

IODD

IO Device Description

Specification

related to
IO-Link Communication Specification V1.1.3
and
IODD Schemas V1.1

Draft Version 1.1.3
July 2020

Order No: 10.012

File name: IO-Device-Desc-Spec_10012_dV113_Jul20

Prepared, approved and released by the IO-Link **Community**

Any comments, proposals, requests on this document are appreciated. Please use www.io-link-projects.com for your entries and provide name and email address.

Login: *IO-Link-DD*

Password: *Report*

Important notes:

- The IO-Link **Community** Rules shall be considered prior to the development and marketing of IO-Link products. The document can be downloaded from the www.io-link.com portal.
- Any IO-Link device shall provide an associated IODD file.
- Any device which uses the IODD as device description shall provide easy access to the file and potential updates. It is the responsibility of the device manufacturer to test the IODD file with the help of the IODD-Checker tool available per download from www.io-link.com.

Disclaimer:

The attention of adopters is directed to the possibility that compliance with or adoption of IO-Link **Community** specifications may require use of an invention covered by patent rights. The IO-Link **Community** shall not be responsible for identifying patents for which a license may be required by any IO-Link **Community** specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. IO-Link **Community** specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

The information contained in this document is subject to change without notice. The material in this document details an IO-Link **Community** specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of this specification in any company's products.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE IO-LINK **COMMUNITY** MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR USE.

In no event shall the IO-Link **Community** be liable for errors contained herein or for indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party. Compliance with this specification does not absolve manufacturers of IO-Link equipment, from the requirements of safety and regulatory agencies (TÜV, BIA, UL, CSA, etc.).

IO-Link® is registered trade mark. The use is restricted for members of the IO-Link **Community. More detailed terms for the use can be found in the IO-Link **Community** Rules on www.io-link.com.**

Conventions:

In this specification the following key words (in **bold** text) will be used:

- | | |
|----------------------------|--|
| may: | indicates flexibility of choice with no implied preference. |
| should: | indicates flexibility of choice with a strongly preferred implementation. |
| shall: | indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interoperability and to claim conformity with this specification. |
| highly recommended: | indicates that a feature shall be implemented except for well-founded cases. Vendor shall document the deviation within the user manual and within the manufacturer declaration. |

Publisher:

IO-Link **Community**

Haid-und-Neu-Str. 7

76131 Karlsruhe

Germany

Phone: +49 721 / 96 58 590

Fax: +49 721 / 96 58 589

E-mail: info@io-link.com

Web site: www.io-link.com

© No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

Content

94		
95	1	Introduction7
96	2	Related documents and references7
97	2.1	References.....7
98	2.2	Related documents.....8
99	3	Definitions and abbreviations.....8
100	3.1	Definitions8
101	3.2	Abbreviated terms8
102	4	Basic structure9
103	5	Files10
104	5.1	Main IODD file.....10
105	5.2	Language files (optional)11
106	5.3	Image files (optional)11
107	5.4	Standard definitions files12
108	5.5	Schema files12
109	6	Description mechanisms.....12
110	6.1	Names of elements and attributes12
111	6.2	Ids.....13
112	6.3	Referencing.....13
113	6.4	Text localization13
114	7	Device Description13
115	7.1	Notation of XML structure14
116	7.2	Basic structure of the main IODD file16
117	7.3	Metainformation16
118	7.3.1	DocumentInfo (m; o for language file)16
119	7.3.2	ProfileHeader (m)17
120	7.3.3	ProfileBody (m).....17
121	7.3.4	File validation18
122	7.4	Device identity.....19
123	7.4.1	Device variant collection20
124	7.5	Device function21
125	7.5.1	Features22
126	7.5.2	Data type collection23
127	7.5.3	Data types24
128	7.5.3.1	Simple data types25
129	7.5.3.1.1	General25
130	7.5.3.1.2	BooleanT25
131	7.5.3.1.3	UIntegerT26
132	7.5.3.1.4	IntegerT.....27
133	7.5.3.1.5	Float32T27
134	7.5.3.1.6	StringT.....28
135	7.5.3.1.7	OctetStringT29
136	7.5.3.1.8	TimeT29
137	7.5.3.1.9	TimeSpanT29
138	7.5.3.2	Complex data types30
139	7.5.3.2.1	General30
140	7.5.3.2.2	Arrays.....30

141	7.5.3.2.3	Records	33
142	7.5.3.3	Process data union data types	38
143	7.5.3.3.1	ProcessDataInUnionT	38
144	7.5.3.3.2	ProcessDataOutUnionT	39
145	7.5.4	Variable collection	39
146	7.5.4.1	StdVariableRef	39
147	7.5.4.2	DirectParameterOverlay	43
148	7.5.4.3	Variable	45
149	7.5.4.4	RecordItemInfo	47
150	7.5.5	Process data collection	48
151	7.5.6	Error type collection	51
152	7.5.7	Event collection	53
153	7.5.8	User interface	54
154	7.5.8.1	ProcessDataRef collection (o)	55
155	7.5.8.2	MenuSets (m)	56
156	7.5.8.3	Menu collection	58
157	7.5.8.4	VariableRef	59
158	7.5.8.5	RecordItemRef	64
159	7.5.8.6	MenuRef	65
160	7.5.9	Rules for write-only variables	66
161	7.6	Communication characteristics	66
162	7.7	Language dependent description texts	73
163	7.7.1	PrimaryLanguage (m)	74
164	7.7.2	Language (o)	75
165	8	Compatibility	75
166	Annex A	IODD schemas	76
167	Annex B	Definitions of IODD quantity structure	77
168	Annex C	IODD test (normative)	78
169	1.1	General	78
170	1.2	Schema test via an IODD Checker tool	78

171
172
173

Figures

174	Figure 1	– Structure of main IODD file following ISO 15745-1	9
175	Figure 2	– Structure of language file	11
176	Figure 3	– Basic structure of main IODD file	16
177	Figure 4	– DocumentInfo element	16
178	Figure 5	– ProfileHeader element	17
179	Figure 6	– ISO15745Reference element	17
180	Figure 7	– Stamp element	18
181	Figure 8	– DeviceIdentity element	19
182	Figure 9	– DeviceVariantCollection element	21
183	Figure 10	– DeviceFunction element	22
184	Figure 11	– Features element	22
185	Figure 12	– DatatypeCollection element	24
186	Figure 13	– Data type hierarchy	24

187	Figure 14 – BooleanT	25
188	Figure 15 – UIntegerT	26
189	Figure 16 – Float32T	27
190	Figure 17 – StringT	28
191	Figure 18 – OctetStringT	29
192	Figure 19 – TimeT	29
193	Figure 20 – TimeSpanT	29
194	Figure 21 – ArrayT	30
195	Figure 22 – RecordT	33
196	Figure 23 – ProcessDataInUnionT	39
197	Figure 24 – ProcessDataOutUnionT	39
198	Figure 25 – VariableCollection element.....	39
199	Figure 26 – StdVariableRef element	41
200	Figure 27 – StdRecordItemRef element	43
201	Figure 28 – DirectParameterOverlay element.....	44
202	Figure 29 – Direct parameter overlay	45
203	Figure 30 – Variable element	46
204	Figure 31 – RecordItemInfo element	47
205	Figure 32 – ProcessDataCollection element.....	48
206	Figure 33 – Condition element	49
207	Figure 34 – ProcessDataIn element	50
208	Figure 35 – ProcessDataOut element.....	51
209	Figure 36 – ErrorTypeCollection element	51
210	Figure 37 – StdErrorTypeRef element	52
211	Figure 38 – ErrorType element	52
212	Figure 39 – EventCollection element.....	53
213	Figure 40 – StdEventRef element	53
214	Figure 41 – Event element	54
215	Figure 42 – UserInterface element	54
216	Figure 43 – ProcessDataRefCollection element	55
217	Figure 44 – ProcessDataInfo element	55
218	Figure 45 – ProcessDataRecordItemInfo element	56
219	Figure 46 – <Role>MenuSet element	57
220	Figure 47 – MenuCollection element	58
221	Figure 48 – VariableRef element.....	59
222	Figure 49 – Event data structure	63
223	Figure 50 – Button element.....	63
224	Figure 51 – RecordItemRef element.....	64
225	Figure 52 – MenuRef element.....	65
226	Figure 53 – CommNetworkProfile element	66
227	Figure 54 – CommNetworkProfile element – IO-Link variant.....	66
228	Figure 55 – TransportLayers element – IO-Link variant.....	67
229	Figure 56 – Test element	68

230	Figure 57 – ConnectionT abstract type.....	69
231	Figure 58 – Connection element – M5ConnectionT variant.....	71
232	Figure 59 – Connection element – M12-5ConnectionT variant.....	72
233	Figure 60 – Connection element – OtherConnectionT variant.....	73
234	Figure 61 – ExternalTextCollection element.....	74
235	Figure 62 – PrimaryLanguage element.....	74

236
237

Tables

238	Table 1 – IODD XML types	15
239	Table 2 – Allowed combinations of datatype, displayFormat, gradient and offset.....	60
240	Table 3 – Standard variables with special display	62
241	Table 4 – Wire colors	70
242	Table 5 – Wire functions	70
243	Table 6 – IODD quantity structure	77
244	Table 7 – Checker business rule set for IODDs	78

245
246

Revision Log

247

Version	Originator	Date	Change Note / History / Reason
1.1	Team Technology	09-August-2011	Final version.
1.1.3	CC/PG30 - IODD	2020-07-31	Document in Review

248

1 Introduction

An IODD (IO Device Description) is a set of files that formally describes a device e.g. IO-Link Device.

The IODD is created by the device vendor and shall be sufficient for IO-Link Tools to identify, communicate, parameterize and diagnose the device.

The set of files consists of the main IODD file, optional language files and optional picture files.

An IODD is mandatory for each IO-Link Device. This specification defines the IODD for IO-Link Devices that conform to the **IO-Link Interface and System Specification Version 1.1.3**.

2 Related documents and references

2.1 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.

