Optional: element/attribute may be present if value differs from default value

Attributes
Attributes are used for additional information of the corresponding element. Element with attributes are marked with a superscripted “A” in tables as well as in the figures.

Default values
Optional elements and/or attributes may be omitted. In this case a default value is used. The default value is always “0”/“false”. In case of any difference the defalut value is specified in the “Description” column.

3.4 Referencing of elements and attributes

Elements and SubElements are represented by a separating colon, e.g. *Element:SubElement*.

Elements and their Attributes are represented by a separating ‘@’, e.g. *Element@Attribtute*.

Reference elements and attributes are written in *italic* letters.

4 General

4.1 ESI File

Every EtherCAT Slave shall be delivered with an EtherCAT Slave Information (ESI) file.

4.2 Crc32 Calculation

Cyclic Redundancy Check (CRC) of the binary content of the Tag definition. The data type of the CRC is UINT32. The FCS is calculated for the content between the start tag and the end tag (excluding the start tag and CRC32 attribute, including the end tag, e.g. </Module>):

```
<Device Crc32="12334567">
...
</Device>
```

or

```
<Module Crc32="12334567">
...
</Module>
```

CRC Polynomial $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$
$= 104C11DB7h$

NOTE: The polynomial is described in ISO 3309.

4.3 Element VendorSpecific

Information provided in a *VendorSpecific* element does not have to be supported, and usually is not, by the broad range of EtherCAT configuration tools. For this reason all relevant information for proper usage of the slave device shall be provided by the elements/attributes specified by this standard.

4.4 Hints

4.4.1 Path Description

Other files may be referenced within the ESI file. The path description to the reference file follows the rules described in Table 3.

No absolute file pathes shall be used.

Table 3: Reference to Files

Reference	File Path
Reference file within the same folder	ReferencedFile.xml
Reference file in folder above	..\ReferencedFile.xml
Reference file in extra folder	ExtraFolder\ReferencedFile.xml

No special characters shall be used for file names and file paths. This includes language specific letters or special characters which may not be properly shown by operating systems or programs with different language settings or properly interpreted by file systems including servers.

4.4.2 Not Character Symbols

“not a character” symbols (e.g. TAB, LineFeed) shall not be used. Exceptions are allowed for Comment and Descriptions elements.

5 Schemas

An ESI file is based on the following schemas:

- EtherCATInfo.xsd

  This schema describes the mandatory ESI file structure. The schema requires the EtherCATBase.xsd file in the same folder.

- EtherCATBase.xsd

  This schema describes complex EtherCAT specific data types. Those data types are described in an extra file so that several files can use them.

- EtherCATDiag.xsd

  This schema describes the diagnosis messages of one of more EtherCAT devices.

  The diagnosis messages can be handled in two ways:

  - It is integrated in the mandatory ESI file
  - It is describe in a separate file based on the EtherCATDiag.xsd schema. A reference in the ESI file will point to the diagnosis file where the messages are supposed to be.

- EtherCATDict.xsd

  This schema describes the dictionary structure of an EtherCAT device. It includes the description of data types and objects used within the object dictionary of the device.

  The Dictionary can be handled in two ways:

  - It is included in the mandatory ESI file
  - It is described in a separate file based on the EtherCATDict.xsd schema. A reference in the ESI file will point to the dictionary file where the Dictionary is supposed to be.

- EtherCATModule.xsd

  This schema describes the modules structure (physical ones and functional ones) of one device.

  The module description can be handled in two ways:

  - It is included in the mandatory ESI file
  - It is described in a separate file based on the EtherCATModule.xsd schema. A reference in the ESI file will point to the module description file where the module description is supposed to be.

6 XML Data Types

XML Schema Data Types shown in Table 4 and Table 5 are based on *XML Schema: Datatypes*, which is part 2 of the specification of the XML Schema language (http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/datatypes.xml).

Table 4: XML Schema - Simple Data Types

Data Type	Description
xs:string	ASCII string “not a character” symbols (e.g. TAB, LineFeed) shall not be used NOTE: They may be used in XML specific comments.
xs:boolean	Boolean value ‘0’: FALSE ‘1’: TRUE
xs:hexBinary	012345 → 0x01 LSB, 0x45 MSB

Table 5: XML Schema - Derivative Data Types

Data Type	Description
xs:NMTOKEN	An Nmtoken (name token) is any mixture of name characters. A name is an Nmtoken with a restricted set of initial characters. Disallowed initial characters for names include digits, diacritics, the full stop and the hyphen.
xs:int	String of bits (signed 32-bit integer), based on an integral number Range: -2147483648…0…+2147489647
xs:integer	String of bits (unbounded integer), based on an integral number Range: -∞…0…+∞

Table 6: EtherCATBase Type HexDecValue

Data Type	Description
HexDecValue	Represents a hexadecimal value either in hexadecimal or decimal format, e.g. 12345 → 12345 (dec) #x12345 → 0x12345 (hex), whereas 0x45 is LSB and 0x01 MSB Only positive values are allowed. Possible exceptions are mentioned in the corresponding element definition.

7 Base Data Types

NOTE: Data type definition will become part of ETG.1020 specification. This chapter is valid until ETG.1020 containing the chapter data types is released. Then refer to ETG.1020.

Table 7 describes the Base Data Types.

Table 7: Base Data Types

CoE data type					
Index	**Name**	**Base Data Type**	**Description**	**Bit Size**	**Range**
0x0001	BOOLEAN	BOOL BIT	‘0’: FALSE ‘1’: TRUE	1	
0x001E	BYTE	USINT	Unsigned Byte	8	
		Bit String			
0x0030	BIT1	BIT1		1	
0x0031	BIT2	BIT2		2	
0x0032	BIT3	BIT3		3	
0x0033	BIT4	BIT4		4	
0x0034	BIT5	BIT5		5	
0x0035	BIT6	BIT6		6	
0x0036	BIT7	BIT7		7	
0x0037	BIT8	BIT8		8	
0x002D	BITARR8	BITARR8	8 individual Bits	8	
0x002E	BITARR16	BITARR16	16 individual Bits	16	
0x002F	BITARR32	BITARR32	32 individual Bits	32	
		Signed Integer			
0x0002	INTEGER8	SINT	Short Integer	8	-128 to 127
0x0003	INTEGER16	INT	Integer	16	-32 768 to 32 767
0x0010	INTEGER24	INT24		24	
0x0004	INTEGER32	DINT	Double Integer	32	-2^{31} to 2^{31-1}
0x0012	INTEGER40	INT40		40	
0x0013	INTEGER48	INT48		48	
0x0014	INTEGER56	INT56		56	
0x0015	INTEGER64	LINT	Long Integer	64	-2^{63} to 2^{63-1}
		Unsigned Integer			
0x0005	UNSIGNED8	USINT	Unsigned Short Integer	8	0 to 255
0x0006	UNSIGNED16	UINT	Unsigned Integer / Word	16	0 to 65 535
0x0016	UNSIGNED24	UINT24		24	
0x0007	UNSIGNED32	UDINT	Unsigned Double Integer	32	0 to 2^{32-1}
0x0018	UNSIGNED40	UINT40		40	
0x0019	UNSIGNED48	UINT48		48	
0x001A	UNSIGNED56	UINT56		56	

CoE data type					
0x001B	UNSIGNED64	ULINT	Unsigned Long Integer	64	0 to 2^{64-1}
		Floating			
0x0008	REAL32	REAL	Floating point	32	
0x0011	REAL64	LREAL	Long float	64	

Table 8 describes the Base Data Types with variable length.

Table 8: Base Data Types with variable length

CoE data type					
Index	**Name**	**Base Data Type**	**Description**	**Bit Size**	**Range**
		Strings			
0x0009	VISIBLE_STRING	STRING(n)	VisibleString (1 octet per character)	8*n	
		Octet field			
0x000A	OCTET_STRING	ARRAY [0..n] OF BYTE	Sequence of octets (data type BYTE)	8*(n+1)	
0x000B	UNICODE_STRING	ARRAY [0..n] OF UINT	Sequence of UINT	16*(n+1)	

8 State Transitions

Table 9 describes all defined EtherCAT State Transitions which are selectable via the xs:NMTOKEN data type.

Table 9: Allowed State Transitions

Data Type	Description
xs:NMTOKEN	Allowed values in high commas: ‘IP’: Init → Pre-Operational ‘PS’: Pre-Operational → Safe-Operational ‘PI’: Pre-Operational → Init ‘SP’: Safe-Operational → Pre-Operational ‘SO’: Safe-Operational → Operational ‘SI’: Safe-Operational → Init ‘OS’: Operational → Safe-Operational ‘OP’: Operational → Pre-Operational ‘OI’: Operational → Init ‘IB’: Init → Bootstrap ‘BI’: Bootstrap → Init

9 EtherCATInfo

9.1 General

The EtherCATInfo element is the root element of the EtherCAT Slave Device description. Figure 2 shows the class structure of the EtherCAT slave information.

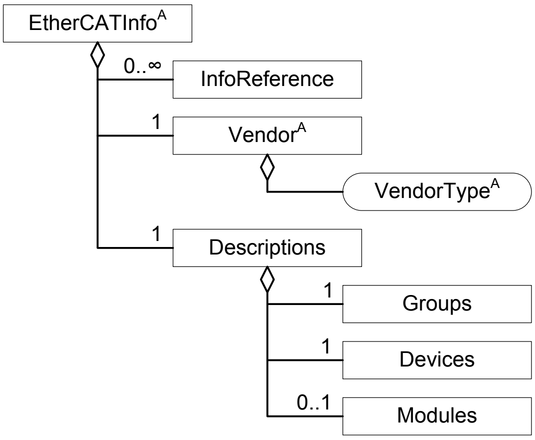

Figure 2: EtherCAT Slave Information class diagram

Table 10 describes the attributes of element EtherCATInfo.

Table 10: Attributes of EtherCATInfo

Element	Data Type	Use	Description
Version	xs:string	O	EtherCAT device description schema version used as schema for this device description file.

Table 11 describes the content of element EtherCATInfo.

Table 11: Content description of EtherCATInfo

Element	Data Type	Use	Description
InfoReference	xs:string	O	File name of an external file based on EtherCATModule.xsd which describes Modules
Vendor^A	VendorType^T	M	Describes the identity of the device vendor with its name and EtherCAT Vendor ID assigned by the EtherCAT Technology Group
Descriptions	--	M	Describes the EtherCAT device(s) using the elements Groups, Devices and Modules.

Table 12 describes the attributes of EtherCATInfo.

Table 12: Attributes of Vendor

Attribute	Data Type	Use	Description
FileVersion	xs:int	O	Version of the EtherCAT Slave Information (ESI) file. This version is vendor specific and is not evaluated by the configuration tool. NOTE: Do not confuse FileVersion with the schema version!

Table 13 describes the content of element Description.

Table 13: Content description of Descriptions

Element	Data Type	Use	Description
Groups	--	M	Similar devices can be assigned to one group. The structuring of devices to groups is used by a configuration tool. The element Groups may define one or several groups with e.g. name and a bitmap symbol. The assignment of a device to a group is made within the element *Device:Group.*
Devices	--	M	Element Devices may describe one or several devices with their EtherCAT features such as SyncManagers, FMMUs, and Dictionary.
Modules	--	O	Element Modules describes all possible modules that can be configured for a modular or complex device. This is typically, but not exclusively, used for devices supporting the Modular Device Profile (ETG.5001)

10 Groups

The structuring of devices in groups is used by the configuration tool to group devices together. There is no slave functionality connected to the Groups element.

Figure 3 shows how the Groups element is composed.

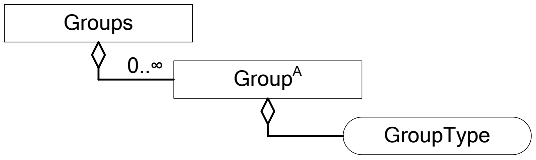

Figure 3: Content of Groups

Table 14 describes the content of element Groups.

Table 14: Content description of Groups

Element	Data Type	Use	Description
Group^A	GroupType^T	O	One Group groups similar devices with slightly different features.

Table 15 describes the attributes of element Group.

Table 15: Attributes of Group

Attribute	Data Type	Use	Description
SortOrder	xs:int	O	Helps to display multiple groups in the order intended by the vendor. Groups are sorted in ascending order of this value.
ParentGroup	--	O	For future use

11 Devices

Figure 4 shows how the Devices element is composed. It describes all settings and features necessary to run the device.

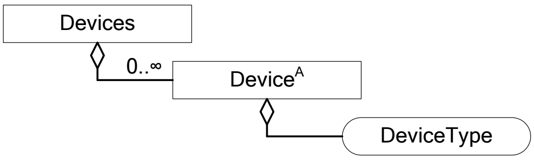

Figure 4: Content of Devices

Table 16 describes the content of element Devices.

Table 16: Content description of Devices

Element	Data Type	Use	Description
Device^A	DeviceType^T	O	Holds all information about the device like SyncManager and FMMU, object dictionary, data types and the PDO mapping and assign description.

Table 17 describes the attributes of element Devices.

Table 17: Attributes of Device

Attribute	Data Type	Use	Description
Invisible	xs:boolean	O^D	D: Mandatory for devices which have no EtherCAT functionality (no ESC), e.g. power supply device. Allowed values: '0': EtherCAT slave (has EtherCAT Slave Controller (ESC)) '1': No EtherCAT slave (i.e. not ESC). The device is shown by hardware configuration tools, but is not represented with data within the master configuration file.
Physics	PhysicsType^T	M	Physics at individual ports
Crc32	HexDecValue	O	CRC Checksum used to check device description against alteration. Refer to 4.2.

12 Modules

Figure 5 shows how the Modules element is composed. It is typically, but not exclusively, used when the EtherCAT slave device is structured according to the Modular Device Profile (ETG.5001). It describes physical modules (e.g. modular I/O terminals) as well as functional modules with different operation modes (e.g. different synchronization modes).

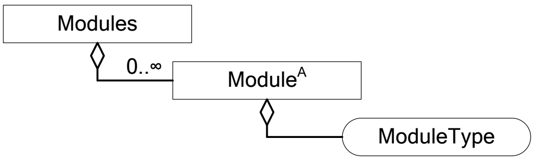

Figure 5: Content of Module

Table 18 describes the content of element Modules.

Table 18: Content description of Modules

Element Name	Data Type	Use	Description
Module^A	ModuleType^T	O	Description of a single module

NOTE: The Modules description can be integrated into the EtherCAT Slave Information (ESI) file (based on EtherCATInfo.xsd) or described in a separate file (based on EtherCATModule.xsd). In the latter case a reference to this file has to be listed in the element *EtherCATInfo:InfoReference*.

Table 19 describes the attributes of Modules.

Table 19: Attributes of Module

Attribute	Data Type	Use	Description
Crc32	HexDecValue	O	CRC Checksum used to check module description against alteration. Refer to clause 4.2.

13 Types

13.1 AccessType

Figure 6 shows how the AccessType type is composed.

Figure 6: Content of AccessType

Table 20 describes the attributes of AccessType.

Table 20: Attributes of AccessType

Attribute	Data Type	Use	Description
ReadRestrictions	xs:NMTOKEN	O	Read access only available in the selected ESM state(s). Access Type shall be rw or ro. PreOP Read access only in PreOP PreOP_SafeOP Read access only in PreOP and SafeOP PreOP_OP Read access only in PreOP and OP SafeOP Read access only in SafeOP SafeOP_OP Read access only in SafeOP and OP OP Read access only in OP configtool: for compatibility reasons also “PreOp” should be accepted and handled the same way as “PreOP”. This allows configuration tools to handle legacy ESI files
WriteRestrictions	xs:NMTOKEN	O	Write access only available in the selected ESM state(s). Access Type shall be rw or wo. PreOP Write access only in PreOP PreOP_SafeOP Write access only in PreOP and SafeOP PreOP_OP Write access only in PreOP and OP SafeOP Write access only in SafeOP SafeOP_OP Write access only in SafeOP and OP OP Write access only in OP configtool: for compatibility reasons also “PreOp” should be accepted and handled the same way as “PreOP”. This allows configuration tools to handle legacy ESI files

13.2 ArrayInfoType

Figure 7 shows how the ArrayInfoType is composed.

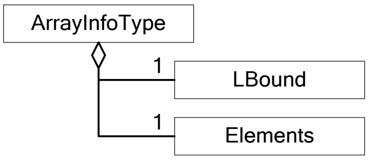

Figure 7: Content of ArrayInfoType

Table 21 describes the content of ArrayInfoType.

Table 21: Content description of ArrayInfoType

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
LBound	xs:integer	M	Index of first array element. For usage see Table 23. Data range: 0 … 255
Elements	xs:integer	M	Number of array elements Data range: 1 … 255 (when used as ARRAY Information) n+1 (when used for ARRAY [0..n] of XYZ) See Table 23 for details.

13.3 DataTypeType

The DataTypeType describes how a data type can be built.

All data types which are used in the Object element shall be defined including base data types as defined in clause 7.

Figure 8 shows how DataTypeType is composed.

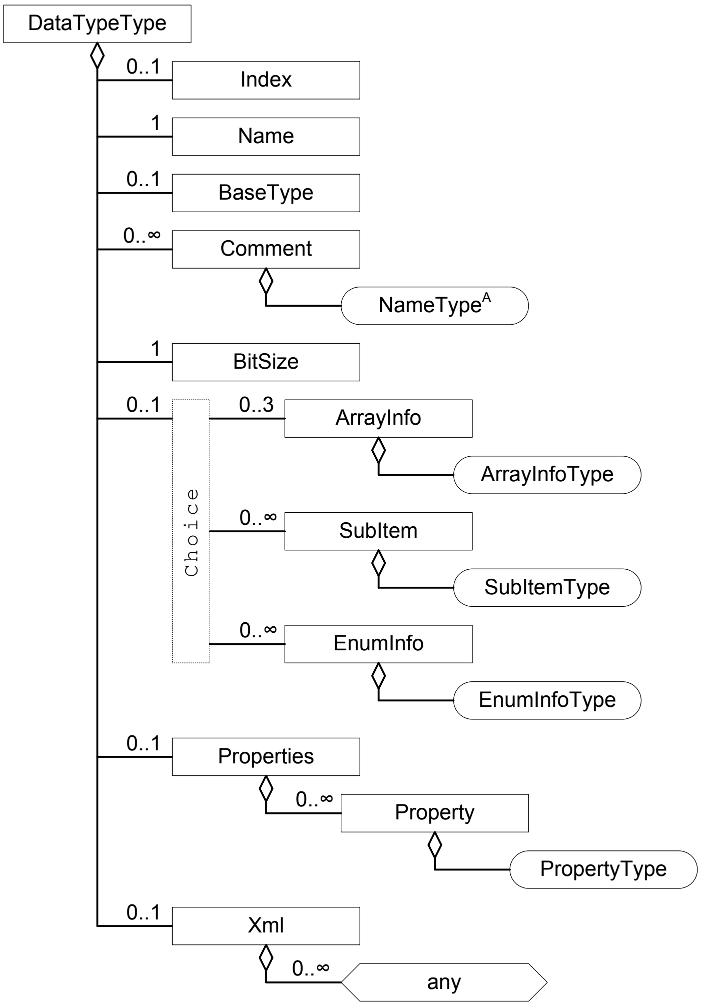

Figure 8: Content of DataTypeType

Table 22 describes the content of DataTypeType.