IO-Link Interface and System Specification Version 1.1.3, June 2019, Order No: 10.002

IO-Link Test Specification Version 1.1.3, Order No: 10.032

ANSI/IEEE Std 754-2008, *IEEE Standard for Binary Floating-Point Arithmetic*

IETF RFC 2083, *PNG (Portable Network Graphics) Specification Version 1.0*, available at <http://tools.ietf.org/html/rfc2083>

ISO 639-1:2002, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*

ISO 15745-1:2003, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*

ISO 15745-1 Amd 1:2007, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description, Amendment 1*

IEC 60757:1983, *Code for designation of colours*

The Unicode Standard, **V11.0.0**, available at <http://www.unicode.org/>

ITU-T recommendation V.42 (03/2002), *Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion*, available at <http://www.itu.int/rec/T-REC-V.42-200203-I/en>

REC-xml-20081126, *Extensible Markup Language (XML) 1.0 (Fifth Edition) – W3C Recommendation 26 November 2008*, available at <http://www.w3.org/TR/xml/>

REC-xmlschema-1-20041028, *XML Schema Part 1: Structures Second Edition – W3C Recommendation 28 October 2004*, available at <http://www.w3.org/TR/xmlschema-1/>

REC-xmlschema-2-20041028, *XML Schema Part 2: Datatypes Second Edition – W3C Recommendation 28 October 2004*, available at <http://www.w3.org/TR/xmlschema-2/>

2.2 Related documents

ANSI INCITS 4-1986 (R2007), *Information Systems – Coded Character Sets – 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)* (predecessor of ISO/IEC 646)

IEC 60559:2011, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

IETF RFC 3629, *UTF-8, a transformation format of ISO 10646*, available at <http://tools.ietf.org/html/rfc3629>

IETF RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, available at <http://tools.ietf.org/html/rfc5905>

ISO/IEC 13239:2002, *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures*

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO/IEC 10646:2003/Amd 7:2017, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 15948:2004, *Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification*

REC-xslt-19991116, *XSL Transformations (XSLT), Version 1.0, W3C Recommendation 16 November 1999*, available at <http://www.w3.org/TR/xslt>

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this document, the terms and definitions given in ISO 15745-1:2003 apply.

IO-Link Tool

Device engineering tool for the IO-Link Master and the connected IO-Link Devices. Used for parameterization and diagnosis of IO-Link Devices on the basis of the IODD.

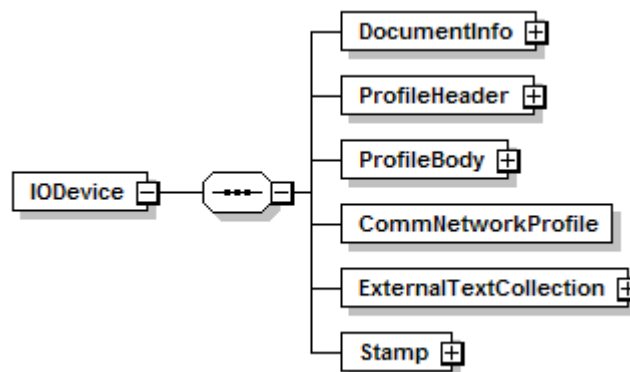
3.2 Abbreviated terms

ANSI	American National Standards Institute (http://www.ansi.org/)
ASCII	American Standard Code for Information Interchange (see ANSI INCITS 4-1986 (R2007) and the US variant of ISO/IEC 646:1991)
BIPM	Bureau International des Poids et Mesures (http://www.bipm.org/)
C/Q	Connection for communication (C) or switching (Q) signal (SIO)
CRC	Cyclic Redundancy Check
DI	Digital Input
DO	Digital Output
I/Q	NC, DI or DO
IEC	International Electrotechnical Commission (http://www.iec.ch/)
IEEE	Institute of Electrical and Electronics Engineers (http://www.ieee.org/)
IETF	Internet Engineering Task Force (http://www.ietf.org/)
IO or I/O	Input / Output

325	IODD	IO Device Description
326	ISDU	Indexed Service Data Unit
327	ISO	International Standardization Organisation (http://www.iso.org/)
328	ITU	International Telecommunication Union (http://www.itu.int/)
329	LF	Line Feed
330	MSXML	Microsoft XML Core Services (see http://msdn.microsoft.com/en-us/library/ms763742%28VS.85%29.aspx)
331		
332	NC	Not Connected
333	PLC	Programmable Logic Controller
334	PNG	Portable Network Graphics (see RFC 2083 and ISO/IEC 15948:2004)
335	RFC	Request for Comments
336	SIO	Standard Input Output (digital switching mode)
337	UCS	Universal Multiple-Octet Coded Character Set (see <i>The Unicode Standard</i> or ISO/IEC 10646:2003/Amd 7:2010)
338		
339	UL	Underwriters Laboratories (http://www.ul.com/)
340	UTC	Coordinated Universal Time (Temps Universel Coordonné) (coordinated by the BIPM) (corresponds to GMT = Greenwich Mean Time)
341		
342	UTF	UCS Transformation Format (see <i>The Unicode Standard</i> or ISO/IEC 10646:2003/Amd 7:2010)
343		
344	W3C	World Wide Web Consortium (http://www.w3.org/)
345	XML	Extensible Markup Language (see REC-xml-20081126)
346	XSD	XML Schema Definition (see REC-xmlschema-1-20041028 and REC-xmlschema-2-20041028)
347		
348	XSL	Extensible Stylesheet Language
349	XSLT	XSL Transform (see REC-xslt-19991116)

350 **4 Basic structure**

351 The following figure shows the basic structure of the main IODD file. It follows the ISO 15745-
 352 1:2003 standard regarding the device profile and communication network profile. It consists of
 353 the elements DocumentInfo, ProfileHeader, ProfileBody, CommNetworkProfile, ExternalText-
 354 Collection and the Stamp.



355
 356 **Figure 1 – Structure of main IODD file following ISO 15745-1**

5 Files

Conceptionally, the IO Device Description consists of the set of files created by the device vendor, and the set of standard definition files which are part of this specification. IO-Link Tools combine information from both sets of files to get the complete device description.

All IODD XML files shall use “UTF-8” for the encoding. They shall use the namespace <http://www.w3.org/2001/XMLSchema-instance> with the prefix “xsi” and the namespace <http://www.io-link.com/IODD/2010/10> with the prefix “iodd”. A schemaLocation for the namespace <http://www.io-link.com/IODD/2010/10> to the required schema shall be given. For the main IODD file, this is IODD1.1.xsd, and for the language files this is IODD-Primitives1.1.xsd. The schema file name shall be given without any path prefix. No other namespaces shall be described. IODD XML files shall not use any DTD (Document Type Definition, see <http://www.w3.org/TR/xml/#sec-prolog-dtd>).

All XMLs generated by the vendor shall be checked by the IODD Checker software before delivery. This Checker is a tool available from the IO-Link web site (<http://www.io-link.com/>). It checks the content of the device description and if no errors were found writes a checksum over the file contents into the element Stamp at the end of the XML-file.

IO-Link Tools shall compare the checksum in the Stamp with the checksum calculated from the file contents. It is recommended to reject the IODD if there is a mismatch. Tools may then omit schema validation and additional checks.

IO-Link Tools shall use the file name of the IODD only to discover the language files that are associated with the main IODD file. Apart from that, tools shall not evaluate the file name; they always evaluate the file’s content. The device-specific file name is only intended for better legibility.

Adherence to the rules for file names makes it possible that all IODDs can be stored side-by-side in a single directory.

File names must not only be different in upper and lower case. Case sensitivity of default parts of file names shall be adhered to.

The following special characters are permitted in vendor name and device name: _, #, -

All files of the set of files belonging to a specific IODD shall have the same <vendor name> part in their file names. The <vendor name> should be the same for all IODDs of the same vendor. It is not required that the <vendor name> in the file name matches exactly the content of the DeviceIdentity/@vendorName attribute or the standard variable V_VendorName in the device. Usually, the latter two also contain the legal form of the company, e.g. “Inc.”, “AG”, “S.A.” but this is not included in the vendor name part of the file name.

5.1 Main IODD file

The file name shall follow the following rule:

<vendor name>-<device name>-<date of file creation>-**IODD**<schema version>.xml

e.g. VendorX-DeviceY-20110603-IODD1.1.xml

Contains information (in XML) about the identification of the device, communication characteristics, parameters, process data and diagnosis data.

The IODD must always entirely contain texts in the PrimaryLanguage (English). The IODD may contain texts in further languages.

A style sheet for the vendor-specific description of Devices for a certain browser (optional):

e.g. VendorX-IODD1.1.xsl

The IODD shall not reference such style sheets with a processing instruction (<?xml-stylesheet ... ?>).

5.2 Language files (optional)

To add support for additional languages after an IODD has been released, separate language files (in XML) may be created. Their file name must exactly match the name of the main IODD file, except that there is an additional language designation before the file name extension:

```
<vendor name>-<device name>-<date of file creation>-IODD<schema version>-<language>.xml
```

The “language” part follows ISO 639-1:2002. The “language” part shall correspond to the value of the ‘xml:lang’ attribute inside the language file. There shall be no additional language file for languages already covered in the main IODD file. The “language” part consists of two letters.

The ‘Text’ and ‘TextRedefine’ elements contained in the additional language file shall follow the same rules as specified for the respective elements in additional languages inside the main IODD.

e.g. VendorX-DeviceY-20110603-IODD1.1-ru.xml

Additional language file containing texts in Russian.

An IO-Link Tool shall select the appropriate language from the main IODD file or the accompanying language files according to its user interface language settings. A tool shall ignore files whose filename does not match to the naming convention of additional IODD files.

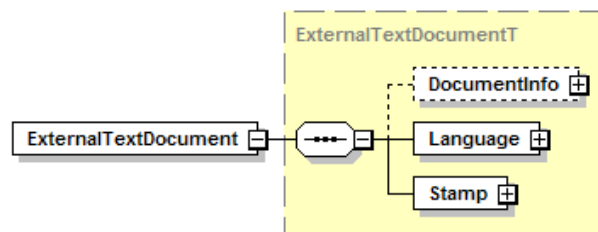


Figure 2 – Structure of language file

5.3 Image files (optional)

The file format shall be PNG (file extension .png, see RFC 2083 or ISO/IEC 15948:2004). The same rules for permitted characters apply as in section ‘Files’ (see above).

<vendor name>-logo.png

Vendor logo. 160 x 90 pixel, landscape format. The background of the logo should be transparent.

<vendor name>-<picture name>-icon.png

Device variant icon. 48 x 48 pixel.

<vendor name>-<picture name>-pic.png

Device variant picture. Min. 160 x 160 pixel, max. 320 x 320, square.

<vendor name>-<picture name>-con-pic.png

Device variant connection picture. Min. 160 x 160 pixel, max. 320 x 320, square.

The device variant icons and device variant pictures are referenced from the DeviceIdentity/ DeviceVariantCollection/DeviceVariant elements. The device variant connection pictures are referenced from the CommNetworkProfile/TransportLayers/PhysicalLayer/Connection

elements. The referenced image files shall accompany the main IODD file for stamping and delivery.

5.4 Standard definitions files

IODD-StandardDefinitions1.1.xml

This file contains the definition of standardized variables, error types and events (see [IO-Link Interface and System Specification Version 1.1.3](#)) plus English language texts.

IODD-StandardDefinitions1.1-de.xml, *-es.xml, *-fr.xml, *-it.xml, *-ja.xml, *-ko.xml, *-pt.xml, *-ru.xml, *-zh.xml.

Additional language file containing texts in German, Spanish, French, Italian, Japanese, Korean, Portuguese, Russian, Chinese.

IODD-StandardUnitDefinitions1.1.xml

This file contains the definitions of all available unit codes plus English language texts.

Those files are part of the standard and shall not be changed. Vendors of IO-Link Tools should use those files instead of hard-coding standardized things.

Additional language files for standard definitions files will be provided by the IODD subteam when needed on the IO-Link website.

5.5 Schema files

Schema files are needed to validate the structure of XML-files and to aid in editing.

IODD1.1.xsd

IODD-schema; includes the following sub-schemas:

IODD-Primitives1.1.xsd

includes basic schema elements

IODD-Datatypes1.1.xsd

includes schema elements for the definition of data types

IODD-Events1.1.xsd

includes schema elements for the definition of error types and events

IODD-Variables1.1.xsd

includes schema elements for the definition of variables

IODD-UserInterface1.1.xsd

includes schema elements for the definition of the user interface

IODD-Communication1.1.xsd

includes schema elements for the definition of the communication network profile

IODD-StandardDefinitions1.1.xsd

schema for the definition of system-specific elements used to validate the file IODD-StandardDefinitions1.1.xml and IODD-StandardUnitDefinitions1.1.xml

IODD-WirelessCommunication1.1.xsd

includes schema elements for the definition of the wireless communication network profile

6 Description mechanisms

6.1 Names of elements and attributes

Following one common pattern, the names of the elements begin with an uppercase letter while the names of the attributes begin with a lowercase letter. When names consist of several words, each word (except for the first in case of an attribute) starts with an uppercase letter. No separator character (like `_`) is used.

6.2 Ids

The values of the attribute 'id' shall follow the regular expression pattern:

"[A-Za-z][A-Za-z0-9 _-]*[A-Za-z0-9]".

Ids shall be unique within the elements of the same type. The prefix "STD_" is reserved for ids in the standard definition files and shall not be used for elements in the main IODD.

6.3 Referencing

Each element that can be referenced within the IODD contains an explicit attribute 'id'. The referencing element contains a type-dependent attribute with the following composition: <type>Id

Examples: textId, datatypeId, menuId, variableId

6.4 Text localization

All text components of the different languages which are referenced in the IODD are allocated in the ExternalTextCollection (for further information see "Language-Dependent Description Texts").

The text components of the different languages are referenced in the relevant location according to a key (textId).

Further languages can be added in an appropriate file (see chapter 5).

The PrimaryLanguage in the IODD must be completely available. If there is a further language added in the IODD or in a separate language file, not all entries must be given. In this case, the interpreter has to go back to the entry of the PrimaryLanguage.

7 Device Description

For IO-Link Engineering Tools no conformance classes are specified. IO-Link EngineeringTools shall support the IO-Link Interface and System Specification Version 1.1.3 completely. For interpretation of the IO Device Description Specification V1.1.3 **Error! Reference source not found.** the following requirements shall be fulfilled.

An IO-Link Engineering Tool shall support:

- All IO-Link devices
- If the tool provides a catalog, at least one catalog entry for each IODD (derived from DeviceName)
- All data types
- All Variables incl. StandardVariables
- Menus, including Buttons
- ErrorTypes incl. StandardErrorTypes
- Conditions
- User Role "Specialist"
- Texts in the PrimaryLanguage (English)
- Refresh of dynamic variable values (on demand or cyclic)
- When variables marked with attribute modifiesOtherVariables set to "true" are changed, the tool shall either notify the user that other variables are possibly changed, or the tool shall automatically reload.
- When parameters are up- or downloaded, any ErrorTypes that occur shall be displayed in a way that the user is able to associate them with the parameter that

523 triggered it, either by displaying it at the parameter or by telling the parameter name in
524 the error log.

- 525 • the complete list of error types contained in the standard definition file.
- 526 • the complete list of event codes contained in the standard definition file.
- 527 • Variables with accessRights = "wo" shall never be part of any download sequence
- 528 • Variables with accessRights = "wo" shall always be handled as a single write request
- 529 • The Button assigned text referenced in IODD /Button/Description element to variable
530 StdVariableRef[@id='V_SystemCommand']/StdSingleValueRef shall be shown in a
531 message box with OK and Cancel, see chapter 7.5.8.4 Description
- 532 • if the Button references a SystemCommand or another write-only variable which is
533 defined as SingleValue, tools ought to apply ActionStartedMessage, see chapter
534 7.5.8.4 ActionStartedMessage

535

536 An IO-Link Tool should support, but is not obliged to:

- 537 • Separate catalog entries for each DeviceVariant
- 538 • A separate display of the Process Data (in addition to V_ProcessDataIn /
539 V_ProcessDataOut)
- 540 • Display of the Connection Description
- 541 • Texts in other Languages than the PrimaryLanguage
- 542 • User Roles other than "Specialist"
- 543 • Vendor logo, pictures and icons.
- 544 • For menu entries having gradient and/or offset: additional display of the raw value
- 545 • Names of ValueRanges
- 546 • Events incl. StandardEvents
- 547 • ErrorTypes and Events: additional display of the raw values

548

549 IO-Link Tools distinguish devices by their VendorID and DeviceID, and the IODDs of a device
550 by the IODD version they are based on and their DocumentInfo/@releaseDate. Tools shall
551 prefer V1.0.1 IODDs over V1.0 IODDs, and within the IODD version newer IODDs over older
552 IODDs. For V1.1 IO-Link devices there shall be only a single current (most recent) IODD
553 based on V1.1, and if the device is compatible to IO-Link V1.0, additionally a single current
554 IODD based on V1.0.1 or V1.0.

555 7.1 Notation of XML structure

556 The XML structure is hierarchical. As the whole structure is too complex to show in one figure,
557 the description is split into a series of figures, starting with the root element and descending
558 into the details.

559 Following each figure showing the structure of a particular section of the IODD, all the
560 elements and their attributes are listed in the order in which they appear inside the figure.

561 The description of elements and attributes follows this pattern:

562 **Element_or_attribute_name (Use[, XML_type])**
563 Semantics of the element or attribute. If the element or attribute has a value, a
564 possible default or fixed value is also described here.

565 **Element_or_attribute_name** is the name of the element or attribute. Remember that element
566 names start with an uppercase letter while attribute names start with a lowercase letter.

Use is one of the following letters:

m Mandatory

o Optional

c Conditional (depends on, see description):

Schema is not powerful enough to formulate the complex IODD rules, therefore business logic has to be checked by IODD Checker, see Annex C

XML_type is the XML schema data type of the element or attribute value (if applicable). Do not confuse this with the data types that the device's variables and process data may use. XML_type may be

- one of the basic XML types defined in REC-xmlschema-2-20041028. The namespace "xsd:" is omitted for brevity.

- one of the IODD XML types defined in IODD-Primitives1.1.xsd (see Table 1).

Table 1 – IODD XML types

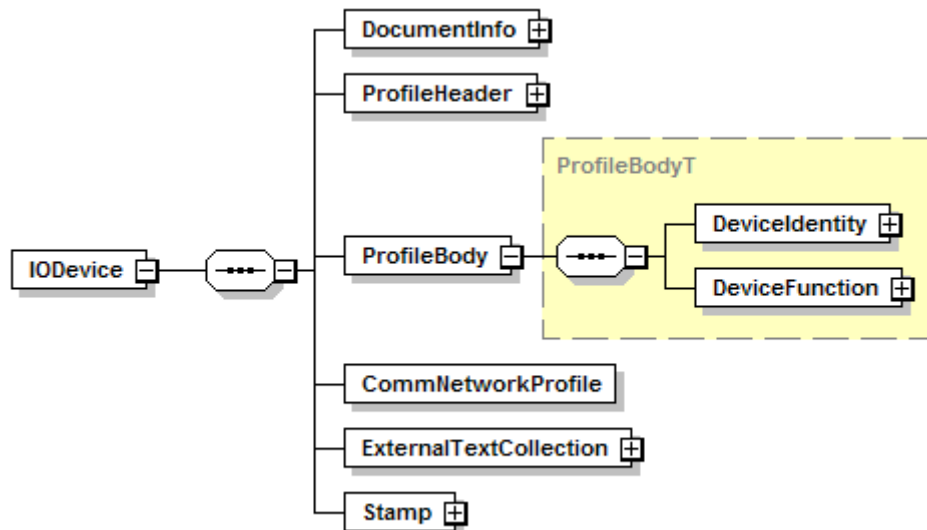
XML_type	Defined as	Use
IdT	xsd:string with pattern: "[A-Za-z][A-Za-z0-9 _-]*[A-Za-z0-9]"	Used for an 'id' attribute at an element so it can be referenced.
RefT	xsd:string with pattern: "[A-Za-z][A-Za-z0-9 _-]*[A-Za-z0-9]" (same as IdT)	Used as a reference to some element that has an 'id' attribute.
SubindexT	xsd:unsignedByte restricted to 1..255 (0x01..0xFF)	For sub-adressing within an index.
BitCountT	xsd:unsignedShort	For bit offsets and bit lengths.
IsduLengthT	xsd:unsignedByte restricted to 1..232	For lengths (in octets) which must fit into an ISDU
DeviceIdT	xsd:unsignedInt restricted to 1..16777215 (0x000001..0xFFFFF)	For a device ID.
CharacterEncodingT	xsd:string, either "UTF-8" or "US-ASCII"	The character encoding of a string.
VersionT	xsd:string with pattern: "\d+(\.\d+){1,7}"	To express a version of e.g. the IO-Link specification, the IODD Checker, the IODD instance.
AccessRightsT	xsd:string, either "ro", "rw" or "wo"	Access rights read only, read-write or write-only.

Further restrictions to these XML types are mentioned directly at the XML type or in the element / attribute description.

583

584 **7.2 Basic structure of the main IODD file**

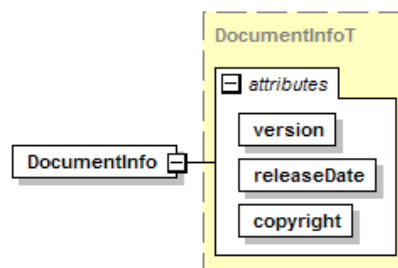
585 Figure 3 shows the basic structure of a device in a device description.



586

587

Figure 3 – Basic structure of main IODD file

588 **7.3 Metainformation**589 **7.3.1 DocumentInfo (m; o for language file)**

590

591

Figure 4 – DocumentInfo element

592 Here the vendor inserts the information for the IODD.

593 **version (m, VersionT)**

594 The 'version' attribute contains the version of the concrete instance and not the version of the IODD specification. The vendor shall increase this version for each official release of the IODD for a particular device.

597 **releaseDate (m, date with pattern "\d{4}-\d{2}-\d{2}")**

598 The date information in the IODD file name shall correspond to the 'releaseDate' attribute in the DocumentInfo element. There shall be no more than one official release of the IODD for a particular device per day. IO-Link Tools shall rely on this date for determining the newest version of the IODD for a device.

602 **copyright (m, string)**

603 Vendor-specific copyright text.

604 e.g.