Table 22: Content description of DataTypeType

Element Name	Data Type	Use	Description
Index	HexDecValue	O	Reserved for future use

Element Name	Data Type	Use	Description
Name	xs:string	M	Name of data type NOTE: For allowed/recommended names refer to Table 23.
BaseType	xs:string	O^D	D: Usage according to Table 23.Specifies the base data type of a complex data type or an object data type For allowed values refer to clause 7.
Comment	$NameType^T$	O	Optional comment on the data type
BitSize	xs:int	M	Length of data type in bit. *BitSize* is calculated according to the data type - BaseType: *BitSize* is the length of the base data type - ENUM information: *BitSize* is the length of bits used for the enumerated values. *BitSize* is used to adopt the Enum length to the object entry length. - ARRAY Information: *BitSize* is the length of the used *BaseType* multiplied by the Number of Elements. - ARRAY: *BitSize* is size of *Subindex* 0 (data type shall be USINT = 8 Bit) plus alignment information plus *BitSize* of ARRAY Information *DataType*. NOTE: *Subindex* 0 plus alignment information is equal to the *BitOffset* of *Subindex* 1. - RECORD: *BitSize* is the sum of Bit of all elements including alignment information, i.e. Subindex 0 (data type shall be USINT = 8 Bit) to last subindex including alignment. NOTE: Information for Complete Access The information *DataType:BitSize* is needed to know the complete size of the object including alignment information. *SubItem:BitSize* is needed to know the size of each element. *SubItem:BitOffset* is needed to know the offset of each element, including alignment information.
ArrayInfo	$ArrayInfoType^T$	O	Used to describe data type ARRAY Information and OCTET_STRING. Refer to Table 23.
SubItem	$SubItemType^T$	O	Describes the elements of a RECORD or ARRAY data type. The order of the SubItem elements reflects the order of the subindices in the device's object dictionary.
EnumInfo	$EnumType^T$	O	Used for data types with exclusive values only
Properties	--	O	General description of additional properties. Can be used for example to define function groups
Properties:Property	PropertyTypeT	O	Property information (refer to clause 13.18)

Element Name	Data Type	Use	Description
Xml	--	O	General XML description for further information possible (similar to properties description above, but more generous)

Data Type Composition

Depending on the data type to be defined different elements are required:

Table 23: Data Type Composition

CoE (Object) Data Type	Help Data Type construction	Mandatory Elements	Description	Name format
Base Data Type (excluding "ARRAY [0..n] OF *XYZ*")		Name BitSize	For allowed base data types refer to clause 7	As defined in clause 7
ARRAY		Name BitSize SubItem	Describes an object of data type ARRAY, i.e.: First element *SubItem* (SI0): *SubItem*:*SubIdx* = 0 Type shall be USINT Second element *SubItem* (SI1-n by using ARRAY Information) **no** *SubItem*:*SubIdx*, Type = DTyyyyARR	Recommended: DTyyyy yyyy = index of object
	ARRAY Information	Name BaseType BitSize ArrayInfo	Describes an object of data type ARRAY, i.e.: SI1-n. *ArrayInfo:LBound* = 1 NOTE: The object data type does not include SI0	Recommended: DTyyyyARR yyyy = index of object
RECORD		Name BitSize SubItem	Describes an object of data type RECORD First Element SubItem (SI0): *SubItem:SubIdx* = 0 Type shall be USINT Following SubItems: *SubItem:SubIdx* > 0	Recommended: DTyyyy yyyy = index of object
ENUM		Name BaseType BitSize EnumInfo	Describes data type for an enumerated object entry. BaseType shall be of either Type USINT, UINT, UDINT, BOOL	Recommended: DTyyyyENxx yyyy = index of object in the index range from 0x800-0xFFF xx = Bitlength of Enum

CoE (Object) Data Type	Help Data Type construction	Mandatory Elements	Description	Name format
ARRAY [0..n] OF *XYZ*		Name BaseType BitSize ArrayInfo	Describes data type for an ARRAY [0..n] OF *XYZ* object entry *ArrayInfo:LBound* = 0 *ArrayInfo:Elements* = n+1. NOTE: Despite ARRAY [0..n] OF *XYZ* are base data types, they shall be defined for usage in the offline object dictionary.	ARRAY [0..n] OF *XYZ* *XYZ* = Base data type according ETG.1020

13.4 DeviceType

DeviceType is used to describe an EtherCAT slave device. Figure 9 shows how the DeviceType is composed.

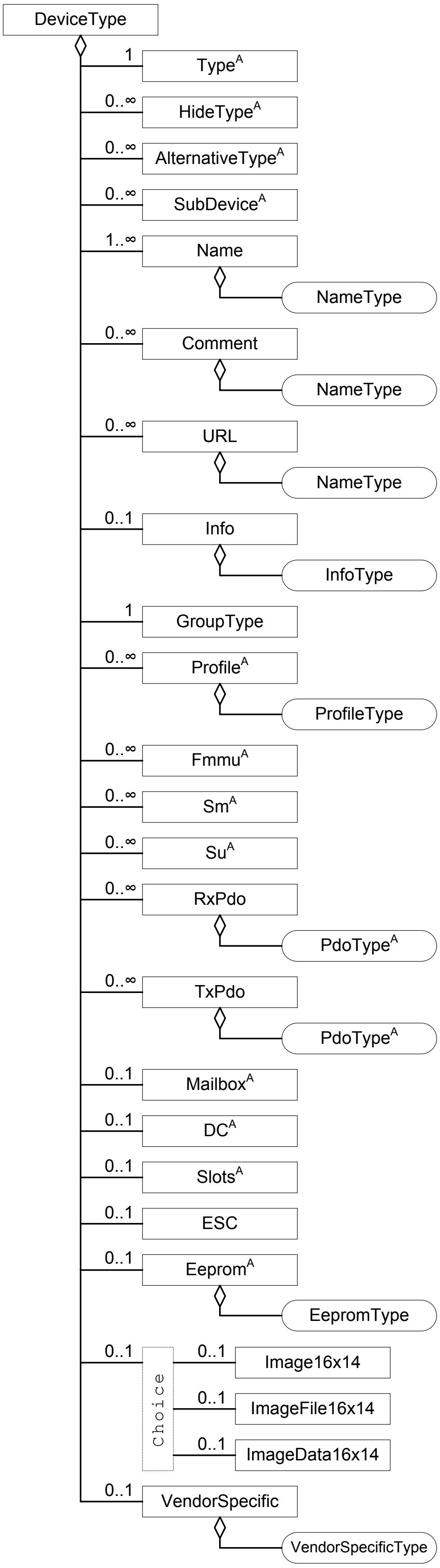

Figure 9: Content of DeviceType

Table 24 describes the content of DeviceType

Table 24: Content description of DeviceType

Element Name	Data Type	Use	Description
Type[A]	xs:string	M	Device identity incl. name, product code, revision no
HideType[A]	xs:string	O	Contains *ProductCode* and may contain *RevisionNo* of device(s) which should not be displayed by a configuration tool any more (e.g. older version of a device) when this (new) device is displayed. NOTE: A configuration tool may support to show also the devices listed by the element *HideType*.
AlternativeType[A]	xs:string	O	Vendor specific configtool: skip element
SubDevice[A]	xs:string	O	Used to display EtherCAT slaves which are built with more than one ESC clearly arranged by a configuration tool. Contains *ProductCode* and *RevisionNo* of ESIs describing additional *SubDevices*.
Name	NameType[T]	M	Detailed name of device shown by a configuration tool (not used for identification)
Comment	NameType[T]	O	Optional comment to describe the device within ESI file (usually not evaluated by tools)
URL	NameType[T]	O	URL for further information on the device. Usually pointing to the vendor’s homepage where up to date ESI files can be downloaded
Info	InfoType[T]	O	Additional information about the device (hardware features of ESC, timeouts)
GroupType	xs:string	M	Reference to a group (described in element *Groups*) to which this device should be assigned to. Name of the handle used in element *Groups:Group:Type*
Profile[A]	ProfileType[T]	O	Description of the used profile and object dictionary including data type definition.
Fmmu[A]	xs:string	O	Definition of FMMU usage Allowed values: ‘Outputs’: FMMU is used for RxPDO ‘Inputs’: FMMU is used for TxPDO ‘MBoxState’: FMMU is used to poll Input Mailbox State (Register 0x080D.0)

Element Name	Data Type	Use	Description
Sm^{A}	xs:string	O	Description of SyncManager including start address and direction. Allowed values: 'MBoxOut': Mailbox Data Master → Slave 'MBoxIn': Mailbox Data Slave → Master 'Outputs': Process Data Master → Slave 'Inputs': Process Data Slave → Master NOTE: Standard SyncManager assignment as specified in ETG.1000, part 4 applies. The first listed SyncManager describes SyncManager0, the following SyncManager describes SyncManager1, etc. If more than one SyncManager of same direction and buffer mode is used attribute *Pdo@Su* is mandatory
Su^{A}	xs:string	O	Defines a timing context by defining different datagrams (possibly different frames) which are identified by this string.
RxPdo	PdoType^{T}	O	Description of RxPDOs (output process data)
TxPdo	PdoType^{T}	O	Description of TxPDOs (input process data)
Mailbox^{A}	--	O	Description of available mailbox protocols (refer to clause 13.4.1)
Dc^{A}	--	O	Description of synchronization mode. This may be Freerun, synchronous with SyncManager event or Distributed Clocks (refer to clause 13.4.2).
Slots^{A}	--	O	Defines the combination possibilities of modules (described in element *Modules*). This may be used when the device supports the Modular Device Profile (ETG.5001). Refer to clause 13.4.3.
ESC	--	O	Init values of ESC watchdog registers (refer to clause 13.4.4)
Eeprom^{A}	EepromType^{T}	O^{D}	D: Use is mandatory Description of SII, e.g. PDI configuration (loaded from ESC during start-up), EEPROM size
Image16x14	xs:string	O^{D}	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Obsolete configtool: skip element
ImageFile16x14	xs:string	O^{D}	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* File path to a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
ImageData16x14	xs:hexBinary	O^{D}	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Hex binary data of a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color

Element Name	Data Type	Use	Description
VendorSpecific	VendorSpecificType^T	O	Vendor specific element of *DeviceType*

Table 25 describes the attributes of Type.

Table 25: Attributes of DeviceType:Type

Attribute	Data Type	Use	Description
ProductCode	HexDecValue	O	Vendor specific product code. Used for identification in conjunction with Vendor ID. (CoE object 0x1018.2) NOTE: The combination of product code and revision number should be unique for one device description and has to match the product code and revision number stored in the EEPROM.
RevisionNo	HexDecValue	O	Vendor specific revision number (CoE object 0x1018.3)
SerialNo	HexDecValue	O	Vendor specific serial number (CoE object 0x1018.4) NOTE: All device with same combination of *ProductCode* and *RevisionNo* have same serial number: write serial number to element *DeviceType:Type@SerialNo* Devices with same combination of *ProductCode* and *RevisionNo* have different serial number: *SerialNo* shall not be used. In this case a serial number shall be written to the SII
CheckProductCode	xs:NMTOKEN	O	This element is used to specify how the product code read from the device's EEPROM is compared against the one of the network configuration (ENI). Allowed values: NONE: Product code is not checked EQ: Product code read from found device shall be equal to the one of the configured device (default value)

Attribute	Data Type	Use	Description
CheckRevisionNo	xs:NMTOKEN	O	This element is used to specify how the Revision Number read from the connected device's EEPROM is compared against the one of the network configuration. Allowed values: NONE: revision number is not checked (default value) EQ: revision number read from found device shall be equal to the one of the configured device EQ_OR_G: revision number read from found device shall be equal or greater than the one of the configured device without distinguishing HW and LW. LW_EQ: low word of revision number read from found device shall be equal with the one of the configured device LW_EQ_HW_EQ_OR_G: low word of revision number read from found device shall be equal with the one of the configured device and the high word shall be equal or greater. HW_EQ: high word of revision number read from found device shall be equal with the one of the configured device HW_EQ_LW_EQ_OR_G: high word of revision number read from found device shall be equal with the one of the configured device and the low word shall be equal or greater.
CheckSerialNo	xs:NMTOKEN	O	This element is used to specify how the serial number read from the connected device's EEPROM is compared against the one of the network configuration. Allowed values: NONE: Serial number is not checked (default value) EQ: Serial number read from found device shall be equal to the one of the configured device
TcSmClass	xs:string	O	Vendor specific configtool: skip element
TcCfgModeSafeOp	xs:boolean	O	Obsolete configtool: use like *InfoType:StateMachine:Behavior@StartToSafeopNoSync*
UseLrdLwr	xs:boolean	O	0: LRW command supported by ESC 1: Indicates the configuration tool to use LRD and LWR commands instead of LRW commands. Shall only be used when ESC does not support the LRW command type.
ModulePdoGroup	xs:int	O	When using elements *Modules/Slots*: Defines the group number to which the PDOs of the this device (described by *Device:RxPdo* and *Device:TxPdo*) are assigned to. Counting starts with '0'. NOTE: The PDOs of a device incl. the PDOs of the modules are listed in the PDO assignment object according to the *ModulePdoGroup* value. In addition to the PDOs of this device, the module PDOs can be assigned to a *ModulePdoGroup* (*Modules:Module:Type@ModulePdoGroup*), too.

Attribute	Data Type	Use	Description
DownloadModuleList	xs:boolean	O	Obsolete configtool: use like *DeviceType:Slots:DownloadModuleIdentList*

Table 26 describes the attributes of HideType.

Table 26: Attributes of DeviceType:HideType

Attribute	Data Type	Use	Description
ProductCode	HexDecValue	O^D	D: Use is mandatory for devices with ESC Product code of the device that will be hidden
RevisionNo	HexDecValue	O^D	D: Use is mandatory for devices with ESC Revision number of the device that will be hidden
ProductRevision	xs:string	O	Obsolete configtool: skip element

Table 27 describes the attributes of AlternativeType.

Table 27: Attributes of DeviceType:AlternativeType

Attribute	Data Type	Use	Description
ProductCode	HexDecValue	O^D	for future use
RevisionNo	HexDecValue	O^D	for future use

Table 28 describes the attributes of SubDevice.

Table 28: Attributes of DeviceType:SubDevice

Attribute	Data Type	Use	Description
ProductCode	HexDecValue	O	Product code of the device(s) that are part of this slave
RevisionNo	HexDecValue	O	Revision number of the device(s) that are part of this slave
PreviousDevice	xs:int	O	Refers to the sub devices as listed in the elements *SubDevice* of the main device with same order: Main device = „0“, first sub device=“1“ … NOTE: This element is not used in a main device.
PreviousPortNo	xs:int	O	Number of logical port of *PreviousDevice* (1, 2, 3; 0 is always IN port) NOTE: This element is not used in a main device.

Table 29 describes the attributes of Profile.

Table 29: Attributes of DeviceType:Profile

Attribute	Data Type	Use	Description
Channel	xs:int	O	Obsolete configtool: skip element

Table 30 describes the attributes of Fmmu.

Table 30: Attributes of DeviceType:Fmmu

Attribute	Data Type	Use	Description
OpOnly	xs:boolean	O	Obsolete configtool: *handle like DeviceType:Sm@OpOnly*
Sm	xs:int	O^D	D: Mandatory if more than one FMMU for the same direction is used to map data to non-consecutive memory areas Assigns this FMMU to a SyncManager NOTE: SyncManager counting starts with 0 NOTE: If SM assigned for the FMMU it shall match with the SM setting in the PDO (refer to Table 83)
Su	xs:int	O	Assigns this FMMU to the related PDO (*Pdo@Su*) and therefore to a SyncManager.

NOTE: The first mentioned (usually topmost) FMMU describes FMMU entity 0 the following FMMU describes FMMU entity 1, etc.

Table 31 describes the attributes of Sm.

Table 31: Attributes of DeviceType:Sm

Attribute	Data Type	Use	Description
MinSize	HexDecValue	O	Minimum SyncManager length in bytes supported by the slave NOTE: May be used to describe the minimum length of the mailbox SyncManagers. This may be checked by a configuration tool.
MaxSize	HexDecValue	O	Maximum SyncManager length in bytes supported by the slave NOTE: May be used to describe the maximum length of the mailbox SyncManagers and/or process data SyncManagers. This may be checked by a configuration tool.
DefaultSize	HexDecValue	O^D	D: Mandatory for Mailbox SyncManager Process Data SyncManager: For process data length the default SyncManager length is calculated based on the default PDO assignment in element *RxPDO* (for Output SyncManager) and *TxPDO* (for Input SyncManager). Default size in bytes of SyncManager NOTE: The configuration tool has to recalculate the SM length if a manual configuration is done.
StartAddress	HexDecValue	O^D	D: mandatory Physical start address of SyncManager NOTE: If a SyncManager is configured for 3-buffer mode the occupied memory is calculated by 3 times of its length. This has to be considered for the calculation of the start address of the following SyncManager.
ControlByte	HexDecValue	O^D	D: mandatory if *Sm@Virtual* = false SyncManager Control Byte (register 0x0804 + y*8; y= number of SyncManager) incl. SyncManager mode and direction

Attribute	Data Type	Use	Description
Enable	HexDecValue	O^D	D: Use is mandatory for mailbox SyncManager. SyncManager Enable Bit (0x0806.0 + y*8) y= number of SyncManager Allowed values: ‘1’: Enabled: SyncManager active controlling memory ‘0’: Disabled: No memory protection by SyncManager For Mailbox SyncManagers: Mailbox Length != 0: Enable = TRUE Mailbox Length == 0: Enable = FALSE For process data SyncManagers: Don’t care NOTE: The configuration tool calculates the process data length and adapts the Enable bit according to the SyncManager length.
OneByteMode	xs:boolean	O	Obsolete configtool: skip element
Virtual	xs:boolean	O	0: SyncManager is not Virtual 1: SyncManager used for configuration only – no use of hardware entity (i.e. no SyncManager registers are used) to ensure data consistency. NOTE: It is used for register access via process data. Register access is always consistence, hence no SM is necessary. In this case PdoType@Virtual shall be “TRUE”, too.
Watchdog	xs:boolean	O	Obsolete configtool: skip element
OpOnly	xs:boolean	O	0: EtherCAT State Machine has to be handled by host controller 1: Master enables SyncManager only in Op state. For all other states the SyncManager is disabled. NOTE: TRUE for devices with digital I/O interface. According to the EtherCAT state machine the outputs shall only be activated in Op state. This is controlled by the host controller application. I/O interface devices do not handle a state machine.
FixedAssignment	xs:boolean	O	Obsolete configtool: skip element

Table 32 describes the attributes of Su.

Table 32: Attributes of DeviceType:Su

Attribute	Data Type	Use	Description

Attribute	Data Type	Use	Description
SeparateSu	xs:boolean	O	0: data of this SyncManager of this device may be sent within a datagram transporting also process data of other slaves 1: Separate SyncUnit for this device, i.e. an extra datagram for this slave is sent. NOTE: For devices which work in 1 Buffer Mode and have a slower cycle time than EtherCAT the SyncManager area might not be updated by the slave for one or more cycles. As result, the WKC might not be incremented and therefore the data of all other slaves addressed by this datagram would also become invalid. By using a separate SyncUnit the data of all other slaves remain valid.
SeparateFrame	xs:boolean	O	0: Data of this device may not be sent with an extra Ethernet frame 1: Data of this device is sent with an extra Ethernet frame
DependOnInputState	xs:boolean	O	For future use
FrameRepeatSupport	xs:boolean	O	0: Device does not support processing of repeated frames within one cycle 1: Device supports processing of repeated frames within one cycle, i.e. SyncManager is read or written multiple times within one cycle. NOTE: A device has to take special care of synchronization behavior when this mode is supported.

Table 33 describes the attributes of Eeprom.

Table 33: Attributes of Eeprom

Element Name	Data Type	Use	Description
AssignToPdi	xs:boolean	O	0: EEPROM access rights are assigned to PDI during state change from Init to PreOp, Init to Boot and while in Boot 1: EEPROM access rights are assigned to PDI in all states except Init

13.4.1 Mailbox

The Mailbox element describes the supported mailbox protocols. Figure 10 shows how the Mailbox element is composed.

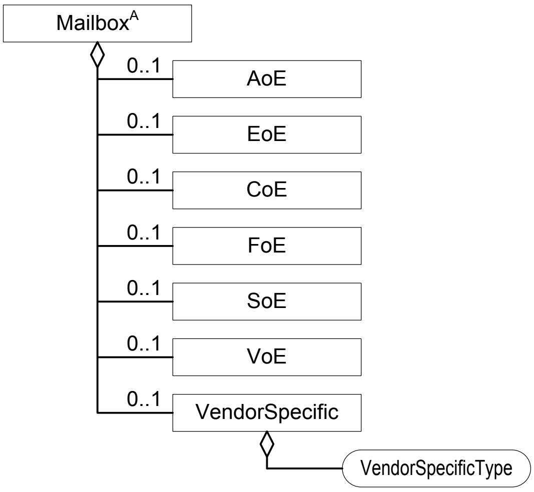

Figure 10: Content of Mailbox

Table 34 describes the attributes of Mailbox.

Table 34: Attributes of DeviceType:Mailbox

Attribute	Data Type	Use	Description
DataLinkLayer	xs:boolean	O	‘0’: Mailbox Data Link Layer not supported (no Mailbox repeat service) ‘1’: Mailbox Data Link Layer supported (Mailbox repeat service); NOTE: Support of Mailbox Data Link Layer (=recovery from lost mailbox frame) is mandatory according to ETG.1000.4 for devices supporting the mailbox service.
RealTimeMode	xs:boolean	O	for future use

Table 35 describes the elements of Mailbox.

Table 35: Content description of DeviceType:Mailbox

Element Name	Data Type	Use	Description
AoE	--	O	not present: Device does not support AoE present: Device supports AoE (ADS over EtherCAT)
EoE	--	O	not present: Device does not support EoE present: Device supports EoE (Ethernet over EtherCAT)
CoE	--	O	not present: Device does not support CoE present: Device supports CoE (CAN application protocol over EtherCAT)
FoE	--	O	not present: Device does not support foE present: Device supports FoE (File Transfer Protocol over EtherCAT)
SoE	--	O	not present: Device does not support SoE present: Device supports SoE (Servo Drive Profile over EtherCAT)
VoE	--	O	not present: Device does not support VoE present: Device supports VoE (Vendor Specific Protocol over EtherCAT)
VendorSpecific	VendorSpecificType^T	O	Vendor specific element of *DeviceType:Mailbox*

13.4.1.1 AoE

The AoE element describes initialization commands for AoE.

Figure 11 describes how the AoE element is composed.

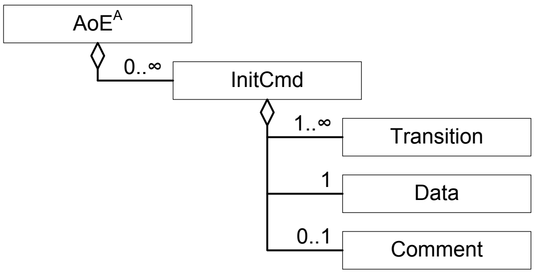

Figure 11: Content of AoE

Table 36 describes the attributes of AoE.

Table 36: Attributes of DeviceType:Mailbox:AoE

Attribute	Data Type	Use	Description
AdsRouter	xs:boolean	O	0: Device does not support ADS router services 1: Device supports ADS router services
GenerateOwnNetId	xs:boolean	O	0: No AoE Net ID is generated 1: The configuration tool generates a AoE Net ID for the device NOTE:This attribute should be set if AoE is supported because the device generally needs a AoE Net ID
InitializeOwnNetId	xs:boolean	O	0: The AoE Net ID is not downloaded in the startup commands 1: The AoE Net ID is downloaded with the start-up commands The attribute *GenerateOwnNetId* shall be set, too.

Table 37 describes the elements of AoE.

Table 37: Content description of DeviceType:Mailbox:AoE

Element Name	Data Type	Use	Description
InitCmd:Transition	xs:NMTOKEN	M	State transitions during which the mailbox protocol specific init command is sent. For allowed state transitions refer to clause 8, excluding transitions from/to Init and Boot
InitCmd:Data	xs:hexBinary	M	Data sent with the ADS init command (excluding Mailbox and AoE Header)
InitCmd:Comment	xs:string	O	Description of the AoE init command

13.4.1.2 EoE

The EoE element describes initialization commands for EoE.

Figure 12 shows how the EoE element is composed.

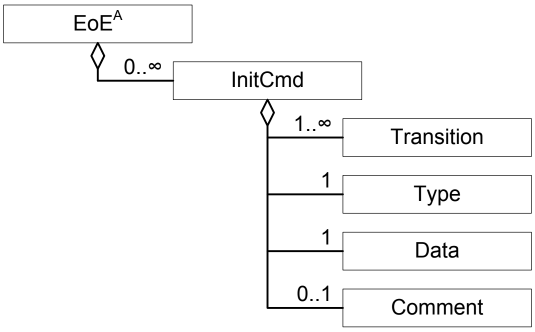

Figure 12: Content of EoE

Table 38 describes the attributes of EoE.

Table 38: Attributes of DeviceType:Mailbox:EoE

Attribute	Data Type	Use	Description
IP	xs:boolean	O	0: no IP address is assigned 1: IP address and virtual MAC address is assigned
MAC	xs:boolean	O	0: No virtual MAC address is assigned 1: Virtual MAC address is assigned via EoE “Set IP Parameter” service
TimeStamp	xs:boolean	O	0: Device does not support time stamping using the DC time stamp 1: Device supports time stamping using the DC time stamp

Table 39 describes the elements of EoE.

Table 39: Content description of DeviceType:Mailbox:EoE

Element Name	Data Type	Use	Description
InitCmd	--	O	Definition of EoE init command
InitCmd:Transition	xs:NMTOKEN	M	State transitions during which the mailbox protocol specific init command is sent. For allowed state transitions refer to clause 8, excluding transitions from/to Init and Boot
InitCmd:Type	xs:int	M	EoE frame type. Allowed values: 0x00 EoE Fragment Request 0x01 Timestamp Response 0x02 IP Parameter Request 0x04 Set MAC Address Filter Request
Init:Cmd:Data	xs:hexBinary	M	Data sent with the EoE command (excluding mailbox and EoE header)
InitCmd:Comment	xs:string	O	Description of init command

13.4.1.3 CoE

If this entry is present the device supports CoE. Figure 13 shows how the CoE element is composed.

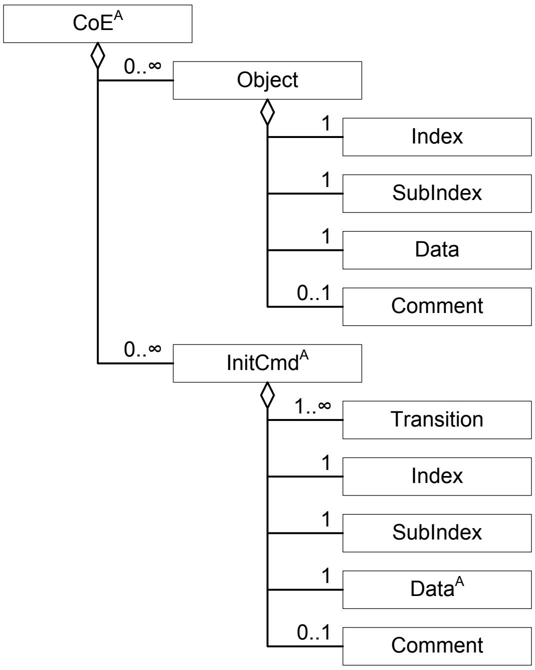

Figure 13: Content of CoE

Table 40 describes the attributes of Device:Mailbox:CoE.

Table 40: Attributes of DeviceType:Mailbox:CoE

Attributes	Data Type	Use	Description
SdoInfo	xs:boolean	O	0: SDO Information Service not supported 1: SDO Information Service supported
PdoAssign	xs:boolean	O	Download of PDO assignment (e.g. 0x1C12, 0x1C13, etc) during start up 0: PDO assignment is not downloaded 1: PDO assignment is downloaded (e.g. set when PDO assignment can be changed) NOTE: Access rights of assign objects shall not be “read only” when *PdoAssign* = true. Padding PDOs that are configured in the PDO assignment are included in the downloaded PDO list.
PdoConfig	xs:boolean	O	Download of PDO Configuration during start up 0: PDO Configuration is not downloaded 1: PDO Configuration is downloaded (set when PDO mapping can be changed) NOTE: Access rights of mapping objects shall not be “read only” when *PdoConfig* = true. Padding PDO entries that are configured in the PDO configuration are included in the downloaded PDO entry list.

Attributes	Data Type	Use	Description
PdoUpload	xs:boolean	O	Device has dynamic process data, i.e. PDO configuration and PDO assignment is uploaded from the device and SyncManager lengths are set according to the calculated PDO length. 0: PDO description taken from ESI and SyncManager length calculated based on the same 1: PDO description uploaded from the slave's object dictionary and SyncManager length calculated based on the same
CompleteAccess	xs:boolean	O	0: SDO complete access not supported 1: SDO complete access supported
EdsFile	xs:string	O	File path of EDS file with Object Dictionary. If available, the EDS file dictionary is used instead of the element *Profile:Dictionary*
DS402Channels	xs:int	O	Obsolete
SegmentedSdo	xs:boolean	O	0: Segmented SDO service not supported 1: Segmented SDO service supported
DiagHistory	xs:boolean	O	0: Diagnosis history object 0x10F3 is not supported 1: Diagnosis history object 0x10F3 is supported NOTE: Refer to ETG.1020, clause Diagnosis

Table 41 describes the elements of CoE.

Table 41: Content description of Device:Mailbox:CoE

Element Name	Data Type	Use	Description
Object	--	O	Obsolete configtool: skip element
Object:Index	xs:int	M	Obsolete configtool: skip element
Object:SubIndex	xs:int	M	Obsolete configtool: skip element
Object:Data	xs:hexBinary	M	Obsolete configtool: skip element
Object:Comment	xs:string	O	Obsolete configtool: skip element
InitCmd^A	--	O	Definition of CoE init command (SDO download service used)
InitCmd:Transition	xs:NMTOKEN	M	State transitions during which the mailbox protocol specific init command is sent. For allowed state transitions refer to clause 8, excluding transitions from/to Init and Boot
InitCmd:Index	HexDecValue	M	CoE object index
InitCmd:SubIndex	HexDecValue	M	CoE object subindex
InitCmd:Data^A	xs:hexBinary	M	CoE object data (excluding mailbox header and CoE header)
InitCmd:Comment	xs:string	O	Comment for this init command

Table 42 describes the attributes of Device:Mailbox:CoE:InitCmd.

Table 42: Attributes of DeviceType:Mailbox:CoE:InitCmd

Attribute	Data Type	Use	Description
Fixed	xs:boolean	O	obsolete configtool: skip element
CompleteAccess	xs:boolean	O	0: Init command shall be sent subindex wise 1: Init command may be sent sub-index wise or via complete access
OverwrittenByModule	xs:boolean	O	0: Always send this init command 1: This init command is not sent when an init command with the same index and subindex is defined in element *Module:Mailbox:CoE:InitCmd*

Table 43 describes the attributes of Device:Mailbox:CoE:InitCmd:Data

Table 43: Attributes of DeviceType:Mailbox:CoE:InitCmd:Data

Attribute	Data Type	Use	Description
AdaptAutomatically	xs:boolean	O	0: Default values defined in *CoE:InitCmd:Data* are sent 1: Default values defined in *CoE:InitCmd:Data* are overwritten by real values, e.g. 0x1C32.02 will be adapted to the cycle time of the EtherCAT master

13.4.1.4 SoE

If this entry is present the device supports SoE.

Figure 14 shows how the SoE element is composed.

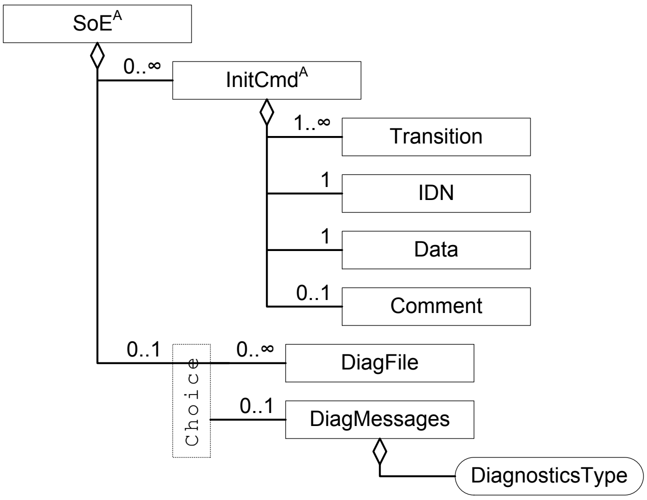

Figure 14: Content of SoE

Table 44 describes the attributes of SoE.

Table 44: Attributes of DeviceType:Mailbox:SoE

Attribute	Data Type	Use	Description
ChannelCount	xs:int	O	Number of supported axis. NOTE: A SoE device can have up to 8 channels.

Attribute	Data Type	Use	Description
DriveFollowsBit3Support	xs:boolean	O	0: SoE Drive does not support the function of Bit 3 of the status word 1: SoE Drive supports the function of Bit 3 of the status word (refer to IEC 61800-7-204)

Table 45 describes the content of SoE.

Table 45: Content description of DeviceType:Mailbox:SoE

Element Name	Data Type	Use	Description
InitCmd	--	O	Definition of SoE init command
InitCmd:Transition	xs:NMTOKEN	M	State transitions during which the mailbox protocol specific init command is sent. For allowed state transitions refer to clause 8, excluding transitions from/to Init and Boot
InitCmd:IDN	xs:int	M	IDN (Ident Number)
InitCmd:Data	xs:hexBinary	M	Data of the IDN (excluding mailbox header and SoE header)
InitCmd:Comment	xs:string	O	Comment for this init command
DiagFile	xs:string	O	Path to DiagHistory file according to EtherCATDiag.xsd NOTE: not the same file than the one for ProfileType:DiagFile. This reference is used for SoE error codes as described at SoE:DiagMessages
DiagMessages	DiagnosticsType^T	O	Definition of diagnosis messages NOTE: in the element SoE the DiagMessages are used to replace SoE error codes by readable text messages. Therefore *SoE:DiagMessages:DiagMessage:TextId* is used as error code and *SoE:DiagMessages:DiagMessage:TextMessage* is used for the message related to this error code.

Table 46 describes the attributes of SoE.

Table 46: Attributes of DeviceType:Mailbox:SoE:InitCmd

Attribute	Data Type	Use	Description
Chn	xs:int	O	Channel to which the init command is sent to. Counting starts with 0 (corresponds to first axis).

13.4.1.5 VoE

For the Vendor Specific Protocol over EtherCAT (VoE) vendor specific information are described in the VoE element.

Figure 15 shows how the VoE element is composed.

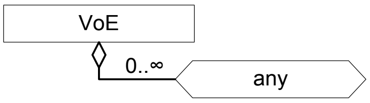

Figure 15: Content of VoE

13.4.2 Dc

Element Dc describes synchronizations modes supported by the device. The mode does not necessarily have to be a DC mode, i.e. using distributed clocks.

Figure 16 shows how the Dc element is composed.

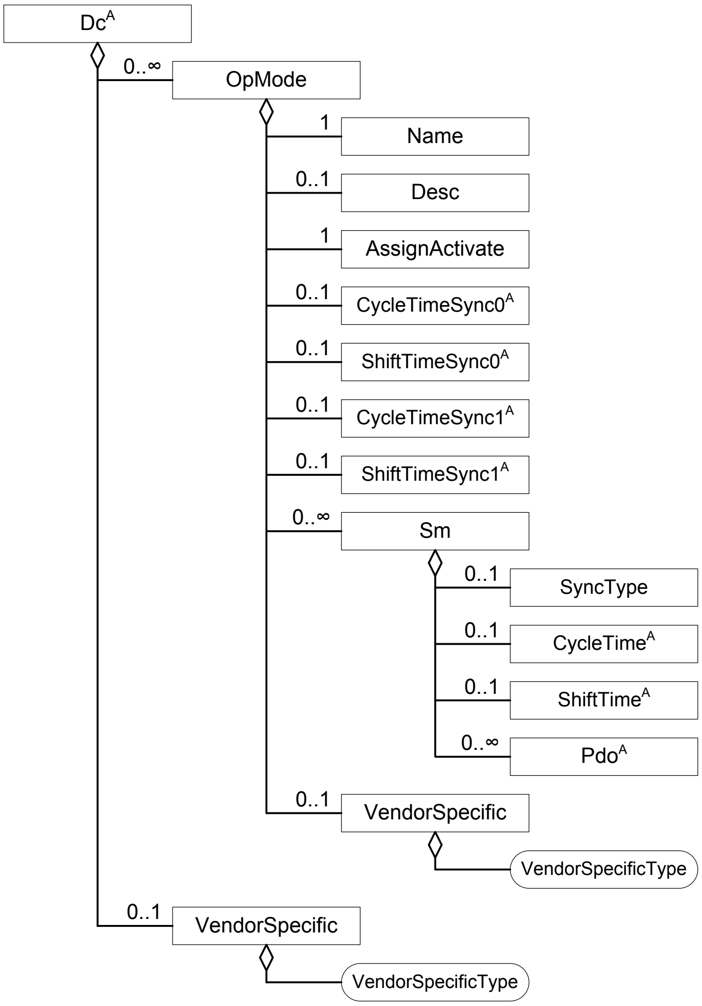

Figure 16: Content of Dc

Table 47 describes the attributes of Dc.

Table 47: Attributes of DeviceType:Dc

Attribute	Data Type	Use	Description
UnknownFRMW	xs:boolean	O	0: Slave processes FRMW commands 1: Slave does not process FRMW commands Shall only be TRUE when ESC does not support this command type.
Unknown64Bit	xs:boolean	O	0: 64 bit time values supported 1: 64 bit time values not supported
ExternalRefClock	xs:boolean	O	0: DC time of device is not triggered by an external source 1: Device synchronizes to an external clock source (e.g. IEEE1588 clock)

Attribute	Data Type	Use	Description
PotentialReferenceClock	xs:boolean	O	0: device cannot be used as reference clock 1: device can be used a reference clock (all necessary registers available) NOTE: For devices supporting the necessary DC registers this attribute should be set to 1. Devices supporting any DC mode automatically can be used as reference clock and this attribute may be omitted.

Table 48 describes the content of DC.

Table 48: Content description of DeviceType:Dc

Element Name	Data Type	Use	Description
OpMode	--	O	Definition of supported operation modes (usually operation modes of synchronization) The first operation mode listed is the default one. NOTE: For definition of operation modes with mailbox init commands the elements *Device:Slots* and Modules are used
OpMode:Name	xs:string	M	Internal handle of operation mode for configuration tool. Every name shall only appear once within one description of a device.
OpMode:Desc	xs:string	O^D	D: Use is mandatory Vendor specific description of operation mode displayed by configuration tool Recommended values: ‘Free Run’ no synchronization ‘SM Synchronous’ synchronized on SyncManager event when process data is written (read) ‘DC-Synchronous‘ synchronized on DC sync event
OpMode:AssignActivate	HexDecValue	M	Value of Latch and Sync Control registers (ESC register 0x0981:0x0980)

Element Name	Data Type	Use	Description
OpMode:CycleTimeSync0 A	xs:int	O	Cycle time of SYNC0 signal (= ESC register 0x09A3:0x09A0) Allowed values: >0: fixed cycle time in nanoseconds 0: cycle time is defined by using the attribute *CycleTimeSync0@Factor* based on sync unit cycle time (= EtherCAT cycle time = cycle time in which the process data of this device is updated) NOTE: A fixed cycle time might be of disadvantage when the sync unit cycle time (i.e. EtherCAT cycle time = cycle time in which the process data of this device is updated) is not a multiple of the fixed time. If *CycleTimeSync0* and *CycleTimeSync0@Factor* are “0” this results in a total cycle time = 0. This may activate special features (single shot).
OpMode:ShiftTimeSync0^A	xs:int	O	Shift (offset) of start time of SYNC0 and SYNC1 signal (as SYNC1 is based on SYNC0) relating to global DC reference in nanoseconds. NOTE: Additionally attribute *ShiftTimeSync0@Factor* may be used to add a value that depends on the sync unit cycle time.
OpMode:CycleTimeSync1 A	xs:int	O	Cycle time of SYNC1 signal. *CycleTimeSync1* shall be an integer multiple of *CycleTimeSync0*. >0: fixed cycle time in nanoseconds 0: cycle time is defined by using the attribute *CycleTimeSync1@Factor* based on either *CycleTimeSync0* or sync unit cycle time (= EtherCAT cycle time = cycle time in which the process data of this device is updated) NOTE: If *CycleTimeSync1* is n * *CycleTimeSync0* with n>1, the SYNC1 signal generation always starts with the omitted signals (refer to Figure 17). Element “CycleTime1” of the ENI file (= ESC register 0x09a4:0x09a7) is calculated as follows: *CycleTimeSync1 - CycleTimeSync0 + ShiftTimeSync1* If *CycleTimeSync1* and *CycleTimeSync1@Factor* are “0” this results in a total cycle time = 0. This may activate special features (single shot).
OpMode:ShiftTimeSync1^A	xs:int	O	Shift (offset) of start time of SYNC1 signal relating to start time of SYNC0 signal in nanoseconds. *ShiftTimeSync1* shall be smaller than *CycleTimeSync0*.
OpMode:Sm^A	--	O	
OpMode:Sm:SyncType	xs:int	O	obsolete configtool: skip element

Element Name	Data Type	Use	Description
OpMode:Sm:CycleTime^A	xs:int	O	obsolete configtool: skip element
OpMode:Sm:ShiftTime^A	xs:int	O	obsolete configtool: skip element
OpMode:Sm:Pdo^A	HexDecValue	O	PDOs assigned to the named SyncManager (*OpMode:Sm:Pdo@Sm*) when this operation mode is selected. NOTE: PDOs assigned by this element have to be defined in the elements *RxPdo/TxPdo*
OpMode:VendorSpecific	VendorSpecific Type^T	O	Vendor specific element of *OpMode*
VendorSpecific	VendorSpecific Type^T	O	Vendor specific element of *DeviceType:Dc*

Figure 17 shows the SYNC1 signal generation behavior if *CycleTimeSync1* = n * *CycleTimeSync0* with n > 1. The first SYNC1 signal always occurs with the n^th SYNC0 signal and never with the first SYNC0 signal.

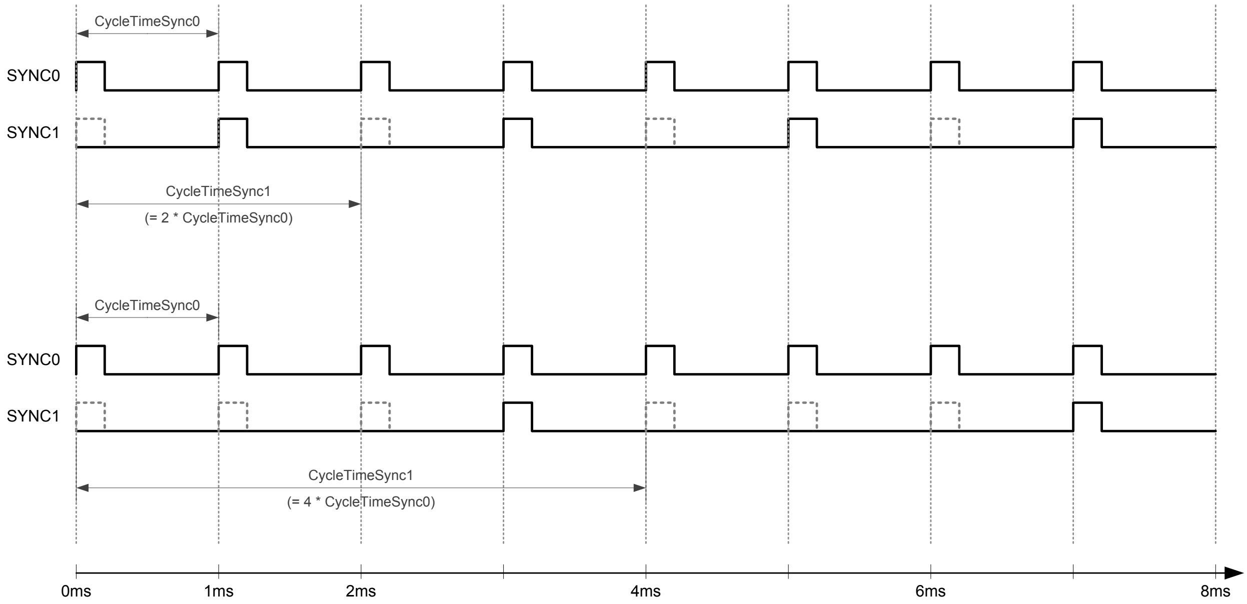

Figure 17: SYNC1 signal generation starts with first omitted signal (example)

Table 49 describes the attributes of *Dc:CycleTimeSync0*.