605 File name: IO-Link-SampleDevice-20200801-IODD1.1.xml

606 DocumentInfo:

607 `<DocumentInfo version="V5.17" releaseDate="2020-08-01" copyright="IO-Link Community"/>`

7.3.2 ProfileHeader (m)

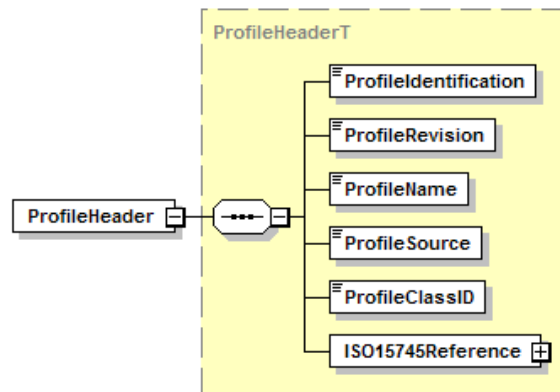


Figure 5 – ProfileHeader element

Within this element, the vendor shall give the following constant information in plain text.

ProfileIdentification (m, string)

Fixed to "IO Device Profile"

ProfileRevision (m, string)

Fixed to "1.1".

ProfileName (m, string)

Fixed to "Device Profile for IO Devices".

ProfileSource (m, string)

Fixed to "IO-Link Consortium"

ProfileClassID (m, NMTOKEN)

Fixed to "Device".

ISO15745Reference (m)

Information about the underlying ISO standard

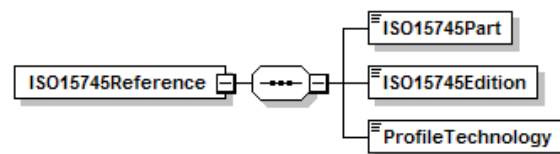


Figure 6 – ISO15745Reference element

ISO15745Part (m, positiveInteger)

Fixed to "1".

ISO15745Edition (m, positiveInteger)

Fixed to "1".

ProfileTechnology (m, string)

Fixed to "IODD".

7.3.3 ProfileBody (m)

The ProfileBody contains the description of identity and functionality of the device.

7.3.4 File validation

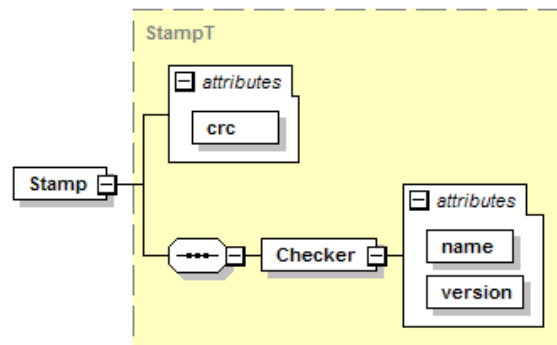


Figure 7 – Stamp element

There is a tool called "IODD Checker" that intensely tests the IODD if it is conformant to this specification. The Checker features a *check* and a *stamp* mode. In *check* mode, errors detected during the checking process are reported, but the file remains unchanged. In *stamp* mode, the Stamp element is always rewritten.

crc (m, unsignedInt)

If no errors are detected during the checking process, the 'crc' attribute is set to the CRC value calculated from the file contents. Otherwise, the 'crc' attribute is set to an invalid value. By checking the CRC, an IO-Link Tool can find out whether the IODD has been altered since the last successful check. In this case, the IODD should be rejected by the tool.

For the CRC, the CRC-32 algorithm is used (see section 8.1.1.6.2 of ITU-T recommendation V.42 (03/2002) or ISO/IEC 13239:2002). Before the actual calculation, the 'crc' attribute is set to an empty string and the checker inserts its name and version into the appropriate attributes. The generated CRC is then inserted into the 'crc' attribute.

The CRC calculation is done as follows: The IODD file is read in binary mode. The stream of bytes is fed into the CRC algorithm until the string `<Stamp crc="` has been processed. The value of the attribute `crc` is skipped, and the CRC calculation continues with the closing quotation mark.

The same is done with external language documents, but after the end-of-file has been reached, the CRC of the main IODD file is converted to decimal representation (no leading zeroes) and the character codes for the digits are fed into the CRC algorithm.

Checker (m)

Identification of the IODD Checker version used to check and stamp this file. If there is a severe bug in a specific Checker version, or the method of calculating the CRC must be modified in the future, IO-Link Tools are able to adapt to this based on the Checker name and version.

name (m, string)

The name of the IODD Checker.

version (m, VersionT)

The version of the IODD Checker.

When writing a new IODD, before applying the IODD checker on it for the first time, it is recommended to set the attributes to the following values:

```

Stamp/@crc = "0"
Stamp/Checker/@name = "" (empty string)
Stamp/Checker/@version = "V0.0.0.0"

```

It is highly recommended, not to insert comments in or after the Stamp element.

7.4 Device identity

On import of a new IODD, IO-Link Tools shall use the pair vendorId and deviceId to decide whether this IODD describes a new device (catalog entries must be added) or this IODD is a new description of an already known device (catalog entries must be updated). This decision shall not be based on the filename of the IODD.

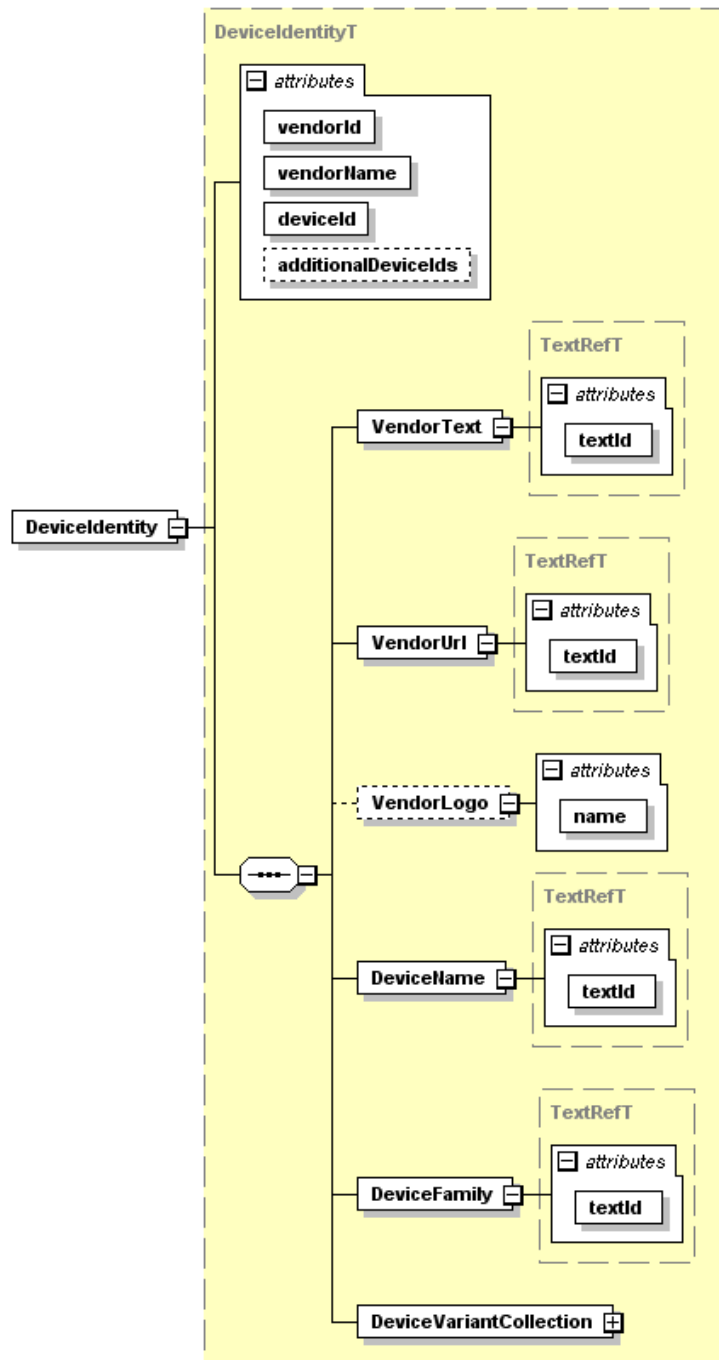


Figure 8 – DeviceIdentity element

vendorId (m, unsignedShort)

Unique identification of the vendor, assigned by the IO-Link **Community**. An IO-Link Tool shall display this id in decimal notation. The vendorId shall match the content of V_DirectParameters_1, subindex 8-9. The allowed value range is 1..65535 (0x0001..0xFFFF).

vendorName (m, string)

Name of the vendor of the device.

684 deviceld (m, DeviceIdT)

685 Vendor-internal unique identification of the device; an IO-Link Tool shall display this id in
686 decimal notation. The deviceld shall match the content of V_DirectParameters_1, subindex 10-
687 12.

688 additionalDevicelds (o, list of DeviceIdT with min. one and max. 255 entries)

689 A list of device IDs which are supported by this device. In other words, this device may replace
690 devices of the same vendor whose device IDs are listed in this attribute.

691 VendorText (m)**692 textId (m, RefT)**

693 A text describing the vendor (a slogan).

694 VendorUrl (m)**695 textId (m, RefT)**

696 The vendor's URL.

697 VendorLogo (o)**698 name (m, string with pattern "([p{L}\d_#]+)+logo.png")**

699 File name of the vendor's logo; in PNG format, 160 x 90 pixels. If the element 'VendorLogo' is
700 used, the image file referenced by the attribute 'name' shall be present.

701 DeviceName (m)**702 textId (m, RefT)**

703 Common name for all variants. IO-Link Tools may use this to group the device variants of a
704 device in the device catalog.

705 DeviceFamily (m)**706 textId (m, RefT)**

707 Vendor-specific classification of the devices. IO-Link Tools may use this for grouping devices
708 in the device catalog.

709 7.4.1 Device variant collection

710 Using the Vendor ID and Device ID read out from an unknown IO-Link Device, it must be
711 possible to uniquely find the latest version of the appropriate IODD. **All variants referenced in**
712 **the Device variant collection share the same data model and menu representation. They may**
713 **differ only in the elements Variant and Connection.** Thus is it not allowed that IO-Link Devices
714 that differ in details described in ProfileBody/DeviceFunction or **CommNetworkProfile except**
715 **for element Connection** share the same combination of Vendor ID and Device ID.

716

717 The things in which the devices may differ are those that are not "seen" by the IO-Link Tool,
718 like:

- 719 • type of plug and length of cable
- 720 • materials: plastics, stainless steel
- 721 • shape: round, ..
- 722 • fastening: through-hole, bracket
- 723 • allowed environmental conditions: temperature range, humidity, shock resistance
- 724 • certificates: CE, UL

725 Devices that only differ in these things may use the same Vendor ID and Device ID and shall
726 be described as different Device Variants in a single IODD. If the device supports ISDU Index
727 19 (V_ProductID), the value read from this ISDU must match exactly to the 'productId' attribute
728 of exactly one DeviceVariant.

729 Examples for things in which devices may **not** differ:

- 730 • measurement ranges (with sensors)

- power range (with actuators)

There shall be at least one device variant.