Table 49: Attributes of Dc:OpMode:CycleTimeSync0

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
Factor	xs:int	O	Used to define SYNC0 signal cycle based on the sync unit cycle time (= EtherCAT bus cycle time = cycle time in which the process data of this device is updated). Sign of value indicates mathematical operation: Positive value: SYNC0 cycle time = sync unit cycle time * |*Factor*| Negative value: SYNC0 cycle time = sync unit cycle time / |*Factor*| NOTE: If *CycleTimeSync0* and *CycleTimeSync0@Factor* are "0" this results in a total cycle time = 0. This may activate special features (single shot).

Table 50 describes the attributes of *Dc:ShiftTimeSync0*.

Table 50: Attributes of Dc:OpMode:ShiftTimeSync0

Element Name	Data Type	Use	Description
Factor	xs:int	O	Additional value to the element ShiftTimeSync0 depending on the sync unit cycle time (= EtherCAT bus cycle time = cycle time in which the process data of this device is updated). Both values are summed up to the total shift time by the configuration tool. Sign of value indicates mathematical operation: Positive value: *ShiftTimeSync0* = sync unit cycle time * |*Factor*| Negative value: *ShiftTimeSync0* = sync unit cycle time / |*Factor*|
Input	xs:boolean	O	0: SyncSignal of ESC is shifted in positive direction in relation to the global/master SyncSignal. Used when slave is synchronized for a synchronous output event 1: DC timing used for inputs - consider input DC timing. Used when input latch event shall be as shortly before the frame collects input data.
OutputDelayTime	xs:int	O	Defines the delay between SYNC0 signal and the actualization of the outputs on the hardware Corresponds to CoE Object 0x1C32.09
InputDelayTime	xs:int	O	Defines the delay between the SYNC0 signal and the actualization of the inputs on the hardware Corresponds to CoE Object 0x1C33.09

Table 51 describes the attributes of Dc:CycleTimeSync1.

Table 51: Attributes of Dc:OpMode:CycleTimeSync1

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
Factor	xs:int	O	Used to define cycle time of SYNC1 signal as a multiple of either the cycle time of SYNC0 signal or the sync unit cycle time (= EtherCAT bus cycle time = cycle time in which the process data of this device is updated). Sign of value indicates mathematical operation: Positive value: *CycleTimeSync1* = |*Factor*| * **SyncCycleTime0** Negative value: *CycleTimeSync1* = |*Factor*| * **sync unit cycle time** The cycle time of SYNC1 signal shall be an integer multiple of the cycle time of SYNC0 signal. This also applies if cycle time of SYNC1 is defined based on sync unit cycle time. NOTE: If *CycleTimeSync1* and *CycleTimeSync1@Factor* are “0” this results in a total cycle time = 0. This may activate special features (single shot).

Table 52 describes the attributes of Dc:ShiftTimeSync1.

Table 52: Attributes of Dc:OpMode:ShiftTimeSync1

Element Name	Data Type	Use	Description
Factor	xs:int	O	For future use
Input	xs:boolean	O	0: SyncSignal of ESC is shifted in positive direction in relation to the global/master SyncSignal. Used when slave is synchronized on a synchronous output event 1: DC timing used for inputs - consider input DC timing. Used when input latch event shall be as shortly before the frame collects input data.
OutputDelayTime	xs:int	O	Defines the delay between SYNC1 signal and the actualization of the outputs on the hardware
InputDelayTime	xs:int	O	Defines the delay between the SYNC1 signal and the actualization of the inputs on the hardware

Table 53 describes the attributes of Sm.

Table 53: Attributes of Dc:OpMode:Sm

Element Name	Data Type	Use	Description
No	xs:int	M	Number of SyncManager described by this element starting with 0 for first SyncManager. Counting starts with 0.

Table 54 describes the attributes of CycleTime.

Table 54: Attributes of Dc:OpMode:Sm:CycleTime

Element Name	Data Type	Use	Description
Factor	xs:int	O	obsolete configtool: skip element

Table 55 describes the attributes of CycleTime.

Table 55: Attributes of Dc:OpMode:Sm:ShiftTime

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
MinAfterSync	xs:int	O	obsolete configtool: skip element
MinBeforeFrame	xs:int	O	obsolete configtool: skip element

Table 56 describes the attributes of Sm:Pdo.

Table 56: Attributes of Dc:OpMode:Sm:Pdo

Element Name	Data Type	Use	Description
OSFac	xs:int	O	t.b.d.

13.4.3 Slots

A slot is used as a place holder for a module. The element Slots describes the allowed combinations of modules as they are described in the element *Description:Modules*. Several Slots can be grouped together to one SlotGroup.

For further information on how to use Slots and Modules refer to the Modular Devices Profile (ETG.5001).

Figure 18 shows how the Slots element is composed.

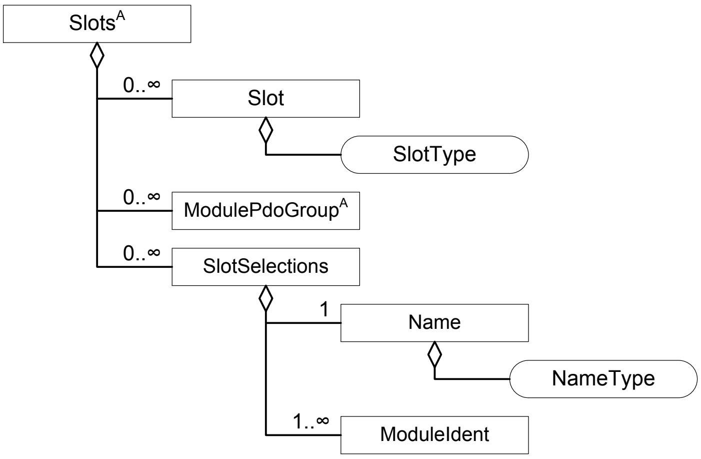

Figure 18: Content of Slot

Table 57 describes the attributes of DeviceType:Slots.

Table 57: Attributes of DeviceType:Slots

Attribute	Data Type	Use	Description
MaxSlotCount	HexDecValue	O	Maximum number of slots for this device
MaxSlotGroupCount	HexDecValue	O	Maximum number of *SlotGroups* possible (e.g. for a DS402 drive there might be 8 *SlotGroups* at the maximum with 0x800 indexes per *SlotGroup*).
SlotPdoIncrement	HexDecValue	O^D	D: Mandatory if *SlotGroupPdoIncrement* is not used PDO Index increment, i.e. distance between the PDO start indexes, of two consecutive slots NOTE: The EtherCAT configuration tool increments the PDO number of the module's PDOs with the *SlotPdoIncrement* multiplied with the module's position.

Attribute	Data Type	Use	Description
SlotGroupPdoIncrement	HexDecValue	O^D	D: Mandatory if *SlotPdoIncrement* ist not used Distance between the PDO start indexes of two consecutive *SlotGroups* The EtherCAT configuration tool increments the PDO number of the module's PDOs with the *SlotGroupPdoIncrement* multiplied with the module's position.
SlotIndexIncrement	HexDecValue	O^D	D: Mandatory if *SlotGroupIndexIncrement* ist not used Object Index increment, i.e. distance between the object (=PDO entry) indexes of two consecutive slots. The index of the objects of the module's objects in the dictionary and entries in the PDO definition is incremented by the *SlotIndexIncrement* multiplied with the module's position
SlotGroupIndexIncrement	HexDecValue	O^D	D: Mandatory if *SlotIndexIncrement* is not used Distance between the start index of two consecutive *SlotGroups* (e.g. with DS402 the *SlotGroupPdoIncrement* might be 0x800)
IdentifyModuleBy	xs:NMTOKEN	O	obsolete configtool: skip element
DownloadModuleIdentList	xs:boolean	O	0: module ident list is not downloaded to the device 1: module ident list 0xF030 is downloaded to the device during PreOp to SafeOp state change
DownloadModuleAddressList	xs:boolean	O	0: module address list is not downloaded to the device 1: module address list 0xF020 is downloaded to the device during PreOp to SafeOp state change

Table 58 describes the content of Slot.

Table 58: Content description of DeviceType:Slots

Element Name	Data Type	Use	Description
Slot	$SlotType^T$	O	Slot description
$ModulePdoGroup^A$	xs:string	O	Describes the alignment of each *ModulePdoGroup*, e.g. if all ModulePdoGroups shall be word aligned
SlotSelections	--	O	Contains a predefined set of modules. These sets can be displayed by a Configuration tool. When one set is selected the defined modules are automatically plugged to the slots in the given order One element *SlotSelections* is used for one predefined set.
SlotSelections:Name	$NameType^T$	M	Name of the module set

Element Name	Data Type	Use	Description
SlotSelections:ModuleIdent	HexDecValue	M	1 to n ModuleIdent elements to be connected to the slots of the device in case this set is selected. Order of ModuleIdent elements is used.

Table 59 describes the attributes of DeviceType:Slots:ModulePdoGroup.

Table 59: Attributes of DeviceType:Slots:ModulePdoGroup

Element Name	Data Type	Use	Description
Alignment	xs:int	O^D	D: Mandatory for every group value used in element *Module:Type:ModulePdoGroup* Alignment for *ModulePdoGroup* in bytes (e.g. 2 for word alignment) Counting starts with ‘0’: First entry describes alignment for *ModulePdoGroup* 0, second for *ModulePdoGroup* 1, etc.) NOTE: *Alignment* only applies on the PDO, not the entries. Alignment within the PDO is made by padding bit entries (index = 0)
RxPdo	HexDecValue	O^D	D: Mandatory for every group value used in element *Module:Type:ModulePdoGroup* Used index for alignment PDO defined by attribute *RxPdo*, i.e. 0x1600 – 0x17FF NOTE: Indices of alignment PDOs shall be used unique i.e. shall not be defined more than once.
TxPdo	HexDecValue	O^D	D: Mandatory for every group value used in element *Module:Type:ModulePdoGroup* Used index for alignment PDO defined by attribute *TxPdo*, i.e. 0x1A00 – 0x1BFF NOTE: Indices of alignment PDOs shall be used unique i.e. shall not be defined more than once.

13.4.4 ESC

Element ESC describes initialization values for ESC watchdog registers and the offset values for RW commands (FPRW, APRW). Default values are ESC specific (ESC init values). These are overwritten if element is available.

If Watchdog Time is set to 0x0000, the Watchdog is disabled.

Figure 19 shows how the ESC element is composed.

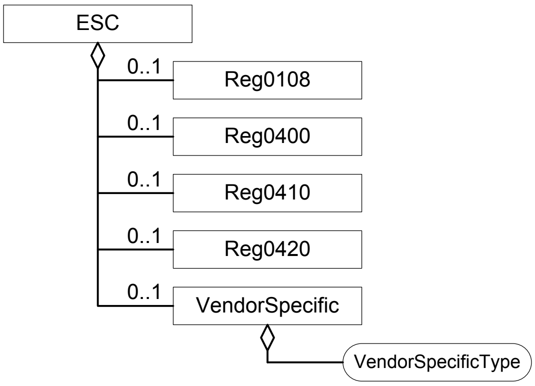

Figure 19: Content of ESC

Table 60 describes the content of ESC.

Table 60: Content description of DeviceType:ESC

Element Name	Data Type	Use	Description
Reg0108	HexDecValue	O	Init value of register Physical Read/ Write Offset 0x0108 Offset of R/W Commands (FPRW/ APRW) between Read address and Write address. RD_ADR = ADR and WR_ADR = ADR + R/W-Offset. Shall be used when the Physical Read/Write Offset should be unequal 0. NOTE: Command is sent before process data communication starts (IP)
Reg0400	HexDecValue	O	Init value of register Watchdog Divider 0x0401:0x400: Number of 25 MHz tics (minus 2) that represents the basic watchdog increment (Default value of this register is 100µs = 2498) Might be used when Watchdog Divider should be different from the default value. Default value: If element is not present no value is written to register 0x400. In that case the default value of the ESC is not overwritten. NOTE: Command is sent before process data communication starts (IP)
Reg0410	HexDecValue	O	Init value of register Watchdog Time PDI 0410: Number of basic watchdog increments (Default value with Watchdog divider 100µs means 100ms Watchdog). Might be used when Watchdog Time PDI should be different from the default value. Default value: If element is not present no value is written to register 0x410. In that case the default value of the ESC is not overwritten. NOTE: Command is sent before process data communication starts (IP)
Reg0420	HexDecValue	O	Init value of register Watchdog Time Process Data 0x0421:0x420: Number of basic watchdog increments. (Default value with Watchdog divider 100µs means 100ms Watchdog) Might be used when Watchdog Time Process Data should be different from the default value. Default value: If element is not present no value is written to register 0x420. In that case the default value of the ESC is not overwritten. NOTE: Command is sent before process data communication starts (IP)
VendorSpecific	VendorSpecificType^T	O	Vendor specific element of *DeviceType:ESC*

13.5 DiagnosticsType

Figure 20 shows how DiagnosticsType is composed.

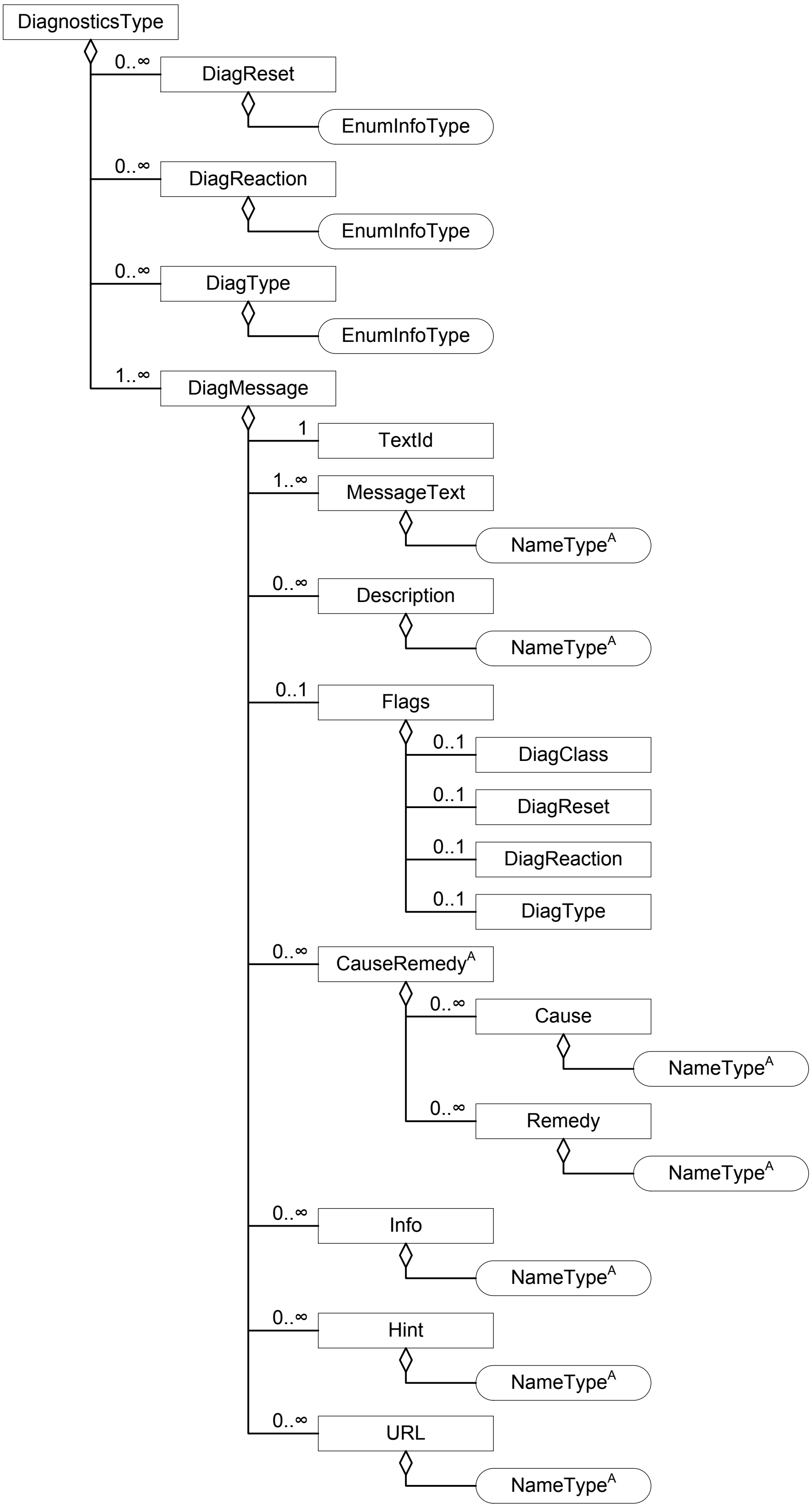

Figure 20: Content of DiagnosticsType

Table 61 describes the content of DiagnosticsType.

Table 61: Content description of DiagnosticsType

Element Name	Data Type	Use	Description
DiagReset	EnumInfoType	O	Definition of a reset instruction. A diagnosis message can refer to this *DiagReset*. Allowed values (for element *Enum*): 0: Reset not possible, fatal hardware error 1: Cold start required 2: Trigger device reset 10: State change to Init required 11: State change to PreOp required 12: State change to SafeOp required >100: Vendor Specific
DiagReaction	EnumInfoType	O	Definition of the reaction in the device if this message is shown, e.g. TorqueOff, ClosedLoopRamp, … A diagnosis message can refer to this *DiagReaction*.
DiagType	EnumInfoType	O	Definition of different diagnosis types, e.g. runtime error, parameter error, … A diagnosis message can refer to this *DiagType*.
DiagMessage	--	O	Definition of one specific diagnosis message
DiagMessage :TextId	HexDecValue	M	Identifier for the diagnosis message
DiagMessage:MessageText	NameType^T	M	Short diagnosis message For using parameters as described in ETG.1020 the following specifiers are allowed: %c, %d, %u, and %s. If the order of the parameters is not the order of the parameters within the message the position can be changed by adding ":n" to the specifier. Counting starts with 0. Example %u:2 shows the third parameter as unsigned value in defined message.
DiagMessage:Description	NameType^T	O	Detailed description of the message
DiagMessage:Flags	--	O	
DiagMessage:Flags: DiagClass	xs:NMTOKEN	O	Allowed values: error warning information
DiagMessage:Flags: DiagReset	xs:int	O	Enum value (referrer) of the pre-defined *DiagReset* instruction
DiagMessage:Flags: DiagReaction	xs:int	O	Enum value (referrer) of the pre-defined *DiagReaction*
DiagMessage:Flags: DiagType	xs:int	O	Enum value (referrer) of the pre-defined *DiagType*

Element Name	Data Type	Use	Description
DiagMessage: CauseRemedy[A]	--	O	Pairs of cause and remedy in conjunction to this diagnosis message.
DiagMessage: CauseRemedy:Cause	NameType[T]	O	Cause of
DiagMessage: CauseRemedy:Remedy	NameType[T]	O	way to resolve this error
DiagMessage:Info	NameType[T]	O	More information about the error / message, e.g. reference to chapter in documentation, …
DiagMessage:Hint	NameType[T]	O	Hints about the error / message. In case of no clear cause or remedy is known this might help the end user to find reasons for the error / message. Also what happens if an error is ignored might be describend here.
DiagMessage:URL	NameType[T]	O	URL that containing further information

Table 62 describes the attribute of DiagMessage:CauseRemedy.

Table 62: Attribute description of DiagMessage : CauseRemedy

Element Name	Data Type	Use	Description
Idx	xs:string	M	assigns cause description to remedy description

13.6 DictionaryType

DictionaryType describes data types used to describe entries in the object dictionary (element *DataType*) and the object dictionary itself (element *Object*).

The element *Device:Profile:Dictionary* is only used by the configuration tool to display the object dictionary offline. This gives the end user the possibility to adopt device specific parameters (easier). If the elemenet *Profile:Dictionary* is supported at least all mandatory objects shall be described herewith.

Figure 21 shows how the DictionaryType element is composed.

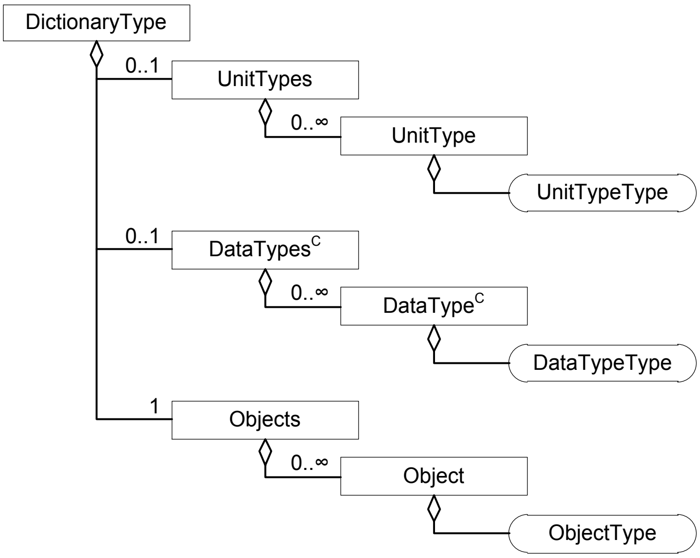

Figure 21: Content of Dictionary:Object

Table 63 describes the content of DictionaryType.

Table 63: Content description of DictionaryType

Element Name	Data Type	Use	Description
UnitTypes	--	O	Definition of unit types that can be used in element *DictionaryType:Objects:Object:Info:Unit*
UnitTypes:UnitType	UnitTypeType^T	O	Definition of one unit type that can be used in element *DictionaryType:Objects:Object:Info:Unit*
DataTypes	--	O^D	D: Mandatory if objects are defined in element *Objects* List of all data types. Base Data Types and complex data types which are used in the element *Objects* shall be defined. Complex data types shall be based on Base Data Types (refer to clause 7).
DataTypes:DataType	DataTypeType^T	O	Description of a data type which is used in the element *Objects*
Objects	--	M	List of Objects of the device's object dictionary. Used by configuration tool to display Object Dictionary offline. NOTE: If element *Dictionary:Objects* is supported at least the mandatory objects are available. Optional objects may not be available in the offline object dictionary.
Objects:Object	ObjectType^T	O	Describes one object of the device's object dictionary

13.7 EepromType

The EtherCAT Slave Controller loads its configuration data (e.g. PDI configuration) during start up from the Slave Information Interface which usually is implemented as EEPROM. The *EepromType* describes at least the mandatory EEPROM data or more.

EEPROM data for general communication, for bootstrap mode and optional data are distinguished. The Eeprom Element is mandatory for the ESI file of slaves with ESC.

Additional information is described by Categories. Each category is identified by a Category number (*CatNo*).

For additional information on the SII structure refer to clause ETG.1020, clause SII and ETG.1000.

Figure 22 shows how the EepromType is composed.

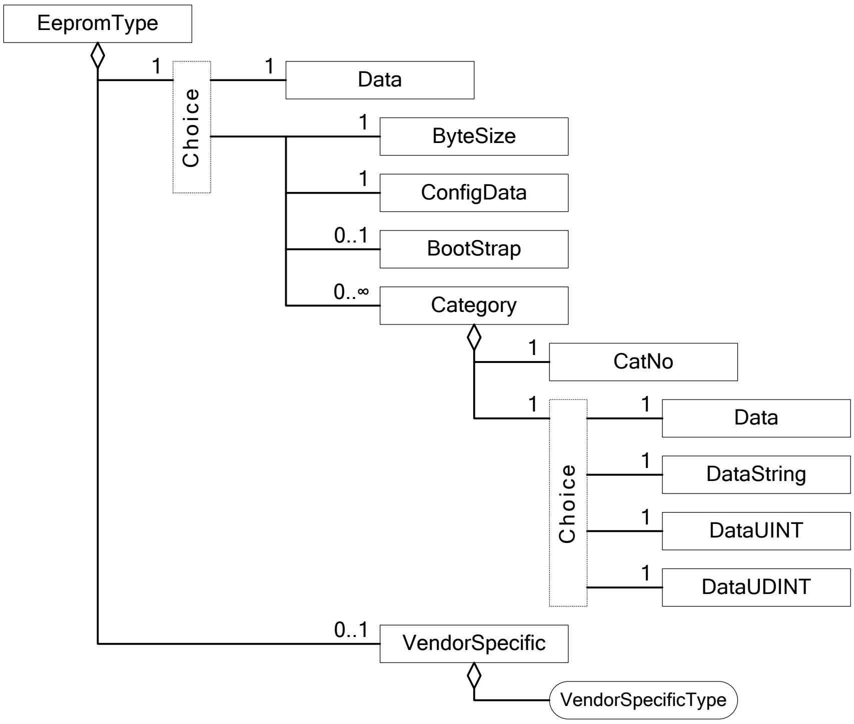

Figure 22: Content of EepromType

Table 64 describes the content of EepromType.

Table 64: Content description of EepromType

Element Name	Data Type	Use	Description
Data	xs:hexBinary	M^D	D: either *Data* or (*ByteSize* and *ConfigData* and *BootStrap* and *Category*) Complete EEPROM data. Length is implicit.
ByteSize	xs:int	M^D	D: either *Data* or (*ByteSize* and *ConfigData* and *BootStrap* and *Category*) Byte size of connected EEPROM device
ConfigData	xs:hexBinary	M^D	D: either *Data* or (*ByteSize* and *ConfigData* and *BootStrap* and *Category*) Data of first 7 words of EEPROM (Configuration Area). Following “0” may be omitted NOTE: the value for Configured Station Alias in the element *Eeprom:ConfigData* should be 0x0000 as the Configured Station Alias value normally is changed by the user using a master/configuration tool if necessary.
BootStrap	xs:hexBinary	O^D	D: either *Data* or (*ByteSize* and *ConfigData* and *BootStrap* and *Category*) Start address and length of mailbox SyncManagers (EEPROM data 0x0017:0x0014). Indicates that device supports bootstrap. NOTE: Settings shall be consistent with corresponding EEPROM values.

Element Name	Data Type	Use	Description
Category[A]	--	O[D]	D: either *Data* or (*ByteSize* and *ConfigData* and *BootStrap* and *Category*) Describes SII category information according to ETG.1000, part 6. NOTE: A configuration tool that is able to write EEPROM data shall write the additional categories based on the information in the ESI file, i.e. element *Device:Sm, Device:Fmmu, Device:Dc*, etc.
Category:CatNo	xs:int	M	Available categories described by the Category type number (ETG.1000, part 6 and ETG.1020) NOTE: A configuration tool that supports EEPROM download shall download the *CatNo* and corresponding data when SII content is generated. This shall be done even if the *CatNo* is unknown. CatNo 0x0001...0x0009: device specific categories which must not be overwritten CatNo 0x0800...0x0FFF: vendor specific All other CatNo reserved
Category:Data	xs:hexBinary	M[D]	D: either *Data* or *DataString* or *DataUINT* or *DataUDINT* is mandatory Category data represented in binary format
Category:DataString	xs:string	M[D]	D: either *Data* or *DataString* or *DataUINT* or *DataUDINT* is mandatory Category data represented in STRING format
Category:DataUINT	xs:int	M[D]	D: either *Data* or *DataString* or *DataUINT* or *DataUDINT* is mandatory Category data represented in UINT format
Category:DataUDINT	xs:int	M[D]	D: either *Data* or *DataString* or *DataUINT* or *DataUDINT* is mandatory Category data represented in UDINT format
VendorSpecific	VendorSpecificType[T]	O	Vendor specific EEPROM data which is only evaluated by a vendor specific configuration tool.

NOTE: Data format of EEPROM is little endian (e.g. ConfigData byte order: 0x0140, 0x0141, 0x0150, 0x0151, etc.)

Table 65 describes the attributes of Category.

Table 65: Attributes of EEPROMType:Category

Attribute	Data Type	Use	Description
PreserveOnlineData	xs:boolean	O	0: Data of this category may be overwritten by configuration tool when downloading EEPROM data 1: Data of this category shall not be changed in EEPROM NOTE: This can be used e.g. for devices with calibration information within a category that should not be overwritten by default data.

13.8 EntryType

EntryType describes an Entry of a PDO Mapping. Figure 23 shows how the EntryType element is composed.

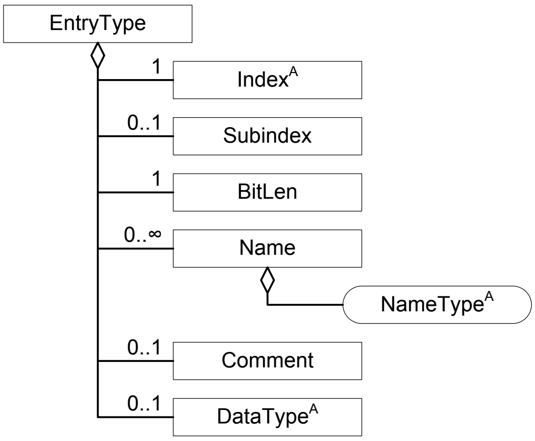

Figure 23: Content of EntryType

Table 66 describes the content of EntryType.

Table 66: Content description of EntryType

Element Name	Data Type	Use	Description
Index^A	HexDecValue	M	PDO index of mapped object NOTE: For padding PDO entries use Index = 0. When no Modular Device Profile (ETG.5001) is used: RxPDO/TxPDO entries: Index area 0x2000 to 0x5FFF for vendor specific objects or 0x6000 to 0x9FFF according to Device profile (e.g. CiA402). When Modular Device Profile used: RxPDO entries: Index area 0x7000 to 0x7FFF TxPDO entries: Index area 0x6000 to 0x6FFF
SubIndex	HexDecValue	O^D	D: Use is mandatory if *Index* != 0 PDO subindex of mapped object Value Range: 0…255 NOTE: For padding PDOs use index = 0. If Index is a variable Type, enter SubIndex=0
BitLen	xs:int	M	Bit length of mapped object Value (n) might be smaller than the actual bit size of the mapped object/entry. In that case bit 0-n shall be mapped as process data.
Name	NameType^T	O^D	D: Use is mandatory if *Index* != 0 Name of the mapped object
Comment	xs:string	O	Interpretation of process data value which may be shown by configuration tool (e.g. Diagnosis Message Object 0x1f03:4: “1: New Diag messages available, 0: No new Diag Messages available”)
DataType^A	xs:string	O^D	D: Use is mandatory if *Index* != 0 Data type of the mapped object Only Base data types (no data type STRING(n)) are allowed, refer to clause 7.

Table 67 describes the attributes of Index.

Table 67: Attributes of EntryType:Index

Attribute	Data Type	Use	Description
DependOnSlot	xs:boolean	O	0: The index is not adapted 1: The index is adapted depending on the slot number and the *SlotIndexIncrement* value in the element *Slots* of the device description NOTE: Used for PDO Mapping entries in element *Modules*
DependOnSlotGroup	xs:boolean	O	0: The index is not adapted 1: The index is adapted depending on the slot group number and the *SlotGroupIndexIncrement* value in the element Slots of the device description NOTE: Used for PDO Mapping entries in element *Modules*

Table 68 describes the attributes of DataType.

Table 68: Attributes of EntryType:DataType

Attribute	Data Type	Use	Description
DScale	xs:string	O	Only if *Entry:DataType* is INT: This attribute is used to specify the range of the input data. Allowed values in high commas: ‘+/-10’: scale value from -10 to +10 ‘0-10’: scale value from 0 to +10 ‘0-20’: scale value from 0 to +20 ‘4-20’: scale value from +4 to +20 ‘0.1°’: multiply value with 0.1 ‘0-10(16)’: scale value from 0 to +10 ‘0-20(16)’: scale value from 0 to +20 ‘0,01°’ , scale value in 0,01° ‘0-5’: scale value from 0 to +5 ‘0-30’: scale value from 0 to +30 ‘0-50’: scale value from 0 to +50 ‘+/-5’: scale value from -5 to +5 ‘+/-2,5’: scale value from -2,5 to +2,5 ‘+/-100’: scale value from -100 to +100 ‘0-5(16)’: scale value from 0 to +5 ‘0-30(16)’: scale value from 0 to +30 ‘0-50(16)’: scale value from 0 to +50 ‘+/-75mV’: scale value from -75mV to+75mV
SwapData	xs:NMTOKEN	O	Defines swapping of PDO entry value for displaying. Allowed values: Swap_HB_LB = swap hi and lo bytes Swap_HW_LW = swap hi and lo words Swap_HB_LB_HW_LW = swap hi and lo words as well as hi and lo bytes

13.9 EnumInfoType

The EnumInfoType describes how an enum is built. Figure 24 shows how the EnumInfoType is composed.

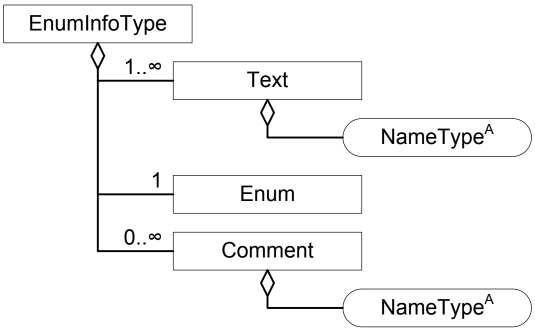

Figure 24: Content of EnumInfoType

Table 69 describes the content of EnumInfoType.

Table 69: Content of EnumInfoType

Element Name	Data Type	Use	Description
Text	NameType^T	M	*Enum* Entry Text
Enum	xs:int	M	Enum Entry Number NOTE: Enum number shall be unique within one ENUM element
Comment	NameType^T	O	Description of *Enum* entry

13.10 GroupType

Devices can be grouped together. This may be useful for configuration tools to display devices of one vendor or device type structured under the *GroupType* name in a tree view.

GroupType describes the device group. Figure 25 shows how the GroupType is composed.

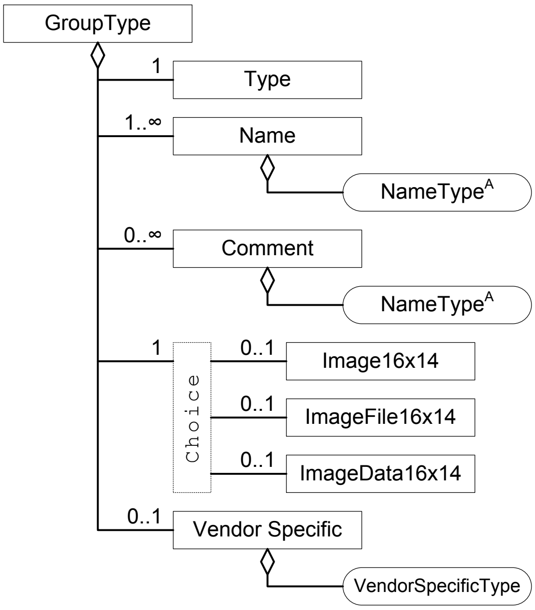

Figure 25: Content of GroupType

Table 70 describes the content of GroupType.

Table 70: Content description of GroupType

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
Type	xs:string	M	A reference handle corresponding to the *GroupType* value in *Description:Devices:Device:Group*
Name	NameType^T	M	Name for this group shown by a configuration tool
Comment	NameType^T	O	Optional comment
Image16x14	xs:string	O^D	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Obsolete configtool: skip element
ImageFile16x14	xs:string	O^D	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14*File path to a BMP file (dimension are 16x14 pixels with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
ImageData16x14	xs:hexBinary	O^D	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14*Hex binary data of a BMP file (dimensions are 16x14 pixels with 16 colors) which might be shown by a configuration tool 0xFF00FF is used for transparent color
VendorSpecific	VendorSpecificType^T	O	Vendor specific element of Vendor

13.11 InfoType

The InfoType describes hardware specifics of the EtherCAT Slave Controller, state machine and mailbox timeouts.

Figure 26 shows how the InfoType is composed.

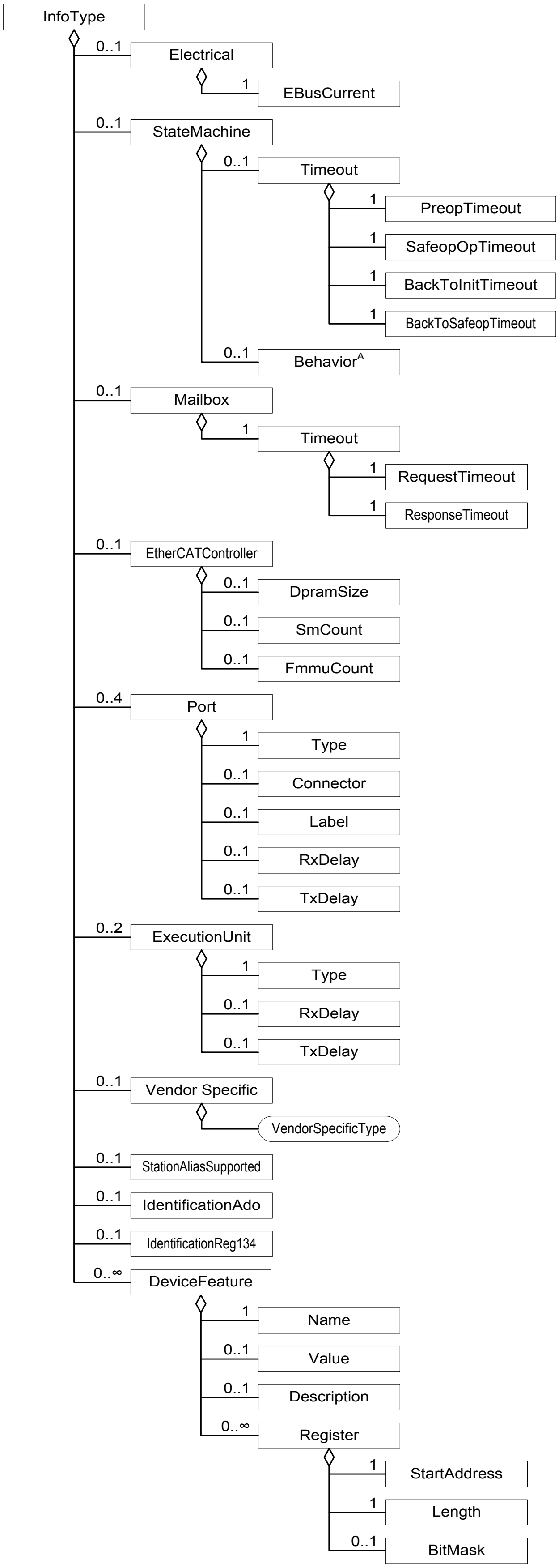

Figure 26: Content of InfoType

Table 71 describes the content of InfoType.

Table 71: Content description of InfoType

Element Name	Data Type	Use	Description
Electrical	--	O	Description of electrical properties of device
Electrical:EBusCurrent	xs:int	M	Power consumption on E-Bus in [mA]. Negative values are power suppliers.
StateMachine	--	O	Describes implemented behavior and attributes of the device’s EtherCAT state machine.
StateMachine:Timeout	--	O	Timeout values describe in milliseconds how long the master waits for the confirmation of a requested state change. The value measures from the time of sending the state change request until the state change is confirmed the latest.
StateMachine:Timeout:PreopTimeout	xs:int	M	Timeout time for state transition from INIT → PREOP/BOOT Default value: 3000ms
StateMachine:Timeout:SafeopOpTimout	xs:int	M	Timeout time for state transition from SAFEOP → OP PREOP → SAFEOP Default value: 10000ms
StateMachine:Timeout:BackToInitTimeout	xs:int	M	Timeout time for state transition from OP/SAFEOP/PREOP/BOOT → INIT SAFEOP → PREOP Default value: 5000ms
StateMachine:Timeout:BackToSafeopTimeout	xs:int	M	Timeout time for state transition from OP → SAFEOP Default value: 200ms
StateMachine:Behavior^{A}	--	O	State to which the master sets the slave at start up
Mailbox	--	O	Details for mailbox communication
Mailbox:Timeout	--	M	Timeout values in milliseconds for mailbox communication
Mailbox:Timeout: RequestTimeout	xs:int	M	Time the master re-tries to initiate a successful mailbox request (= sending the mailbox request datagram for this time until the WKC becomes valid or the timeout expires) Default value: 100ms
Mailbox:Timeout: ResponseTimeout	xs:int	M	Time the master re-tries to successfully read a mailbox response (= sending the mailbox read command for this time until the WKC becomes valid or the timeout expires) Default value: 1000ms NOTE: This time should be longer than the maximum time the slave needs to produce the mailbox response.

Element Name	Data Type	Use	Description
EtherCATController	--	O	Description of ESC features NOTE: May be used when device supports flexible usage of hardware entities
EtherCATController: DpramSize	xs:int	O	DPRAM (i.e. above address 0x1000) size of ESC in Byte
EtherCATController: SmCount	xs:int	O	Number of SyncManagers supported by ESC
EtherCATController: FmmuCount	xs:int	O	Number of FMMUs supported by ESC
Port	--	O	Description of configured available ports
Port:Type	xs:NMTOKEN	M	Allowed values: ‘MII’ ‘EBUS’ ‘NONE’
Port:Connector	xs:string	O	For future use
Port:Label	xs:string	O	Contains the text of the label printed on the corresponding port on the device, e.g. “IN” or “EtherCAT IN”
Port:RxDelay	xs:int	O	For future use
Port:TxDelay	xs:int	O	For future use
Port:PhysicalPhyAddr	xs:int	O	Logical PHY address used for accessing PHY via MII management interface (value for register 0x0512). Allowed values: -1: PHY of this port cannot be accessed via MII ≥0: hardware configured PHY address of PHY NOTE: PHY address offset of ESC is added to logical PHY address (e.g. element *Port:LogicalPhyAddr* = 10 and PHY offset of ESC = 16. Logical PHY address is 26. This value shall be equal to the hardware configured addess of the PHY). Default: *LogicalPhyAddr* = logical port number (0-3)
ExecutionUnit	--	O	Description of available execution unit
ExecutionUnit:Type	xs:NMTOKEN	M	Allowed values: 'NONE' 'PRIMARY': Execution unit after Port 0 'SECONDARY': for future use
ExecutionUnit:RxDelay	xs:int	O	For future use
ExecutionUnit:TxDelay	xs:int	O	For future use
VendorSpecific	VendorSpecific Type^T	O	Vendor specific element of InfoType
StationAliasSupported	xs:NMTOKEN	O	obsolete configtool: skip element

Element Name	Data Type	Use	Description
IdentificationAdo	HexDecValue	O	Identification information of the device. Element defines the ESC memory address where the Identification ID is saved. Length: 2 Byte EXAMPLE: If IdenficationAdo is 0x0012 the register "Configured Station Alias" is used If IdentificationAdo is 0x1000 Input Data is used
IdentificationReg134	xs:boolean	O	Device supports Explicit Device Identification according ETG.1020/ETG.1000 Note: This is the recommended way for identifying a slave with application microcontroller and ID selector.
DeviceFeature	--	O	For future use
DeviceFeature:Name	xs:string	M	For future use
DeviceFeature:Value	xs:string	O	For future use
DeviceFeature:Description	xs:string	O	For future use
DeviceFeature:Register	--	O	For future use
DeviceFeature:Register: StartAddress	xs:int	M	For future use
DeviceFeature:Register: Length	xs:int	M	Fur future use
DeviceFeature:Register: BitMask	HexDecValue	O	For future Use

Table 72 describes the attributes of StateMachine:Behavior. Lower states exclude upper states. If no attribute is available no state restrictions are set.