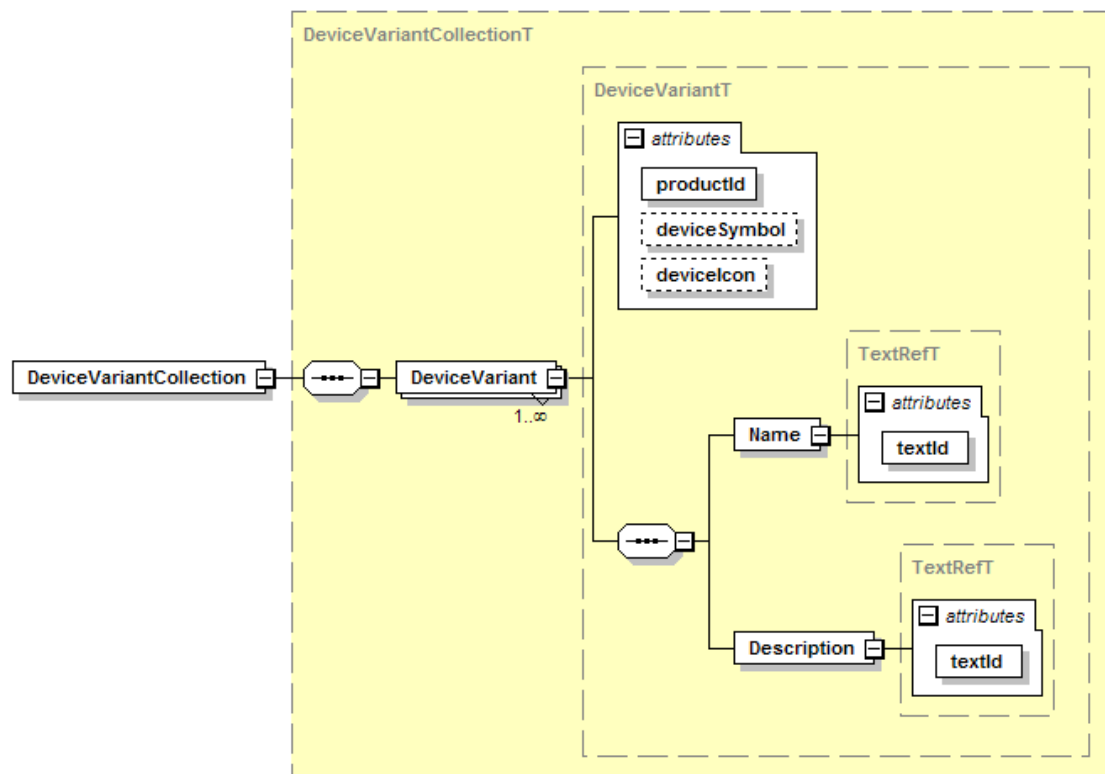


Figure 9 – DeviceVariantCollection element

productid (m, string)

Text; max. 64 octets UTF-8 coded. Uniquely identifies the product within the DeviceVariants. 'productid' in IODD corresponds to the ISDU standard parameter V_ProductID. If V_ProductID is not implemented in the device only one single device variant shall be referenced in the IODD.

deviceSymbol (o, string with pattern "([p{L}\d_#]+)+pic\.png")

File name of the device symbol. If this attribute is used, the referenced image file shall be present.

deviceIcon (o, string with pattern "([p{L}\d_#]+)+icon\.png")

File name of the device icon. If this attribute is used, the referenced image file shall be present.

Name (m)

textid (m, RefT)

Used to build the catalog entries for the device variants in the IO-Link Tool. Shall be unique for each DeviceVariant **within each** supported language.

It shall correspond to the product name in the vendor's catalogue or to the name which is labelled on the product.

Description (m)

textid (m, RefT)

Descriptive text of the device.

7.5 Device function

The entire functionality of the device is collected here. Parameters, process data, data types, error codes and events are defined. Their significances, addresses, and data fields are identified as well as a grouping of the views in menus is defined.

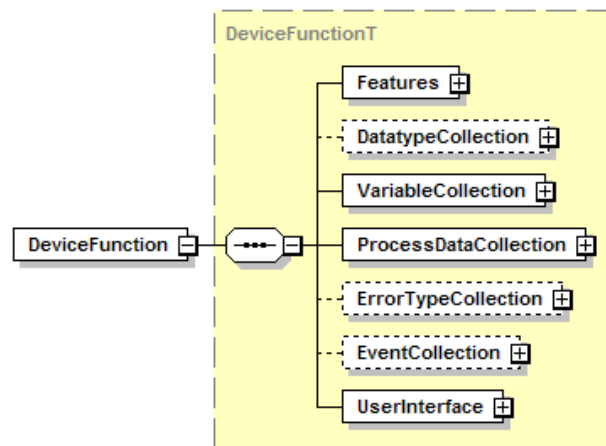


Figure 10 – DeviceFunction element

7.5.1 Features

Supported standardized features of the device are described.

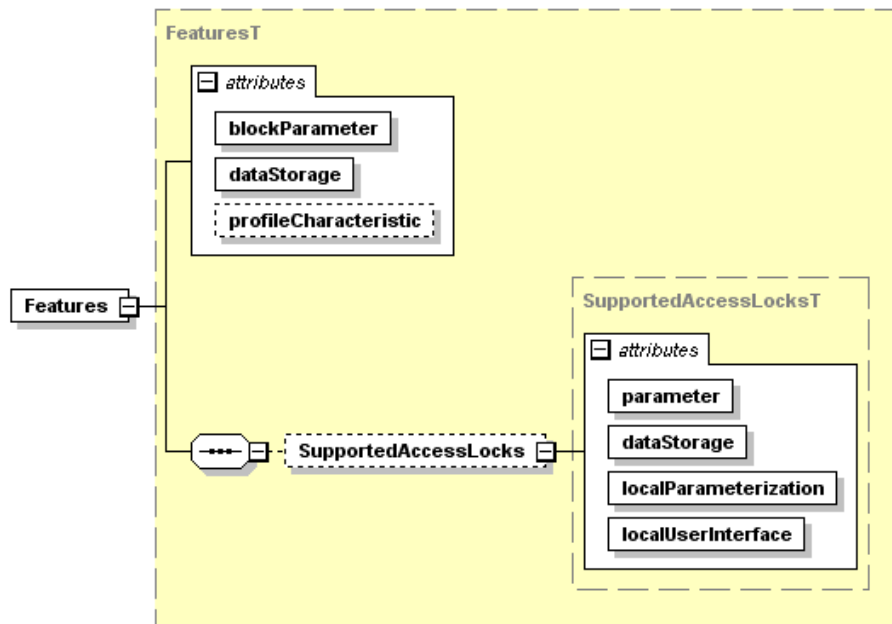


Figure 11 – Features element

blockParameter (m, boolean)

This attribute defines if a device supports the functionality of Block Parameter transmission. See [IO-Link Interface and System Specification Version 1.1.3](#), chapter 10.3.5.

IO-Link Tools shall use Block Parameter transmission if the device supports it and if more than one variable is to be transferred due to a single user action.

Typical Block Parameter sequences:

Upload

- Send SystemCommand ParamUploadStart
- Upload all variables of the current user role
- Send SystemCommand ParamUploadEnd

Download

- 775 • Send SystemCommand ParamDownloadStart
- 776 • Download the changed (or all) variables of the current user role
- 777 • Send SystemCommand ParamDownloadEnd or ParamDownloadStore

778

779 dataStorage (m, boolean)

780 This attribute defines if a device supports data storage functionality.

781 If any standard ISDU variable with accessRights = "rw" is present, this attribute shall not be set
782 to "false".

783 If any device specific ISDU variable with accessRights = "rw" is present, it is highly recommended
784 that this attribute is not set to "false".

785 profileCharacteristic (o, list of unsignedShort with min. one and max. 32 entries)

786 A list of Profile Identifiers (PID) which are supported by this device. This list describes the
787 supported profiles and function classes. This attribute shall be given if any IO-Link Profile is
788 supported.

789 Example: profileCharacteristic="10 49 16384"

790

791 SupportedAccessLocks (c)

792 Shall be present if the standard variable V_DeviceAccessLocks is supported.

793 parameter (m, boolean)

794 Whether parameter access lock is supported. The assigned variable
795 V_DeviceAccessLocks.ParameterAccessLock shall not be referenced in IODD menu.

796 According IO-Link Interface and System Specification Version 1.1.3, it is highly recommended,
797 that parameter lock shall not be implemented in the devices. To provide backward compatibility,
798 the parameter lock bit still has to exist in device (default = "false") and in IODD, but it shall not
799 appear in IODDs menu.

800 dataStorage (m, boolean)

801 Whether data storage access lock is supported. The assigned variable
802 V_DeviceAccessLocks.DataStorageLock shall not be referenced in IODD menu.

803 According IO-Link Interface and System Specification Version 1.1.3, it is highly recommended,
804 that dataStorage lock shall not be implemented in the devices. To provide backward
805 compatibility, the dataStorage lock still has to exist in device (default = "false") and in IODD,
806 but it shall not appear in IODDs menu.

807 localParameterization (m, boolean)

808 Whether local parameterization access lock is supported.

809 localUserInterface (m, boolean)

810 Whether local user interface access lock is supported.

811 7.5.2 Data type collection

812 The DatatypeCollection incorporates all declarations for the reuse of data types (especially
813 useful for records). There shall be no unreferenced Datatype elements. Standardized data types
814 are described in the schema IODD-Datatypes1.1.xsd.

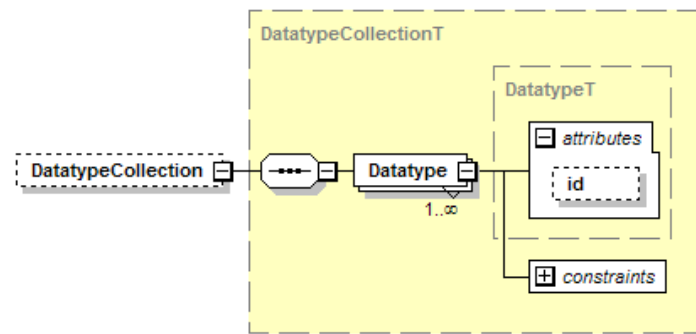


Figure 12 – DatatypeCollection element

For the Datatype element, this figure only shows the elements and attributes common to all data types. The actual selected data type needs additional elements and attributes. See chapter 8 for details.

id (c, IdT)

Datatype elements within the DatatypeCollection shall have an attribute 'id'. Datatype elements outside of the DatatypeCollection shall not have an attribute 'id'.

7.5.3 Data types

The IODD-Datatypes1.1.xsd schema provides derived types for all possible data types. The presence and type of required elements and attributes is checked by this schema.

Actually, the data types form the following hierarchy:

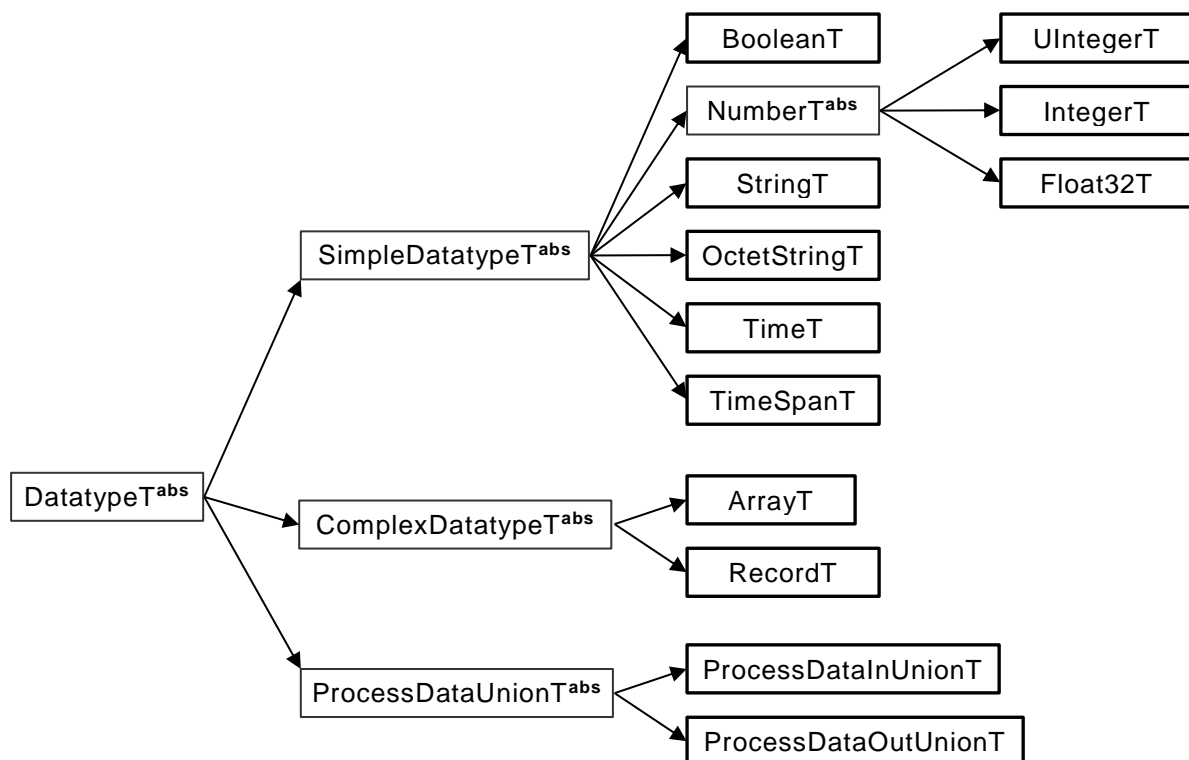


Figure 13 – Data type hierarchy

Each derivation adds elements and/or attributes appropriately.

Excursion on XML schema *abstract types*:

An abstract type can't be used itself. Only non-abstract types which are derived from an

834 abstract type can be used. The instance selects the desired derived type with `xs:type="name`
 835 *of the derived type*".

836 This technique is used here with the 'Datatype' element to adapt the XML structure to the
 837 requirements of the specific data type.

838 For the serialization of the data types see **IO-Link Interface and System Specification Version**
 839 **1.1.3, Annex F**.

840 7.5.3.1 Simple data types

841 The coding of simple data types is shown only for singular use which is characterized by

- 842 • Process data consisting of one simple data type
- 843 • Parameter consisting of one simple data type
- 844 • Subindex (>0) access on individual data items of parameters of complex data types
 845 (arrays, records)

846 7.5.3.1.1 General

847 When the Datatype element appears inside the DatatypeCollection, the attribute 'id' shall be
 848 present. Otherwise, the attribute 'id' shall not be present.

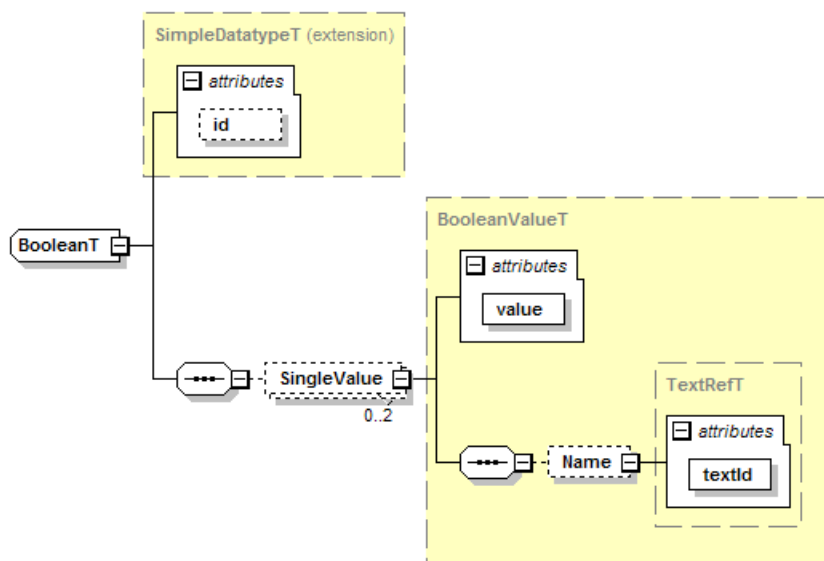
849 SingleValue and ValueRange elements are strongly typed.

850 Where SingleValue and / or ValueRange elements are permitted, the following rules shall be
 851 considered:

- 852 • When neither SingleValue nor ValueRange elements are given, the complete value
 853 range of the data type is allowed. When SingleValue(s) and / or ValueRange(s) are
 854 given, only these values are allowed.
- 855 • In ValueRanges, both the lowerValue and the upperValue are included in the range of
 856 allowed values.
- 857 • In ValueRanges, the lowerValue shall be less than the upperValue (not equal).
- 858 • SingleValues and ValueRanges shall not overlap.

860 7.5.3.1.2 BooleanT

861 Figure 14 shows the IODD representation of the data type BooleanT.



862
 863 **Figure 14 – BooleanT**

Lexical representation:

Conforms to the representation of “boolean” in XML Schema, see
<http://www.w3.org/TR/xmlschema-2/#boolean>

Regular expression pattern: “true|false|1|0”

Example:

```
<Datatype xsi:type="BooleanT">
  <SingleValue value="false">
    <Name textId="TN_Inversion_Off"/>
  </SingleValue>
  <SingleValue value="true">
    <Name textId="TN_Inversion_On"/>
  </SingleValue>
</Datatype>
```

7.5.3.1.3 UIntegerT

Figure 15 shows the IODD representation of the data type UIntegerT.

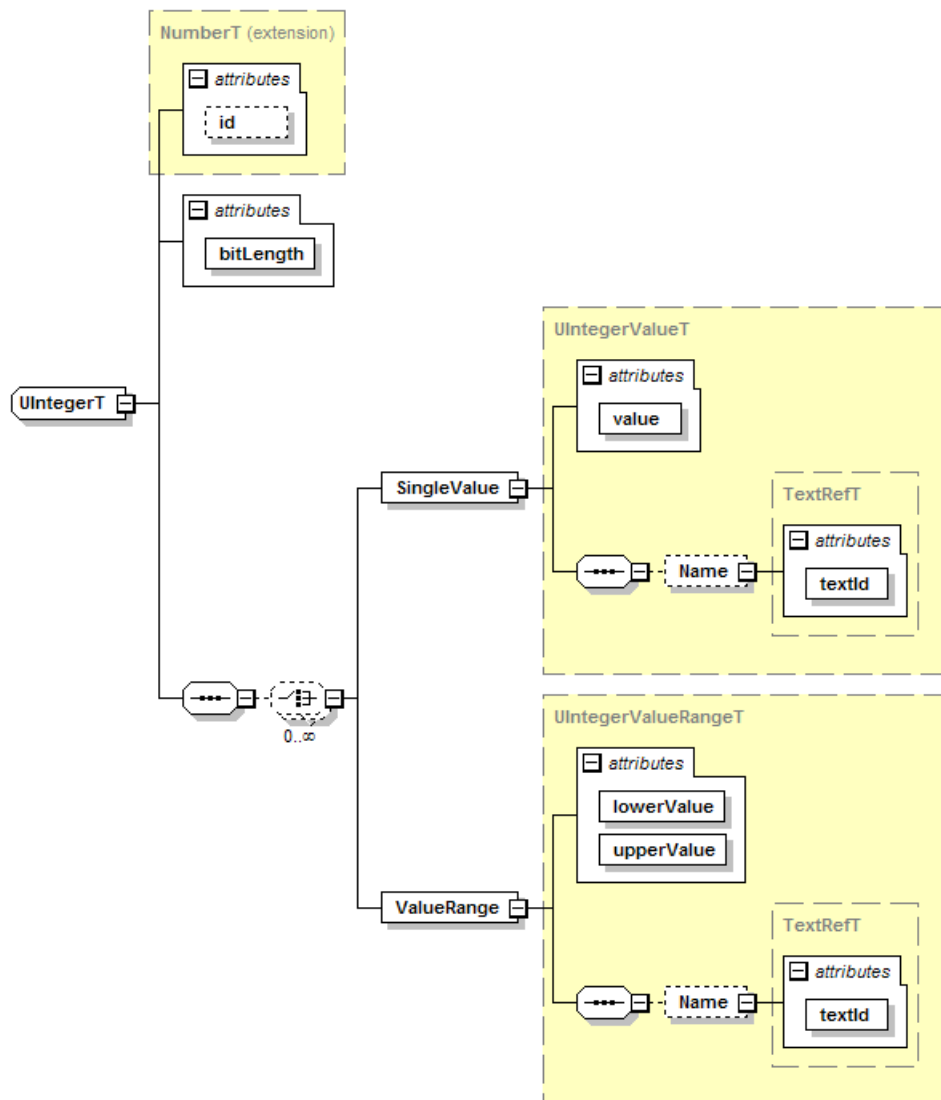


Figure 15 – UIntegerT

bitLength (m, BitCountT)

Specifies the size of the unsigned integer in bits. The allowed value range is 2..64.