Table 72: Attributes of StateMachine:Behavior

Attribute	Data Type	Use	Description
StartToInit	xs:boolean	O	0: Master sets device to Op at start up 1: Master sets slave to INIT at start up
StartToPreop	xs:boolean	O	0: Master sets device to Op at start up 1: Master sets slave to PREOP at start up
StartToSafeop	xs:boolean	O	0: Master sets device to Op at start up 1: Master sets slave to SAFEOP at start up
StartToSafeopNoSync	xs:boolean	O	0: Master sets device to Op at start up 1: Master sets slave only to SAFEOP (not to OP) if the master does not support DC synchronized operation

13.12 ModuleType

Elements *Slots* and *Modules* are used to describe the following device types:

a) gateway device: device to subordinated fieldbusses

b) modular device : device has physical modules

c) module device: process data of this device is structured as modules

Figure 27 shows how the ModuleType is composed.

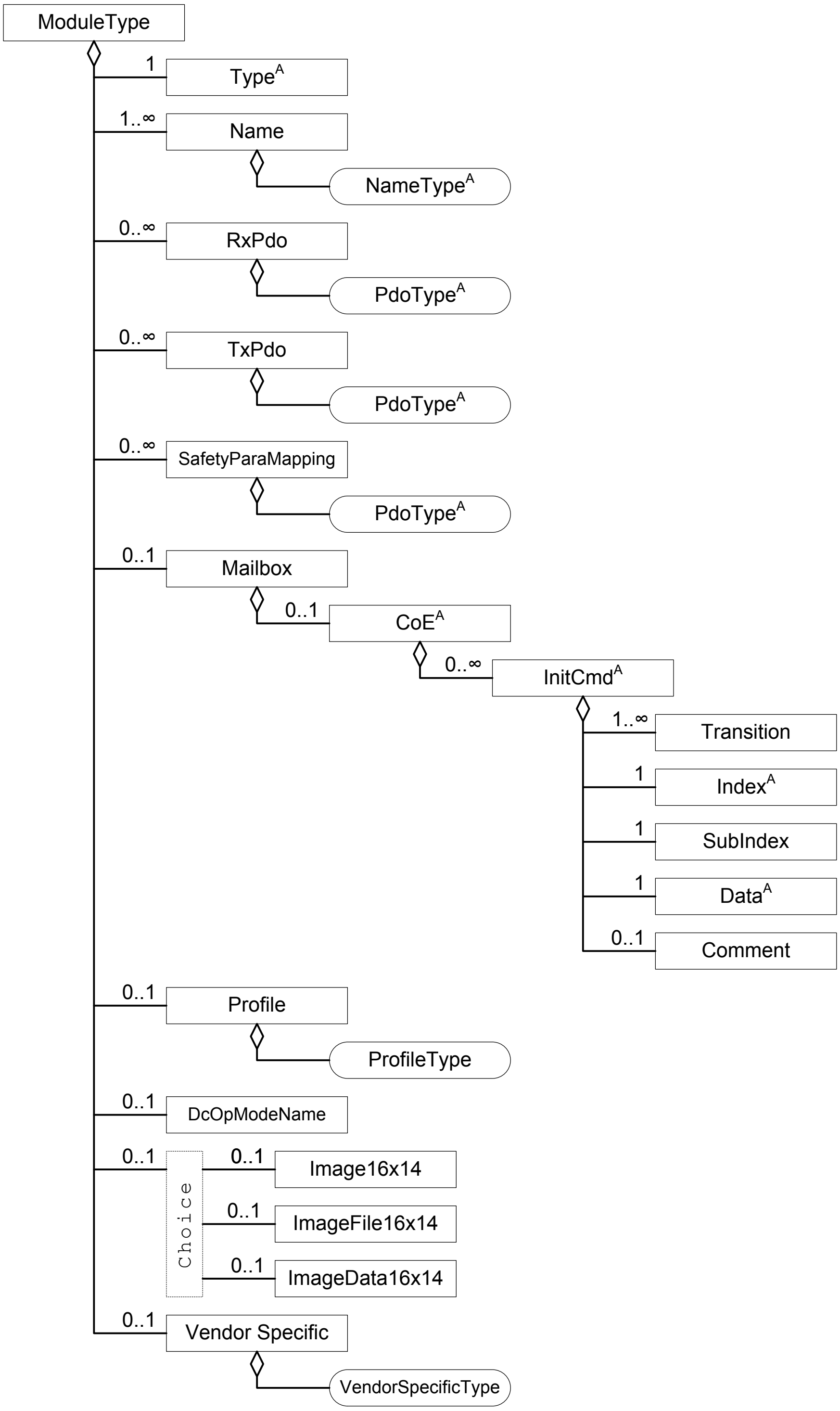

Figure 27: Content of ModuleType

Table 73 describes the content of ModuleType.

Table 73: Content description of ModuleType

Element Name	Data Type	Use	Description
Type^A	xs:string	M	Short name of this module
Name	NameType^T	M	More detailed description of this *Module*

Element Name	Data Type	Use	Description
RxPdo	PdoType[T]	O	RxPDOs of this Module. RxPDOs are added automatically to the RxPDO assignment when this Module is configured. The Index of the RxPDO depends on the position of the Module. The PDO Number of the PDO and the Index of the PDO entries might be adapted.
TxPdo	PdoType[T]	O	TxPDOs of this Module. TxPDOs are added automatically to the TxPDO assignment when this Module is configured. The Index of the TxPDO depends on the position of the Module. The PDO Number of the PDO and the Index of the PDO entries might be adapted.
SafetyParaMapping	PdoType[T]	O[D]	Mapping of the Safety Parameter Set (used for FSoE modules). Mandatory for Safety-over-EtherCAT devices. The existence of this object indicates that it is a safety-relevant module description. From PDOType - following attributes are allowed: Fixed = 1 - following elements are allowed: Index Name Entry SafetyParaMapping:Entry describes the elements that are mapped in the Safety Parameter Set. NOTE: The mapping itself is described in ETG.1020, clause Safety over EtherCAT
Mailbox	--	O	Description of mailbox features
Mailbox:CoE[A]	--	O	If this element is present the module supports CoE
Mailbox:CoE:InitCmd[A]	--	O	List of CoE initialization commands which are sent during the specified state transition
Mailbox:CoE:InitCmd:Transition	xs:NMTOKEN	M	State transitions during which the mailbox protocol specific init command is sent. For allowed state transitions refer to clause 8, excluding transitions from/to Init and Boot
Mailbox:CoE:InitCmd:Index[A]	HexDecValue	M	CoE object index
Mailbox:CoE:InitCmd:Subindex	HexDecValue	M	CoE object subindex

Element Name	Data Type	Use	Description
Mailbox:CoE:InitCmd:Data[A]	xs:hexBinary	M	CoE object data (excluding mailbox header and CoE header)
Mailbox:CoE:InitCmd:Comment	xs:string	O	Comment of this init command
Profile	ProfileType[T]	O	Profile description for this Module Describes the profile and/or subprofile of this module/ of each module channel
DcOpModeName	xs:string	O	Defines the default operation mode for this module. It is a reference to one operation mode as they are defined in the element Device:Dc:OpMode:Name
Image16x14	xs:string	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Obsolete configtool: skip element
ImageFile16x14	xs:string	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* File path to a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
ImageData16x14	xs:hexBinary	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Hex binary data of a BMP file (dimensions are 16x14 pixels with 16 colors) which might be shown by a configuration tool
VendorSpecific	VendorSpecificType[T]	O	Possible vendor specific elements can be added to the module description

Table 74 describes the attributes of ModuleType:Type.

Table 74: Attributes of ModuleType:Type

Attribute	Data Type	Use	Description
ModuleIdent	HexDecValue	M	Identifies the module. The *ModuleIdent* is vendor specific and identifies a module uniquely for one EtherCAT VendorID. NOTE: *ModuleIdent* can be downloaded in the Configured Module Ident List Object (objects 0xF03y) as expected module *ModuleIdent* can be uploaded in the Detected Module Ident List Object (objects 0xF05y) to find out the actually available modules ModuleIdent = 0x0 shall not be used for a real module. It means that no module is connected i.e. “empty slot”.

Attribute	Data Type	Use	Description
ModuleClass	xs:string	O	Name of module class (i.e. module group) to which this module is assigned to. NOTE: Vendor specific unique class name that can be used if the number of allowed modules for one slot is too long to describe them by the *ModuleIdent*.
ModulePdoGroup	xs:int	O	Specifies the PDO mapping order of the modules in the process data image. Modules of the first *ModulePdoGroup* are mapped to the beginning of the process image. Counting starts with '0'
SRA_ParamterSupported	xs:boolean	O^D	D: Mandatory for Safety over EtherCAT Devices supporting SRA Parameter 0: SRA Parameter are not supported 1: SRA Parameter supported

Table 75 describes the attributes of ModuleType:Mailbox:CoE.

Table 75: Attributes of ModuleType:Mailbox:CoE

Attributes	Data Type	Use	Description
SdoInfo	xs:boolean	O	Refer to Table 40
PdoAssign	xs:boolean	O	Refer to Table 40
PdoConfig	xs:boolean	O	Refer to Table 40
PdoUpload	xs:boolean	O	Refer to Table 40
CompleteAccess	xs:boolean	O	Refer to Table 40
EdsFile	xs:string	O	Refer to Table 40
SegmentedSdo	xs:boolean	O	Refer to Table 40
ModuleOD	xs:boolean	O	0: Module does not contain an object dictionary per module 1: Module contains an object dictionary per module

Table 76 describes the attributes of ModuleType:Mailbox:CoE:InitCmd

Table 76: Attributes of ModuleType:Mailbox:CoE:InitCmd

Attribute	Data Type	Use	Description
Fixed	xs:boolean	O	Refer to Table 42
CompleteAccess	xs:boolean	O	Refer to Table 42

Table 77 describes the attributes of ModuleType:Mailbox:CoE:InitCmd

Table 77: Attributes of ModuleType:Mailbox:CoE:InitCmd:Index

Attribute	Data Type	Use	Description
DependOnSlot	xs:boolean	O	0: The index is not adapted 1: The index is adapted depending on the slot number and *Slots:Slot@SlotIndexIncrement*
DependOnSlotGroup	xs:boolean	O	0: The index is not adapted 1: The index is adapted depending on the slot group number and *Slots:Slot@SlotGroupIndexIncrement*

Table 78 describes the attributes of ModuleType:Mailbox:CoE:InitCmd

Table 78: Attributes of ModuleType:Mailbox:CoE:InitCmd:Data

Attribute	Data Type	Use	Description
AdaptAutomatically	xs:boolean	O	0: Default values defined in *CoE:InitCmd:Data* are sent 1: Default values defined in *CoE:InitCmd:Data* are overwritten by real values, e.g. 0x1C32.02 will be adapted to the cycle time of the EtherCAT master

13.13 NameType

Figure 28 shows how the NameType type is composed.

Figure 28: Content of NameType

The attributes of NameType are described in Table 79.

Table 79: Attributes of NameType

Attribute	Data Type	Use	Description
LcId	xs:integer	O	Locale Id codes (Microsoft based) the language of the element. LcId code list (extract): 3081: English (Australia) 1033: English (United States) 2057: English (United Kingdom) 1031: German (Germany) 2055: German (Switzerland) 1036: French (France) 3084: French (Canada) 3076: Chinese (Hong Kong S.A.R.) 2052: Chinese (PRC) 4100: Chinese (Singapore) 1028: Chinese (Taiwan) 1041: Japanese 1043. Dutch (Netherlands) 1040: Italian (Italy) 1027: Catalan 1044: Norwegian (Bokmal) 2068: Norwegian (Nynorsk) 2070: Portuguese (Portugal) 1053: Swedish 1034: Spanish (Traditional Sort) 1049: Russian

13.14 ObjectInfoType

Depending on the data type described by *ObjectType* a default value and/or limits can be given using the ObjectInfoType. Figure 29 shows how the ObjectInfoType is composed.

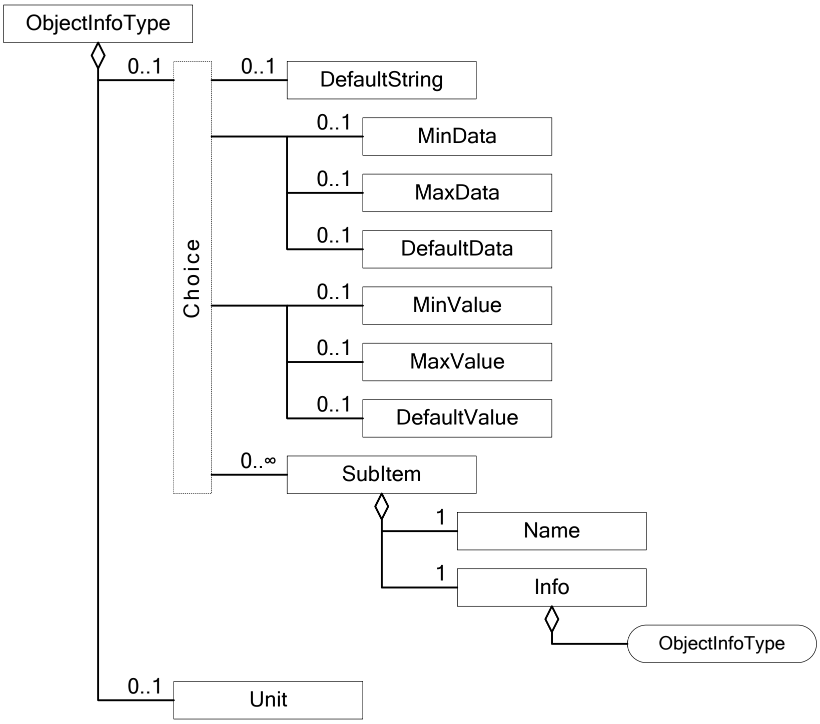

Figure 29: Content of ObjectInfoType

Table 80 describes the content of ObjectInfoType.

Table 80: Content description of ObjectInfoType

Element Name	Data Type	Use	Description
DefaultString	xs:string	O	Default string if data type STRING is used
MinData	xs:hexBinary	O	Minimum value for elements of arbitrary data type
MaxData	xs:hexBinary	O	Maximum value for elements of arbitrary data type
DefaultData	xs:hexBinary	O	Default value for elements of arbitrary data type
MinValue	HexDecValue	O	Minimum value for elements using base data types (excluding base data type with variable length)
MaxValue	HexDecValue	O	Maximum value for elements using base data types (excluding base data type with variable length)
DefaultValue	HexDecValue	O	Default value for elements using base data types (excluding base data type with variable length)
SubItem	--	O	SubItem of an object

Element Name	Data Type	Use	Description
SubItem:Name	xs:string	M	Name of this object entry. The name corresponds to *DataType:SubItem:Name.* When object entry is of data type ARRAY (i.e. no *DataType:SubItem:Name* specified): *SubItem:Name* shall be 'SubIndex xxx' xxx is decimal *SubIndex* of *SubItem*
SubItem:Info	ObjectInfoType^{T}	M	For description of an object entry Object Info according to the ObjectInfoType for each individual SubItem. *SubItem:Info:SubItem* shall not be used, i.e. no recursive usage. NOTE: Default values of *ObjectInfo:SubItem:Info* overwrites *DataType:SubItem:Info*
Unit	HexDecValue	O^{D}	D: Only used when elements *MinValue*, *MaxValue*, *DefaultValue* or *MinData*, *MaxData*, *DefaultData* are used. Unit of the object. 32-Bit Value as defined in ETG.1004 with Bit 0…7: reserved Bit 8…15: Denominator Bit 16…23: Numerator Bit 24…31: Prefix

13.15 ObjectType

Figure 30 shows how the ObjectType is composed. It is used to describe an object of the object dictionary.

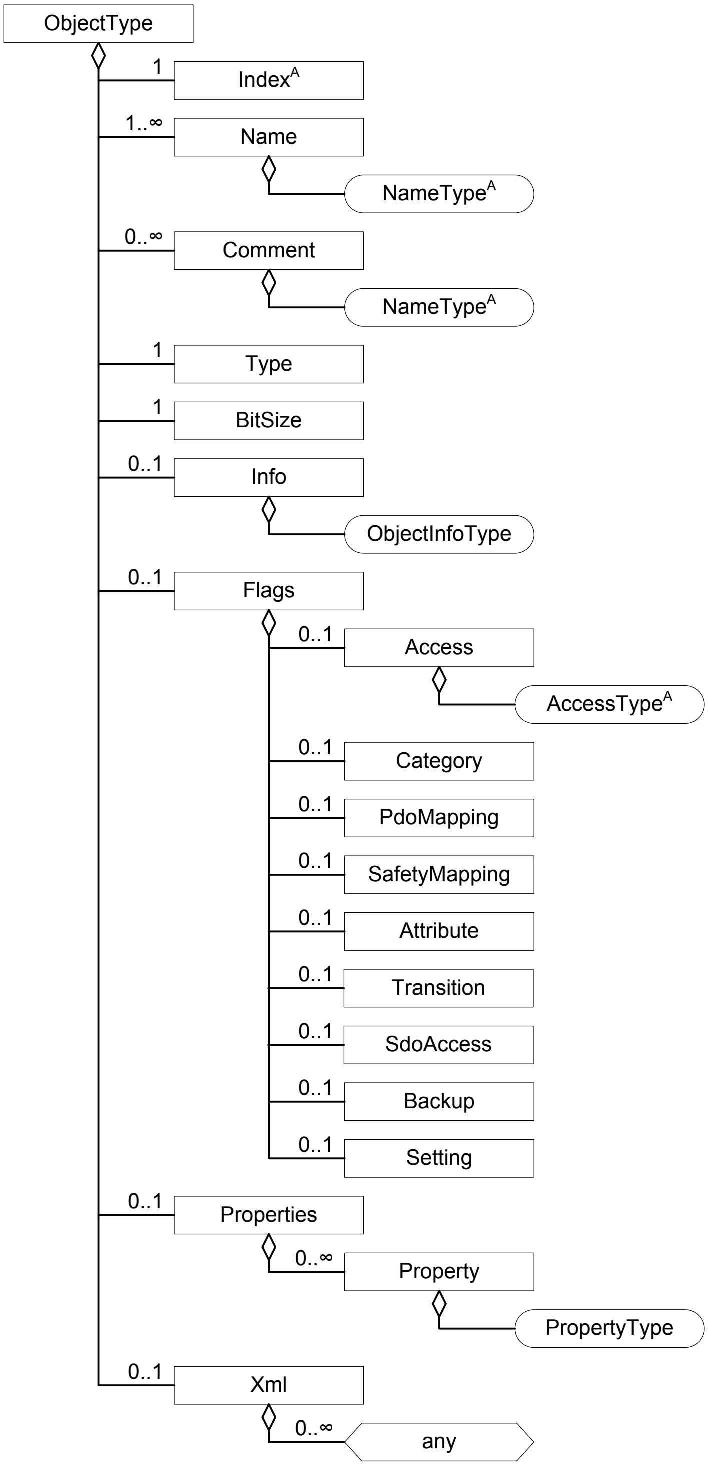

Figure 30: Content of ObjectType

Table 81 describes how the elements of ObjectType.

Table 81: Content description of ObjectType

Element Name	Data Type	Use	Description
Index^A	HexDecValue	M	Object index
Name	NameType^T	M	Name of this object
Comment	NameType^T	O	Comment on this object

Element Name	Data Type	Use	Description
Type	xs:string	M	Type of this object (BaseType or Complex type according to element *DataTypes*) NOTE: All data types used in element *Objects* shall be defined in element *DataType*
BitSize	xs:int	M	Object size in bit including alignment bits
Info	ObjectInfoType^T	O	Object info according to the ObjectInfoType (e.g. default, minimum/ maximum values and *SubItems*)
Flags	--	O	Permissions for object handling Flag information applies to the whole object. If object is of data type ARRAY or RECORD: If Flags are defined in *DataType:Subitem:Flags* those flags are used for the entries (overwrite values given here in *ObjectType:Flags*)
Flags:Access	AccessType^T	O	CoE Access Type Allowed values in high commas: 'ro': readonly (default) 'rw': readwrite 'wo': writeonly Usage according Figure 31. NOTE: The access right can be restricted by the attributes *ReadRestrictions* and *WriteRestrictions*
Flags:Category	xs:NMTOKEN	O	Use of object Allowed values: 'm': mandatory 'o': optional (default) 'c': conditional Usage according Figure 31. NOTE: Values according to CoE object specification
Flags:PdoMapping	xs:NMTOKEN	O	Object can be mapped as TxPDO, RxPDO or both Allowed values: 't', 'T': Transmit PDO (Inputs) 'r', 'R': Receive PDO (Outputs) 'tr', 'TR', 'rt', 'RT': Transmit or Receive PDO Default value: cannot be mapped Usage according Figure 31.
Flags:SafetyMapping	xs:NMTOKEN	O	Object can be mapped in the Safety Data (Input / Outputs) or the Safety Parameter Set: Allowed values: 'si', 'SI': Safe Inputs 'so', 'SO': Safe Outputs 'sio', 'SIO': Safe Input or Output 'sp', 'SP': Safety Parameter Set Default value: cannot be mapped Usage according Figure 31.
Flags:Attribute	HexDecValue	O	Attribute of a SoE IDN. 32 Bit value corresponding to attribute definition in IEC 61158-4-16
Flags:Transition	xs:NMTOKEN	O	obsolete

Element Name	Data Type	Use	Description
Flags:SdoAccess	xs:NMTOKEN	O	Allowed values: 'CompleteAccess': object can be accessed sub-index wise or by SDO complete access 'SubIndexAccess': object can only be accessed subindex wise (default value)
Flags:Backup	xs:int	O	Allowed values: 0: This object is no Backup entry 1: This object is a Backup entry Backup entries are used for Device Replacement NOTE: If true default value for this object should be provided in *Object:Info*
Flags:Setting	xs:int	O	Allowed values: 0: This object is no Setting entry 1: This object is a Setting entry Setting entries are downloaded during start-up by the master NOTE: If true default value for this object should be provided in *Object:Info*
Properties	--	O	General description of additional properties. NOTE: Can be used for example to define function groups
Properties:Property	PropertyType^T	O	Property information (refer to clause 13.18)
Xml	--	O	General XML description for further information possible (similar to properties description above, but more generous)

Figure 31 shows which value is adopted for a flags existing in both elements, *DataType:SubItem:Flags* and *Object:Flag*

NOTE: "FLAG" is a placeholder for Flags: *Access, Category, PdoMapping, SafetyMapping, Backup, and Setting*

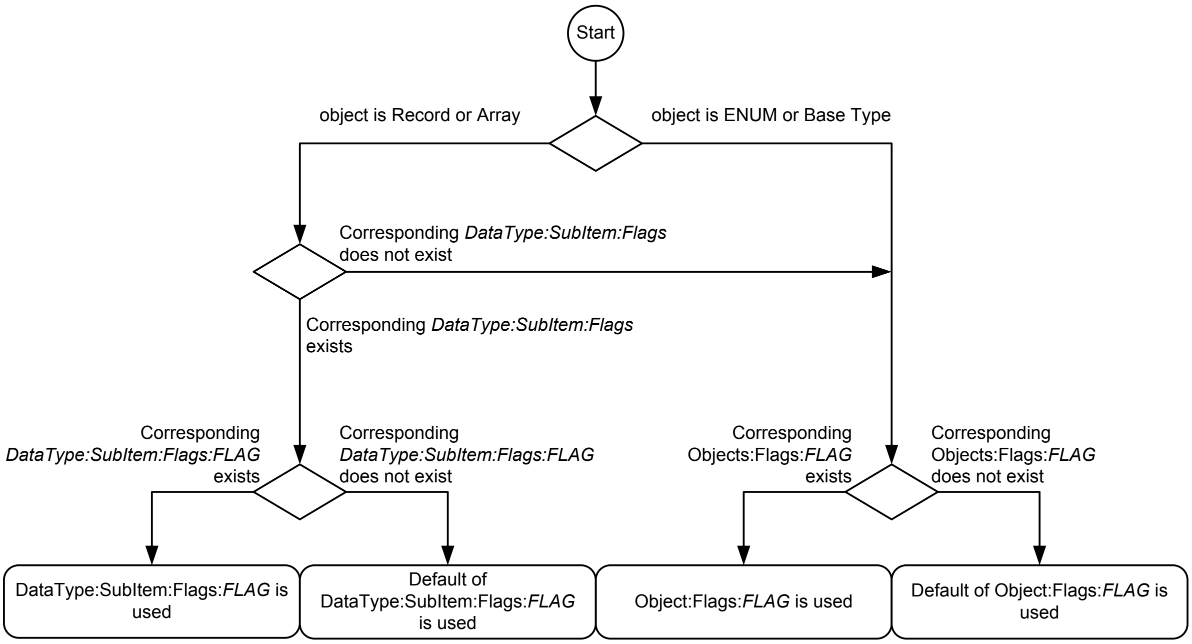
Start
object is Record or Array
object is ENUM or Base Type
Corresponding *DataType:SubItem:Flags* does not exist
Corresponding *DataType:SubItem:Flags* exists
Corresponding *DataType:SubItem:Flags:FLAG* exists
Corresponding *DataType:SubItem:Flags:FLAG* does not exist
Corresponding Objects:Flags:*FLAG* exists
Corresponding Objects:Flags:*FLAG* does not exist
DataType:SubItem:Flags:*FLAG* is used
Default of DataType:SubItem:Flags:*FLAG* is used
Object:Flags:*FLAG* is used
Default of Object:Flags:*FLAG* is used

Figure 31: Rules for Flags of DataType and Object element

Table 82 describes the attributes of ObjectType:Index

Table 82: Attributes of ObjectType:Index

Attribute	Data Type	Use	Description
DependOnSlot	xs:boolean	O	0: The index is not adapted depending on the slot number and the *SlotIndexIncrement* value in the element *Slots* 1: The index is adapted depending on the slot number and the *SlotIndexIncrement* value in the element *Slots*
DependOnSlotGroup	xs:boolean	O	0: The index is not adapted depending on the *SlotGroup* and the *SlotGroupIndexIncrement* value in the element *Slots* 1: The index is adapted depending on the *SlotGroup* and the *SlotGroupIndexIncrement* value in the element *Slots*
OverwrittenByModule	xs:boolean	O	0: Object is part of object dictionary 1: Object is ignored (e.g. when configuration tool supports the elements *Modules* and *Slots* the objects of the objects dictionary are defined in the element *Modules:Profile*)

13.16 PdoType

PdoType is used to describe the RxPDOs and TxPDOs. Figure 32 shows how the PdoType is composed.

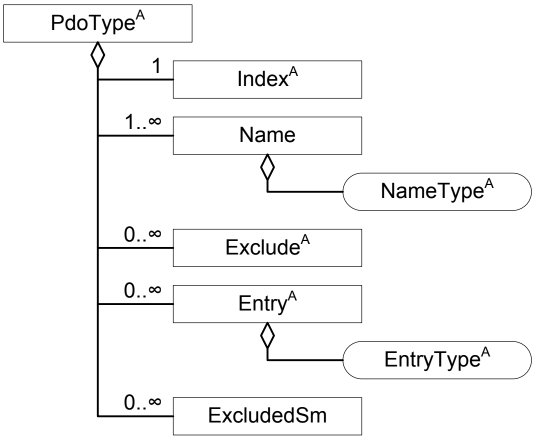

Figure 32: Content of PdoType

Table 83 describes the attributes of PdoType.

Table 83: Attributes of PdoType

Attribute	Data Type	Use	Description
Fixed	xs:boolean	O	0: PDO mapping can be changed, i.e. PDO entries can be deleted or added 1: PDO content is not configurable i.e. complete PDO content is fixed; attribute *Fixed* of the PDO entries (*EntryType*) is overwritten
Mandatory	xs:boolean	O	PDO is configured (= assigned to) in a SyncManager 0: PDO assignment is changeable (e.g. content of object 0x1C12 can be changed) 1: PDO must be assigned to the default SyncManager NOTE: If there are no default PDOs assigned to a SyncManager (refer to attribute *Sm*) the *Mandatory* bit shall not be set.

Attribute	Data Type	Use	Description
Virtual	xs:boolean	O	0: Standard PDO description 1: Two cases: Process data value is read from a register value (ESC memory address 0x0000:0x0EFF). In this case element *Sm@Virtual* shall be "TRUE". Content (=PDO entries) of PDO can be configured manually via the configuration tool. In this case no element *PdoType:Entry* is described. NOTE: Configuration tools shall sort process data byte aligned by default by adding padding PDO entries with *Index*=0 and *BitLen* = [bit size of padding bits].
Sm	xs:int	O	Default SyncManager for this PDO i.e. this PDO is included in the process data image by default NOTE: PDOs which are not assigned to any SyncManager by default may be assigned manually via the configuration tool. In this case the PDO must not be configured as mandatory (*Pdo@Mandatory* = FALSE). TxPDOs shall only be assigned to Input SyncManager. RxPDOs shall only be assigned to Output SyncManager (refer also to attribute *Device@Sm*, Table 24).
Su	xs:int	O	Groups PDOs and defines to which FMMU (and with this SyncManager) they can be assigned to, i.e. *Pdo@Su* and *Fmmu@Su* have the same value.
PdoOrder	xs:int	O	Obsolete configtool: skip element
OSFac	xs:int	O	Default oversampling factor
OSMin	xs:int	O	Minimum oversampling factor
OSMax	xs:int	O	Maximum oversampling factor
OSIndexInc	xs:int	O	Oversampling increment for entry indexes
OverwrittenByModule	xs:boolean	O	0: Object is part of process data configuration 1: Object is ignored (e.g. when configuration tool supports the elements *Modules* and Slots the process data are defined in the element *Modules*)
SRA_Parameter	xs:boolean	O	Attribute shall only be used in safety parameter mapping definition (objects 0x1D00-0x1DFF; ESI element *Modules:SafetyParameterMapping*). Allowed Values: 0: this mapping parameter is written by a safety master in the FSoE connection and shall not be used for the SRA CRC calculation 1: this mapping parameter is part of the SRA parameter set and the values shall be used for calculation of the SRA CRC NOTE: see ETG.5120 for SRA parameter definition

Table 84 describes the content of PdoType.

Table 84: Content description of PdoType

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
Index^{A}	HexDecValue	M	PDO index NOTE: RxPDOs: Index area 0x1600 to 0x17FF TxPDOs: Index area 0x1A00 to 0x1BFF
Name	NameType^{T}	M	PDO name
Exclude^{A}	HexDecValue	O	List of PDO indices that are excluded if this PDO is assigned to a SyncManager PDOs are “mutually exclusive” i.e. if a PDO (PDO_A) is excluded by another PDO (PDO_B) than PDO_B also has to be excluded by PDO_A.
Entry^{A}	EntryType^{T}	O	Description of all entries according to *EntryType*
ExcludedSm	xs:int	O	SyncManager to which this PDO may not be assigned to. default = PDO can be assigned to all SyncManager with matching type/direction NOTE: When PDO can be assigned to any SM with matching direction and type (1/3-buffer mode) this element is not needed.

Table 85 describes the attributes of PdoType:Index.

Table 85: Attributes of PdoType:Index

Attribute	Data Type	Use	Description
DependOnSlot	xs:boolean	O	0: Object Index is fixed 1: Object Index depends on the Slot number to which the module is assigned to
DependOnSlotGroup	xs:boolean	O	0: Object Index is fixed 1: Object Index depends on the *SlotGroup* number to which the module is assigned to

Table 86 describes the attributes of PdoType:Exclude.

Table 86: Attributes of PdoType:Exclude

Attribute	Data Type	Use	Description
DependOnSlot	xs:boolean	O	0: Object Index is fixed 1: Object Index depends on the Slot number to which the module is assigned to
DependOnSlotGroup	xs:boolean	O	0: Object Index is fixed 1: Object Index depends on the *SlotGroup* number to which the module is assigned to

Table 87 describes the attributes of PdoType:Entry.

Table 87: Attributes of PdoType:Entry

Attribute	Data Type	Use	Description
Fixed	xs:boolean	O	0: Entries of this PDO can be edited, added, or deleted (overwritten by attribute *PdoType@Fixed*; refer also to Table 83) 1: Entries of this PDO are fixed

13.17 ProfileType

This element describes an offline object dictionary and, if supported, the profile type. It shall be consistent with the description in element *RxPdo* and *TxPdo*, and the online object dictionary of the slave.

The Object Dictionary can either be described within the ProfileType or a reference can be given to an external Object Dictionary File (refer to clause 15).

Figure 33 shows how the *ProfileType* is composed.

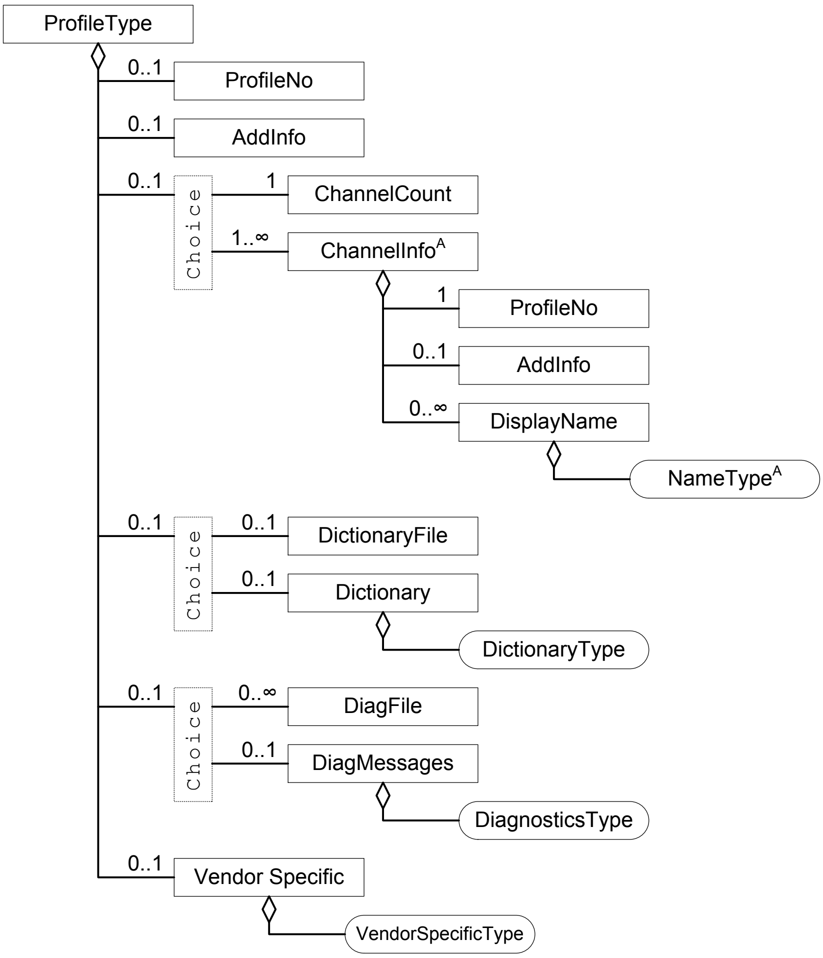

Figure 33: Content of ProfileType

Table 88 describes the content of ProfileType.

Table 88: Content description of ProfileType

Element Name	Data Type	Use	Description
ProfileNo	xs:int	O^D	D: Use according Table 89 Number of device profile (low word of CoE object 0x1000) used by this device (e.g. 5001 for MDP or 402 for CiA402). NOTE: Only profiles specified by ETG may be used.
AddInfo	xs:int	O^D	D: Use according Table 89 Sub-Profile number (high word of CoE object 0x1000; e.g. according to ETG.5001 MDP sub-profile types. If *ProfileNo* = 402, Default Value = 2 (Servo drives).

Element Name	Data Type	Use	Description
ChannelCount	xs:int	M^D	D: Use according Table 89 Number of channels of this device (e.g. an analog device with 4 analog inputs can be described as device with 4 channels)
ChannelInfo^A		M^D	D: Use according Table 89 Describes each channel of a device with its profile and sub-profile number. Used when a device supports several channels and those channels support different profiles.
ChannelInfo:ProfileNo	xs:int	M	Profile number (low word of CoE object 0x1000) of this channel
ChannelInfo:AddInfo	xs:int	O	Sub-profile number (high word of CoE object 0x1000) of this channel If *ChannelInfo:ProfileNo* = 402, Default Value = 2 (Servo drives).
ChannelInfo:DisplayName	NameType^T	O	If a drive supports more than one channel then a name can be assigned to each of them. A configuration tool may show the single channels with the *ChannelInfo:DisplayName* to link them to a drive manager tool NOTE: May be used to assign several channels to one axis; in this case *DisplayName* describes the channel name.
DictionaryFile	xs:string	O	Path to dictionary file according to EtherCATDict.xsd
Dictionary	DictionaryType^T	O	Object Dictionary of this device NOTE: Mandatory objects shall be defined, optional objects may be defined.
DiagFile	xs:string	O	Path to DiagHistory file according to EtherCATDiag.xsd
DiagMessages	DiagnosticsType^T	O	Definition of diagnosis messages
VendorSpecific	VendorSpecificType^T	O	Vendor specific element of *ProfileType*

Table 89 describes the elements ProfileNo, AddInfo, ChannelCount, and ChannelInfo.

Table 89: Description of Profiles

Profile usage	used elements
Slave does not distinct several channels	*ProfileNo*, *AddInfo*
Slave supports several channels with identical profile and sub-profile number	*ProfileNo*, *AddInfo*, *ChannelCount*
Slave supports several channels with different profile and/or sub-profile number	*ChannelInfo* NOTE: one *ChannelInfo* element per channel

Table 90 describes the attribute of ProfileType:ChannelInfo

Table 90: Attributes of ProfileType:ChannelInfo

Attribute	Data Type	Use	Description
OverwrittenByModule	xs:boolean	O	0: Profile description is fixed 1: Profile number and sub-profile number is not considered (i.e. when configuration tool supports the elements *Modules* and *Slots* this object is defined in the element *Modules* of the ESI)

13.18 PropertyType

Figure 34 shows how the PropertyType is composed.

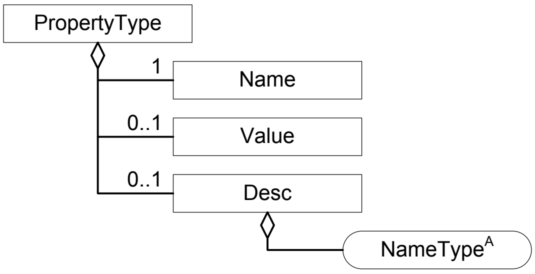

Figure 34: Content of PropertyType

Table 91 describes the content of PropertyType.

Table 91: Content description of PropertyType

Element Name	Data Type	Use	Description
Name	xs:string	M	Name of the property
Value	xs:string	O	Value of the property
Desc	NameType^T	O	Description of the property

13.19 PhysicsType

Figure 35 shows how the *PhysicsType* is composed.

Figure 35: Content of PhysicsType

Table 92 describes the content of PhysicsType.

Table 92: Content description of PhysicsType

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
PhysicsType	xs:string	-	Physic type of each port (MII or E-Bus). 1st character: Physics of logical port 0 (=A) 2nd character: Physics of logical port 1 (=B) 3rd character: Physics of logical port 2 (=C) 4th character: Physics of logical port 3 (=D) Allowed values: 'Y': MII 'H': MII - Fast Hot Connect 'K': E-Bus (LVDS) ' ': Port not used (blank character) Following blanks may be omitted, i.e. 'YY ' is equal to 'YY'.

13.20 SlotType

Elements *Slots* and *Modules* are used to describe the following device types:

a) gateway device: device to subordinated fieldbuses

b) modular devices : device has physical modules

c) module device: process data of this device is structured as modules

Figure 36 shows how the SlotType is composed.

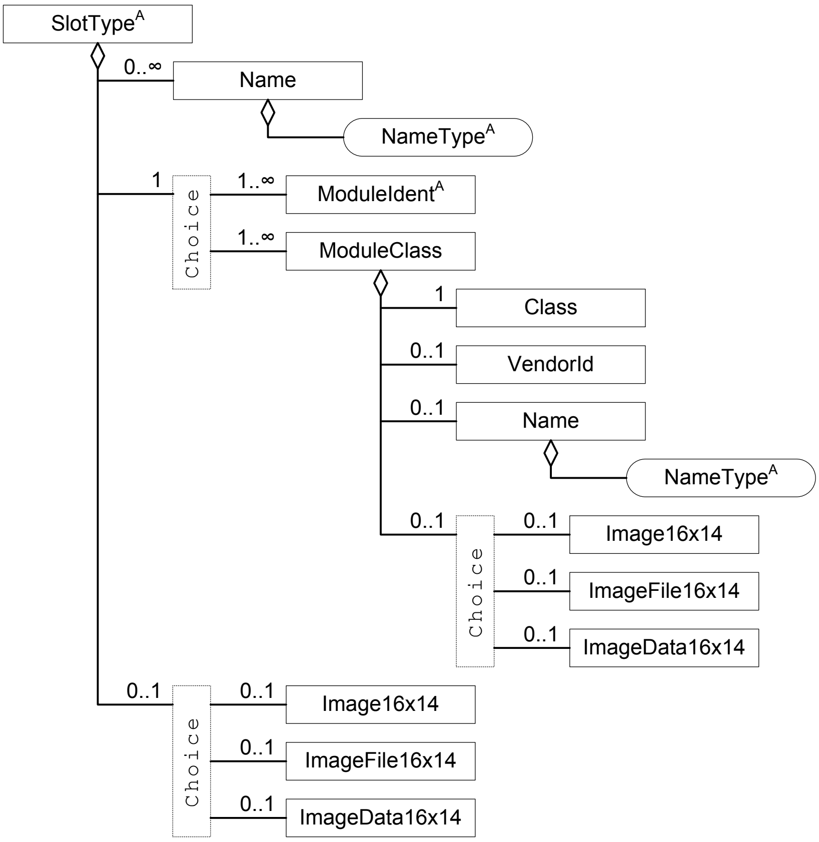

Figure 36: Content of SlotType

Table 93 describes the content of SlotType.