885 **Lexical representation:**

886 Conforms to the representation of “unsignedLong” in XML Schema, see
 887 <http://www.w3.org/TR/xmlschema-2/#unsignedLong>

888 Regular expression pattern: “+?\d+”

889 **Example:**

890
 891 `<Datatype xsi:type="UIntegerT" bitLength="8">`
 892 `<SingleValue value="96">`
 893 `<Name textId="TN_System"/>`
 894 `</SingleValue>`
 895 `</Datatype>`

897 **7.5.3.1.4 IntegerT**

898 For the representation of the data type IntegerT in the IODD and an example see 8.

899 **Lexical representation:**

900 Conforms to the representation of “long” in XML Schema, see
 901 <http://www.w3.org/TR/xmlschema-2/#long>

902 Regular expression pattern: “[+-]?\d+”

903 **7.5.3.1.5 Float32T**

904 Figure 16 shows the IODD representation of the data type Float32T.

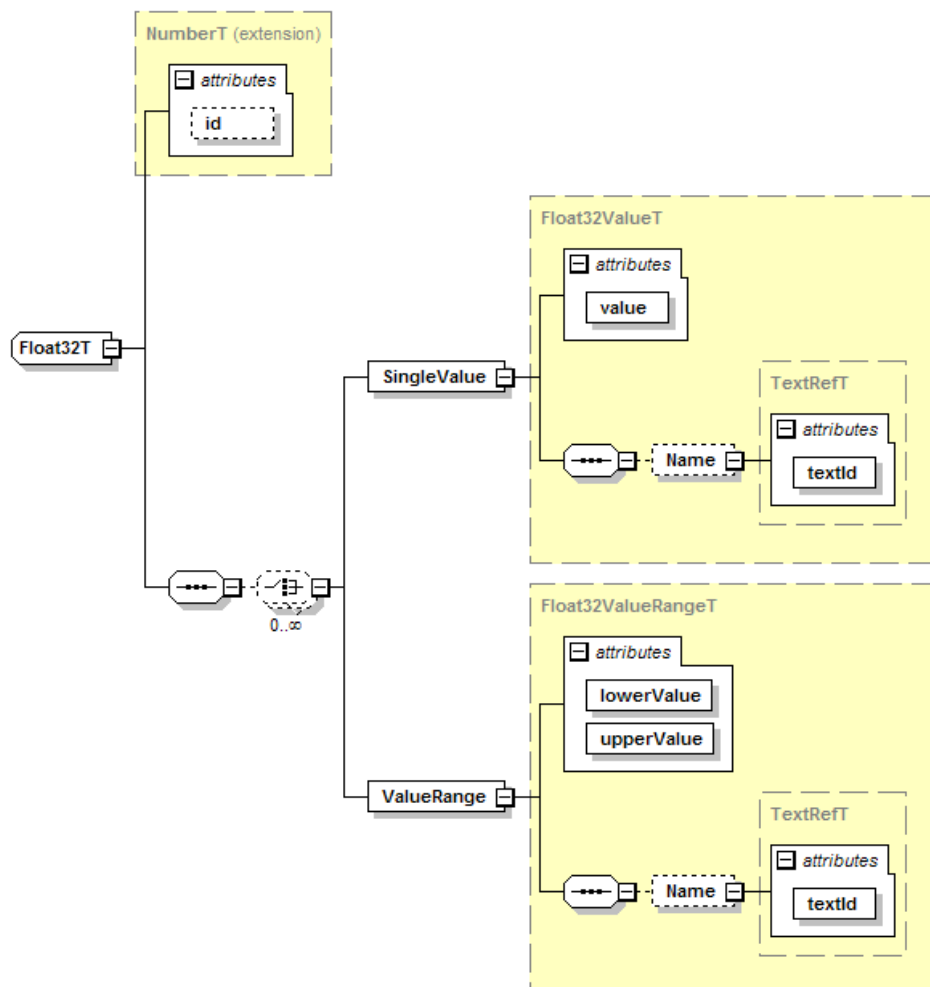


Figure 16 – Float32T

Lexical representation:

Conforms to the representation of “float” in XML Schema, see
<http://www.w3.org/TR/xmlschema-2/#float>

Regular expression pattern: “[+-]?\\d+(\\.\\d+)?([eE][+-]?\\d+)?|-?INF”

Example:

```
<Datatype xsi:type="Float32T">
  <SingleValue value="0.0">
    <Name textId="TN_Zero"/>
  </SingleValue>
  <ValueRange lowerValue="1.0" upperValue="1000.0">
    <Name textId="TN_Valid"/>
  </ValueRange>
</Datatype>
```

7.5.3.1.6 StringT

Figure 17 shows the IODD representation of the data type StringT.

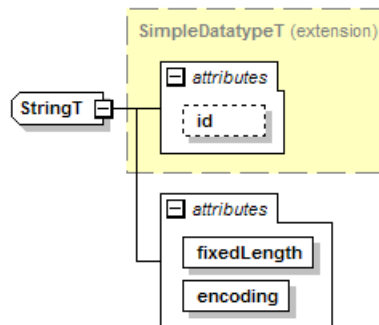


Figure 17 – StringT

fixedLength (m, lsdLengthT)

Specifies the length of the string in octets.

encoding (m, CharacterEncodingT)

The character encoding of the string. Note that US-ASCII consists of 7-bit characters only. Note that string constants in UTF-8 may need more than one octet per character.

Lexical representation:

Conforms to the representation of “string” in XML Schema, see
<http://www.w3.org/TR/xmlschema-2/#string>

Regular expression pattern: “.” (No restriction, just the string.)

Special characters shall be coded according to the XML syntax. See REC-xml-20081126, chapter 2.4 Character Data and Markup.

& → &

‘ → ' (only required if inside a string enclosed in ‘ characters)

> → >

< → <

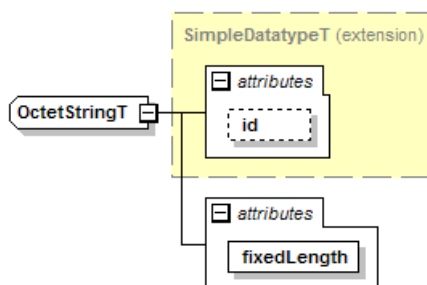
“ → " (only required if inside a string enclosed in “ characters)

Example:

```
<Datatype xsi:type="StringT" fixedLength="64" encoding="UTF-8"/>
```

946 7.5.3.1.7 OctetStringT

947 Figure 18 shows the IODD representation of the data type OctetStringT.



948
949 **Figure 18 – OctetStringT**

950 **fixedLength (m, IsduLengthT)**

951 Specifies the length of the octet string in octets.

952 **Lexical representation:**

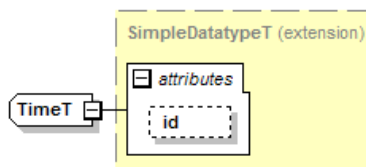
953 Regular expression pattern: "(0x[0-9A-Fa-f][0-9A-Fa-f],)*0x[0-9A-Fa-f][0-9A-Fa-f]"

954 **Example:**

955
956 `<Datatype xsi:type="OctetStringT" fixedLength="10"/>`
957

958 7.5.3.1.8 TimeT

959 Figure 19 shows the IODD representation of the data type TimeT.



960
961 **Figure 19 – TimeT**

962 **Lexical representation:**

963 Follows the representation of "dateTime" in XML Schema, see
964 <http://www.w3.org/TR/xmlschema-2/#dateTime>, but is stricter:

965 Regular expression pattern: "\d{4}\-\d{2}\-\d{2}(T\d{2}:\d{2}:\d{2}(\.\d{1,3})?)?"

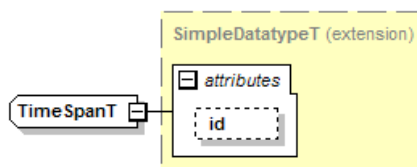
966 (yyyy-mm-dd[Thh:mm:ss[.fff]] where fff = fraction of a second, up to millisecond)

967 **Example:**

968
969 `<Datatype xsi:type="TimeT"/>`
970

971 7.5.3.1.9 TimeSpanT

972 Figure 20 shows the IODD representation of the data type TimeSpanT.



973
974 **Figure 20 – TimeSpanT**

Lexical representation:

Follows the representation of “duration” in XML Schema, see <http://www.w3.org/TR/xmlschema-2/#duration>, but is much stricter:

Regular expression pattern: “[+-]?PT\d+(\.\d{1,3})?S”

Example:

```
<Datatype xsi:type="TimeSpanT"/>
```

7.5.3.2 Complex data types**7.5.3.2.1 General**

Complex data types are combinations of simple data types. Complex data types consist of several simple data types in a packed manner within a sequence of octets. Unused bit space shall be padded with “0”.

The coding of simple data types within complex data types shall be the same as for singular use specified in chapter 8, except for:

BooleanT

The coding of BooleanT is only 1 bit wide. A value of “0” indicates “false” and a value of “1” indicates “true”. There is no padding to an octet.

UIntegerT and IntegerT

The coding of UIntegerT and IntegerT is as wide as indicated by the attribute ‘bitLength’. There is no padding to 1 / 2 / 4 / 8 octets.

7.5.3.2.2 Arrays

Figure 21 shows the IODD representation of the data type ArrayT.

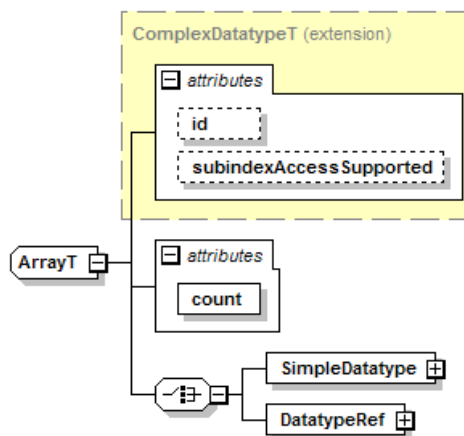


Figure 21 – ArrayT

subindexAccessSupported (o, boolean)

If this attribute is present and set to “false”, individual data items of the array cannot be accessed via their subindex. It is only possible to access the complete array via subindex 0.

count (m, SubindexT)

Specifies the fixed number of data items in the array.

The SimpleDatatype element allows any of the types derived from SimpleDatatypeT. Instead of defining the simple data type inside the array definition, it is also possible to reference the definition of a simple data type from the DatatypeCollection with DatatypeRef/@datatypeId.

Lexical representation:

There is no lexical representation for a value of type ArrayT.

1010 **Alignment:**

1011 Array elements with bitLength < 8 bit shall not cross octet boundaries. For array elements with
1012 bitLength >= 8 bit use bitLength as multiples of 8 bit.

Examples

Notation:

Octet 0 will be transmitted first.

Bit 0 of the octet is right, bit 7 is left.

Octet 0

15	14	13	12	11	10	9	8
				1	0		
0	0	0	0	0	1	0	1

bitOffset in IODD.

Color shows the according subindex, number shows the bit number within the subindex.

Content binary.

Content hexadecimal.