Table 93: Content description of SlotType

Element Name	Data Type	Use	Description
Name	NameType[T]	O	Name of slot(s)
ModuleIdent[A]	HexDecValue	M[D]	D: either *ModuleIdent* or *ModuleClass* List of modules (identified by *ModuleIdent*) which can be assigned to this slot
ModuleClass	--	M[D]	D: either *ModuleIdent* or *ModuleClass* List of *ModulClasses*: Modules of the listed *ModuleClass* can be assigned to this slot.
ModuleClass:Class	xs:string	M	Identifier of one *ModuleClass* that can be assigned to this slot
ModuleClass:VendorId	HexDecValue	O	Modules of this vendor may be assigned to this slot. Default value: value from element *Vendor:Id*
ModuleClass:Name	NameType[T]	O	Name of the ModuleClass may be assigned to this slot.
ModuleClass:Image16x14	xs:string	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Obsolete Configtool: skip element
ModuleClass:ImageFile16x14	xs:string	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* File path to a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
ModuleClass:ImageData16x14	xs:hexBinary	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Hex binary data of a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
Image16x14	xs:string	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Obsolete Configtool: skip element
ImageFile16x14	xs:string	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* File path to a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
ImageData16x14	xs:hexBinary	O[D]	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Hex binary data of a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color

Table 94 describes the attributes of SlotType.

Table 94: Attributes of SlotType

Attribute	Data Type	Use	Description
SlotGroup	HexDecValue	O	Number of *SlotGroup* this slot belongs to. NOTE: Use for complex Slot/Module description when several *SlotGroups* are required.
MinInstances	HexDecValue	M	Minimum number of how many instances of this slot type shall be configured
MaxInstances	HexDecValue	M	Maximum number of how many instances of this slot type can be configured
SlotPdoIncrement	HexDecValue	O^D	D: Mandatory if *SlotGroupPdoIncrement* is not used Index distance between two consecutive PDO mapping objects. Overwrites value of *Slots@SlotPdoIncrement* NOTE: *SlotIndexIcrement* may have to be referenced by attribute *DependOnSlot* of the following elements: *Rx/TxPdo:Index* *Rx/TxPdo:Exclude:Index*
SlotGroupPdoIncrement	HexDecValue	O^D	D: Mandatory if *SlotPdoIncrement* is not used Index distance between the first PDO mapping object of two consecutive Slot Groups Overwrites value of *Slots@SlotGroupPdoIncrement* NOTE: *SlotGroupIndexIcrement* may have to be referenced by attribute *DependOnSlotGroup* of the following elements: *Rx/TxPdo:Index* *Rx/TxPdo:Exclude:Index*
SlotIndexIncrement	HexDecValue	O^D	D: Mandatory if *SlotGroupIndexIncrement* is not used Index distance between object indexes (especially *Rx/TxPdo* entries objects) of two consecutive slots Overwrites value of *Slots@SlotIndexIncrement* NOTE: *SlotIndexIcrement* may have to be referenced by attribute *DependOnSlot* of the following elements: *Profile:Dictionary:Objects:Object:Index* *Mailbox:Coe:InitCmd:Index* *SafetyParaMapping:Index* *Rx/TxPdo:Entry:Index*
SlotGroupIndexIncrement	HexDecValue	O^D	D: Mandatory if *SlotIndexIncrement* is not used Index distance between the first object of two consecutive *SlotGroups* Overwrites value of *Slots@SlotGroupIndexIncrement* NOTE: *SlotGroupIndexIcrement* may have to be referenced by attribute *DependOnSlotGroup* of the following elements: *Profile:Dictionary:Objects:Object:Index* *Mailbox:Coe:InitCmd:Index* *SafetyParaMapping:Index* *Rx/TxPdo:Entry:Index*

Attribute	Data Type	Use	Description
TreeView	xs:NMTOKEN	O	Depending on the chosen NMTOKEN value the configuration tool shows variables of the device grouped by *SlotGroup*, *Slot*, or *Pdo*. Allowed Values: ‘SLOTGROUP’ ‘SLOT’ ‘PDO’

Table 95 describes the attributes of ModuleIdent.

Table 95: Attributes of ModuleIdent

Attribute	Data Type	Use	Description
Default	HexDecValue	O	A configuration tool may use this information to add modules (represented by ModuleIdent) to the slot by default. Allowed value: 0: this module is not used by default for the slot 1: this module is used by default for the slot NOTE: for each slot only one instance of *ModuleIdent@Default* = 1 allowed.

13.21 SubItemType

SubItemType describes an index of a RECORD or ARRAY data type. Figure 37 shows how the SubItemType is composed.

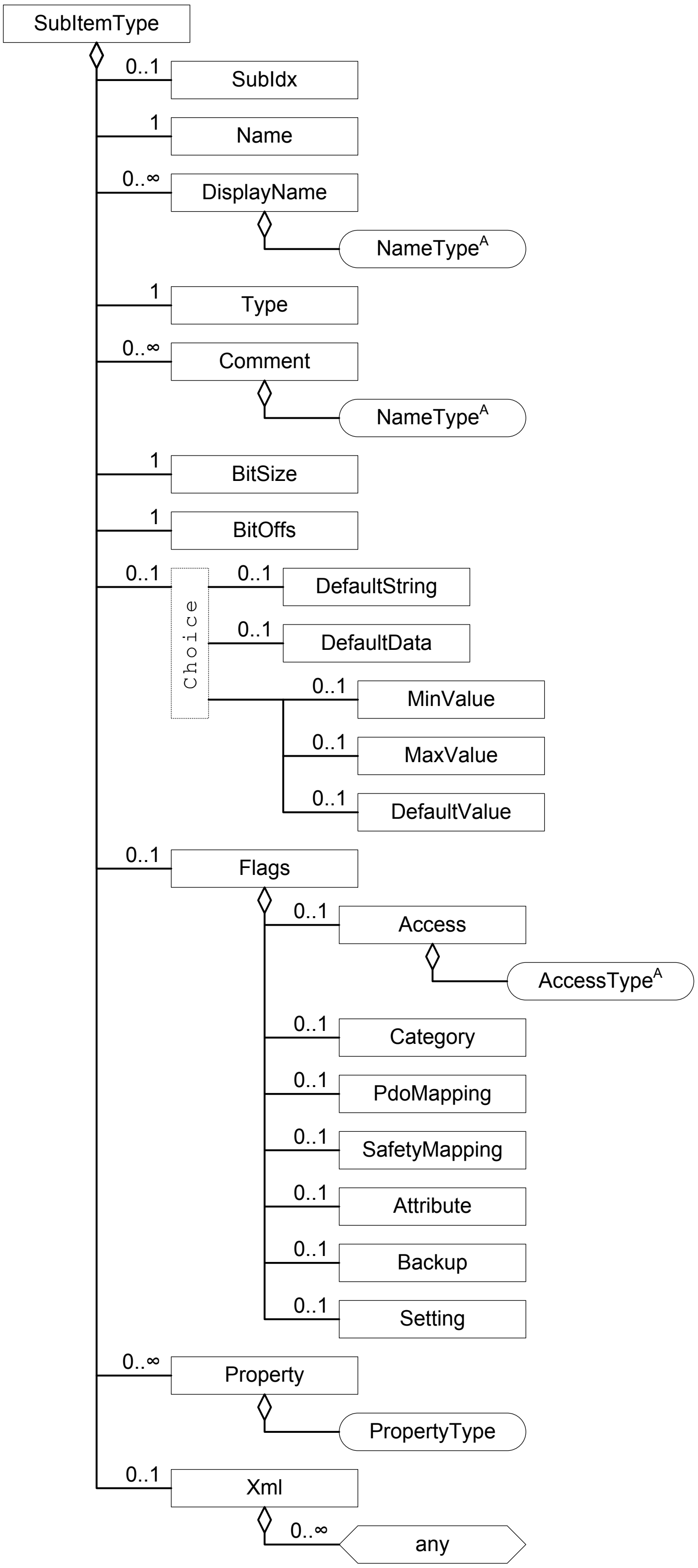

Figure 37: Content of SubItemType

Table 96 describes the content of SubItemType.

Table 96: Content of SubItemType

Element Name	Data Type	Use	Description

Element Name	Data Type	Use	Description
SubIdx	HexDecValue	O^{D}	D: Mandatory for every SubItem when object data type is RECORD Mandatory exclusively for *SubItem* 0 (= subindex 0) when object data type is ARRAY. Usage according to Table 23.
Name	xs:string	M	Name of *SubItem* Name of first *SubItem* (=subindex0) shall be “SubIndex 000”, “number of entries”, or “Number of entries”. If data type ARRAY: Not relevant for second *SubItem*. If data Type RECORD: This name is the identifier to the corresponding entry in the Object definition (*Object:Name*), i.e. spelling must be identical.
DisplayName	NameType^{T}	O	Contains an alternative name for this subindex that might be shown additionally to or instead of value of *Name* by a configuration tool. NOTE: element *Name* holds the name that is used for SDO info and might be defined by specifications (e.g. “Hardware Version”). Element *DisplayName* can be used for an individual term also in different languages.
Type	xs:string	M	Data type of this *SubItem* Usage according to Table 23.
Comment	NameType^{T}	O	Description of the *SubIndex* content
BitSize	xs:int	M	Length of *SubIndex* value in bit
BitOffs	xs:int	M	Bit address of the *SubItem* value starting at 0. NOTE: The bit offset of *SubItem* 0 shall be 0x00. For *SubItems* >0 the bit offset can be chosen without any restriction. Padding bits for alignment do not have to be described explicitly.
DafaultString	xs:string	O	Obsolete Configtool: skip element
DefaultData	xs:hexBinary	O	Obsolete Configtool: skip element
MinValue	HexDecValue	O	Obsolete Configtool: skip element
MaxValue	HexDecValue	O	Obsolete Configtool: skip element
DefaultValue	HexDecValue	O	Obsolete Configtool: skip element

Element Name	Data Type	Use	Description
Flags	--	O	Permissions for object handling For ARRAY and RECORD If Subitems are defined the flags in *SubitemType:Flags* shall be used for the entries (overwrite values given in *ObjectType:Flags*)
Flags:Access	AccessType^T	O	CoE Access Type Allowed values in high commas: ‘ro’: readonly (default) ‘rw’: readwrite 'wo': writeonly Usage according Figure 31. NOTE: The access right can be restricted by the attributes *ReadRestrictions* and *WriteRestrictions*
Flags:Category	xs:NMTOKEN	O	Use of Object Allowed values in high commas: ‘m’: mandatory ‘o’: optional ‘c’: conditional Usage according Figure 31.
Flags:PdoMapping	xs:NMTOKEN	O	SubItem can be mapped as, TxPDO, RxPDO or both Allowed values in high commas; no case sensitivity: 't', ‘T’: Transmit PDO (Inputs) 'r', ‘R’: Receive PDO (Outputs) 'tr', ‘TR’, 'rt', ‘RT’: Transmit or Receive PDO Default value: cannot be mapped (element not available) Usage according Figure 31.
Flags:SafetyMapping	xs:NMTOKEN	O	SubItem can be mapped in the Safety Data (Input / Outputs) or the Safety Parameter Set: Allowed values in high commas; no case sensitivity:: ‘si’, ‘SI’: Safe Inputs ‘so’, ‘SO’: Safe Outputs ‘sio’, ‘SIO’: Safe Input or Output ‘sp’, ‘SP’: Safety Parameter Set Default value: cannot be mapped (element not available) Usage according Figure 31.
Flags:Attribute	HexDecValue	O	For future use
Flags:Backup	xs:int	O	Allowed values: 0: This SubItem is no Backup entry 1: This SubItem is a Backup entry Backup entries are used for Device Replacement (refer to ETG.1020) NOTE: If true default value for this SubItem should be provided in Object:SubItem:Info

Element Name	Data Type	Use	Description
Flags:Setting	xs:int	O	Allowed values: 0: This SubItem is no Setting entry 1: This SubItem is a Setting entry Setting entries are downloaded during start-up by the master (refer to ETG.1020) NOTE: If true default value for this SubItem should be provided in Object:SubItem:Info
Property	PropertyType^T	O	General description of additional properties. Can be used for example to define function groups (refer to clause 13.18).
Xml	--	O	Obsolete Configtool: skip element

13.22 UnitTypeType

UnitTypeType describes the data type for unit descriptions according ETG.1020.

Figure 38 shows how the *UnitTypeType* is composed.

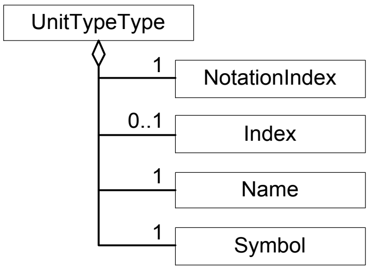

Figure 38: Content of VendorSpecificType

Table 97 describes the content of UnitTypeType.

Table 97: Content of UnitTypeType

Element Name	Data Type	Use	Description
NotationIndex	HexDecValue	M	Notation index (UINT8) of unit type according to ETG.1004 NOTE: if supported, CoE object 0x400 + "notation index" shall hold the same information as this element
Index	HexDecValue	O	Object Index of the unit definition as defined in ETG.1004 Unit Specification. Note: *Index* = *NotationIndex* + 0x400
Name	xs:string	M	Name of unit according to ETG.1004
Symbol	xs:string	M	Symbol of unit according to ETG.1004

13.23 VendorSpecificType

VendorSpecificType describes a type created for vendor specific purposes only.

CAUTION: Data in the *VendorSpecific* element cannot be evaluated by third party configuration tools and should not be used for function-critical information.

Figure 39 shows how the VendorSpecificType is composed.

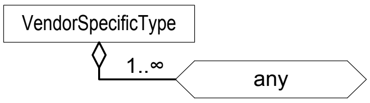

Figure 39: Content of VendorSpecificType

13.24 VendorType

Figure 40 shows how the VendorType is composed.

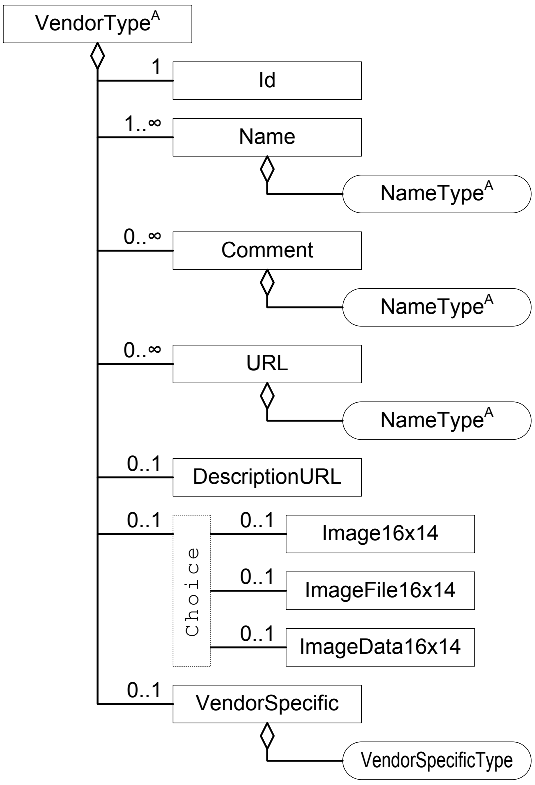

Figure 40: Content of VendorType

Table 98 describes the attributes of VendorType.

Table 98: Attributes of VendorType

Attribute	Data Type	Use	Description
UniqueName	xs:string	O	Obsolete Configtool: skip element

Table 99 describes the content of VendorType.

Table 99: Content description of VendorType

Element Name	Data Type	Use	Description
Id	HexDecValue	M	EtherCAT vendor ID (OD 0x1018.01).
Name	NameType^T	M	Expedient vendor name
Comment	NameType^T	O	Comment
URL	NameType^T	O	Company URL
DescriptionURL	xs:string	O	URL to all ESI files

Element Name	Data Type	Use	Description
Image16x14	xs:string	O^D	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Obsolete Configtool: skip element
ImageData16x14	xs:hexBinary	O^D	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* Hex binary data of a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool 0xFF00FF is used for transparent color
ImageFile16x14	xs:string	O^D	D: either *Image16x14* (obsolete) or *ImageFile16x14* or *ImageData16x14* File path to a BMP file (dimension should be 16x14 with 16 colors) which might be shown by a configuration tool. 0xFF00FF is used for transparent color
VendorSpecific	VendorSpecificType^T	O	Vendor specific element of Vendor

14 EtherCATBase

The class EtherCATBase defines the types listed in Table 100, which are described in detail in clause 13.

Those data types are listed in a separate file (EtherCATBase.xsd) so that they can be used by other files (e.g. EtherCATDict.xsd).

Table 100: EtherCATBase Types

Type
AccessType
ArrayInfoType
DataTypeType
DiagnosticsType
EntryType
EnumInfoType
HexDecValue*
ModuleType
NameType
ObjectInfoType
ObjectType
PdoType
ProfileType
PropertyType
SubItemType
UnitTypeType
VendorSpecificType
VendorType

* HexDecValue (simpleType) is described in clause 6.

15 EtherCATDict

The offline object dictionary can be integrated in the EtherCAT Slave Information file according to element EtherCATInfo (file based on EtherCATInfo.xsd), element *Profile*. Another possibility is to define the objects in an extra file defined by EtherCATDict.xsd. In this case the EtherCATInfo gives a reference to this file (*Description:Devices:Device:Profile:DictionaryFile*) .

Figure 41 shows how the EtherCATDict class is composed.

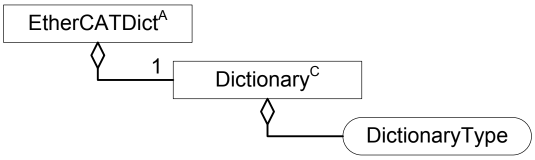

Figure 41: Content of EtherCATDict

Table 101 describes the attributes of EtherCATDict.

Table 101: Attributes of EtherCATDict

Attribute	Data Type	Use	Description
Version	xs:string	O	Version of this dictionary file

Table 102 describes the content of EtherCATDict.

Table 102: Content of EtherCATDict

Data Type	Data Type	Use	Description
Dictionary	DictionaryType^T	M	Information about the vendors’ dictionary

16 EtherCATDiag

The diagnosis information can be integrated in the EtherCAT Slave Information file according to element *ProfileType : DiagMessages*. Another possibility is to describe the diagnosis information in an extra file defined by schema EtherCATDiag.xsd. In this case the EtherCATInfo gives a reference to this file (*Description:Devices:Device:Profile:DiagFile*).

Figure 41 shows how the EtherCATDiag class is composed.

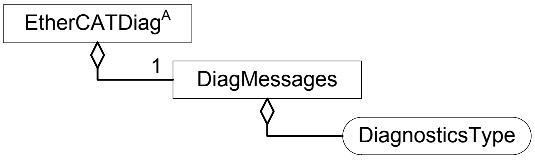

Figure 42: Content of EtherCATDiag

Table 103 describes the attributes of EtherCATDiag.

Table 103: Attributes of EtherCATDict

Attribute	Data Type	Use	Description
Version	xs:string	O	Version of this diagnosis file

Table 104 describes the content of EtherCATDiag.

Table 104: Content of EtherCATDiag

Data Type	Data Type	Use	Description
DiagMessages	DictionaryType^T	M	Diagnosis messages from one source

17 EtherCATModule

The *Modules* description can be integrated into the ESI according to EtherCATInfo (EtherCATInfo.xsd). Another possibility is given with the use of generating an additional file based on EtherCATModule.xsd so that other files can use the description, too. In this case the EtherCATInfo gives a reference to this file via the element *EtherCATInfo:InfoReference*.

Figure 43 shows how the EtherCATModule class is composed.

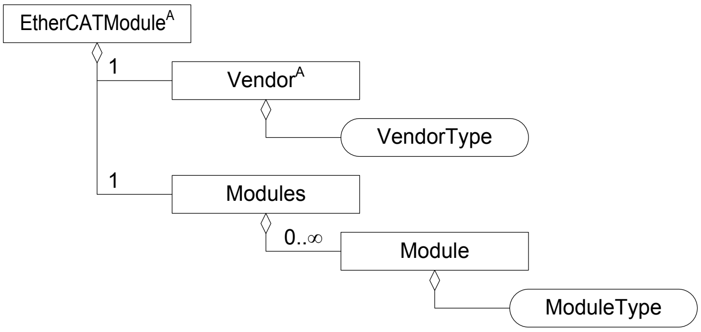

Figure 43: Content of EtherCATModule

Table 105 describes the attributes of EtherCATModule.

Table 105: Attributes of EtherCATModule

Attribute	Data Type	Use	Description
Version	xs:string	O	Version of the schema in which the Modules element is defined

Table 106 describes the content of EtherCATModule.

Table 106: Content of EtherCATModule

Attribute	Data Type	Use	Description
Vendor	VendorType^T	M	Vendor Information
Modules	-	M	Modules main element
Modules:Module	ModuleType^T	O	Module main element

Table 107 describes the attributes of EtherCATModule:Vendor

Table 107: Attributes of EtherCATModule:Vendor

Attribute	Data Type	Use	Description
FileVersion	xs:string	O	Version of the file. NOTE: Do not confuse with Schema Version