0x05

Bit array

```
<Datatype xsi:type="ArrayT" count="3">
  <SimpleDatatype xsi:type="BooleanT"/>
</Datatype>
```

Subindex	Value
1	true
2	false
3	true

Octet 0

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1

0x05

Integer array

```
<Datatype xsi:type="ArrayT" count="4">
  <SimpleDatatype xsi:type="IntegerT" bitLength="2"/>
</Datatype>
```

Subindex	Value
1	0
2	-1
3	1
4	1

Octet 0

7	6	5	4	3	2	1	0
1	0	1	0	1	0	1	0
0	0	1	0	0	1	0	1

0x25

Integer array

```
<Datatype xsi:type="ArrayT" count="7">
  <SimpleDatatype xsi:type="IntegerT" bitLength="4"/>
</Datatype>
```


1062 bitLength (m, BitCountT)

1063 Specifies the total length of the record in bits. The allowed value range is 1..1856.

1064 RecordItem (m)

1065 An individual item of a record, addressed by the subindex.

1066 subindex (m, SubindexT)

1067 Specifies the Subindex assigned to this record item. The record items shall be ordered by
1068 Subindex within the record. The subindex shall be unique within the RecordItems of a Record.

1069 It is recommended that the Subindices occur in increasing order within the octet sequence. If
1070 Subindices are placed in previously unused areas of the octet sequence, one might deviate
1071 from this recommendation. If compatible extensions are foreseen, it is better to reserve enough
1072 Subindices for the unused areas.

1073 bitOffset (m, BitCountT)

1074 bit position of the record item within the octet sequence. The record items within a record shall
1075 not overlap. The allowed value range is 0..1855.

1076 accessRightRestriction (o, AccessRightsT)

1077 This attribute is only applicable for **variables**, not for **record items within** process data.

1078 Individual record items may have less access rights than the record in general. This is
1079 indicated by the attribute 'accessRightRestriction'. For the access to the complete record, this
1080 means:

- 1081 • If the Record is "rw" and the record item is restricted to "ro" the device shall tolerate
1082 (ignore) the data written to this Subindex.

1083 Within the record item, the SimpleDatatype element allows any of the types derived from
1084 SimpleDatatypeT. Instead of defining simple data types inside the record definition, it is also
1085 possible to reference the definition of simple data types from the DatatypeCollection with
1086 DatatypeRef/@datatypeId.

- 1087 • **The attribute accessRightRestriction shall not be set to "wo". An exception to this rule**
1088 **is the V_DirectParameters_1.**

1089 Name (m)**1090 textId (m, RefT)**

1091 Assigns a human readable name to the record item. This name shall be displayed additionally
1092 by the IO-Link Tool.

1093 Description (o)**1094 textId (m, RefT)**

1095 Contains a description of the RecordItem (e.g. information text, help, etc.)

1096 Lexical representation:

1097 There is no lexical representation for a value of type RecordT.

1098 Alignment

1099 **For variables only, integers shall start on an octet boundary. For optimization of transmission**
1100 **performance, this rule is not applicable for process data.**

1101 **If you have several booleans, it is highly recommended to group them together into one or few**
1102 **octets.**

1103

1104 Examples:

1105 Regarding the notation see 8.

1106 Several Booleans in an Octet

1107

```

1108 <Datatype xsi:type="RecordT" bitLength="4">
1109   <Name textId="TN_Switches"/>
1110   <RecordItem subindex="1" bitOffset="0">
1111     <SimpleDatatype xsi:type="BooleanT"/>
1112     <Name textId="TN_Switch1"/>
1113   </RecordItem>
1114   <RecordItem subindex="2" bitOffset="1">
1115     <SimpleDatatype xsi:type="BooleanT"/>
1116     <Name textId="TN_Switch2"/>
1117   </RecordItem>
1118   <RecordItem subindex="3" bitOffset="2">
1119     <SimpleDatatype xsi:type="BooleanT"/>
1120     <Name textId="TN_Switch3"/>
1121   </RecordItem>
1122   <RecordItem subindex="4" bitOffset="3">
1123     <SimpleDatatype xsi:type="BooleanT"/>
1124     <Name textId="TN_Switch4"/>
1125   </RecordItem>
1126 </Datatype>

```

RecordItem	Subindex	Datatype	bitLength	bitOffset	Value
1	1	BooleanT	—	0	true
2	2	BooleanT	—	1	false
3	3	BooleanT	—	2	true
4	4	BooleanT	—	3	false

Octet 0

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1

0x05

A word and an octet

```

1134 <Datatype xsi:type="RecordT" bitLength="24">
1135   <Name textId="TN_Values"/>
1136   <RecordItem subindex="1" bitOffset="8">
1137     <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
1138     <Name textId="TN_Value1"/>
1139   </RecordItem>
1140   <RecordItem subindex="2" bitOffset="0">
1141     <SimpleDatatype xsi:type="UIntegerT" bitLength="8"/>
1142     <Name textId="TN_Value2"/>
1143   </RecordItem>
1144 </Datatype>

```

RecordItem	Subindex	Datatype	bitLength	bitOffset	Value
1	1	UIntegerT	16	8	0x9876
2	2	UIntegerT	8	0	0x12

Octet 0

23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
1	0	0	1	1	0	0	0

0x98

Octet 1

15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
0	1	1	1	0	1	1	0

0x76

Octet 2

7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0
0	0	0	1	0	0	1	0

0x12

Analog value and two signal bits

```

<Datatype xsi:type="RecordT" bitLength="16">
  <Name textId="TN_ProcessData"/>
  <RecordItem subindex="1" bitOffset="2">
    <SimpleDatatype xsi:type="UIntegerT" bitLength="14"/>
    <Name textId="TN_AnalogValue"/>
  </RecordItem>
  <RecordItem subindex="2" bitOffset="1">
    <SimpleDatatype xsi:type="BooleanT"/>
    <Name textId="TN_Signal2"/>
  </RecordItem>
  <RecordItem subindex="3" bitOffset="0">
    <SimpleDatatype xsi:type="BooleanT"/>
    <Name textId="TN_Signal1"/>
  </RecordItem>
</Datatype>

```

RecordItem	Subindex	Datatype	bitLength	bitOffset	Value
1	1	UIntegerT	14	2	0x32F1
2	2	BooleanT	—	1	false
3	3	BooleanT	—	0	true

Octet 0	Octet 1
15 14 13 12 11 10 9 8 13 12 11 10 9 8 7 6 1 1 0 0 1 0 1 1	7 6 5 4 3 2 1 0 5 4 3 2 1 0 1 1 0 0 0 1 0 1
0xCB	0xC5

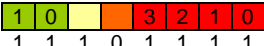
Boolean and enumerations in an octet

```

<Datatype xsi:type="RecordT" bitLength="8">
  <Name textId="TN_ComplexSettings"/>
  <RecordItem subindex="1" bitOffset="0">
    <SimpleDatatype xsi:type="UIntegerT" bitLength="4"/>
    <Name textId="TN_Enum1"/>
  </RecordItem>
  <RecordItem subindex="2" bitOffset="4">
    <SimpleDatatype xsi:type="BooleanT"/>
    <Name textId="TN_Switch1"/>
  </RecordItem>
  <RecordItem subindex="3" bitOffset="5">
    <SimpleDatatype xsi:type="BooleanT"/>
    <Name textId="TN_Switch2"/>
  </RecordItem>
  <RecordItem subindex="4" bitOffset="6">
    <SimpleDatatype xsi:type="UIntegerT" bitLength="2"/>
    <Name textId="TN_Enum2"/>
  </RecordItem>
</Datatype>

```

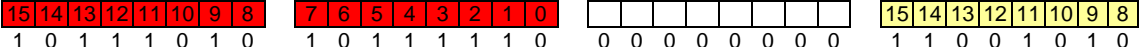
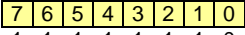
RecordItem	Subindex	Datatype	bitLength	bitOffset	Value
1	1	UIntegerT	4	0	0xF
2	2	BooleanT	—	4	false
3	3	BooleanT	—	5	true
4	4	UIntegerT	2	6	0x3

1195 Octet 0
 7 6 5 4 3 2 1 0

 1 1 1 0 1 1 1 1
 1196 0xEF
 1197

1198 **With a gap** (reserved area for future extension)
 1199

```
1200 <Datatype xsi:type="RecordT" bitLength="40">
1201   <Name textId="TN_Gap"/>
1202   <RecordItem subindex="1" bitOffset="24">
1203     <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
1204     <Name textId="TN_Value1"/>
1205   </RecordItem>
1206   <RecordItem subindex="3" bitOffset="0">
1207     <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
1208     <Name textId="TN_Value2"/>
1209   </RecordItem>
1210 </Datatype>
```

RecordItem	Subindex	Datentyp	bitLength	bitOffset	Value
1	1	UIntegerT	16	24	0xBABE
2	3	UIntegerT	16	0	0xCAFE

1212 Octet 0 Octet 1 Octet 2 Octet 3
 1213 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8

 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0
 1214 0xBA 0xBE 0x00 0xCA
 1215
 1216 Octet 4
 7 6 5 4 3 2 1 0

 1 1 1 1 1 1 1 0
 1217 0xFE
 1218

1219 **Previous example, extended with two record items**
 1220

```
1221 <Datatype xsi:type="RecordT" bitLength="40">
1222   <Name textId="TN_GapFilled"/>
1223   <RecordItem subindex="1" bitOffset="24">
1224     <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
1225     <Name textId="TN_Value1"/>
1226   </RecordItem>
1227   <RecordItem subindex="2" bitOffset="16">
1228     <SimpleDatatype xsi:type="UIntegerT" bitLength="4"/>
1229     <Name textId="TN_Enum"/>
1230   </RecordItem>
1231   <RecordItem subindex="3" bitOffset="0">
1232     <SimpleDatatype xsi:type="UIntegerT" bitLength="16"/>
1233     <Name textId="TN_Value2"/>
1234   </RecordItem>
1235   <RecordItem subindex="4" bitOffset="20">
1236     <SimpleDatatype xsi:type="BooleanT"/>
1237     <Name textId="TN_Switch"/>
1238   </RecordItem>
1239 </Datatype>
```

1240

RecordItem	Subindex	Datentyp	bitLength	bitOffset	Value
1	1	UIntegerT	16	24	0xBABE
2	2	UIntegerT	4	16	0xB
3	3	UIntegerT	16	0	0xCAFE
4	4	BooleanT	–	20	false

1241

1242	Octet 0	Octet 1	Octet 2	Octet 3
	39 38 37 36 35 34 33 32	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8
	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	0 0 0 0 1 0 1 1	15 14 13 12 11 10 9 8
	1 0 1 1 1 0 1 0	1 0 1 1 1 1 1 0	0 0 0 0 1 0 1 1	1 1 0 0 1 0 1 0
1243	0xBA	0xBE	0x0B	0xCA

1244

1245	Octet 4
	7 6 5 4 3 2 1 0
	7 6 5 4 3 2 1 0
	1 1 1 1 1 1 1 0
1246	0xFE

1247

Two signal bits with reserved space

1248

1249

```

1250 <Datatype xsi:type="RecordT" bitLength="12">
1251   <Name textId="TN_ProcessData"/>
1252   <RecordItem subindex="1" bitOffset="0">
1253     <SimpleDatatype xsi:type="BooleanT"/>
1254     <Name textId="TN_Signal2"/>
1255   </RecordItem>
1256   <RecordItem subindex="2" bitOffset="1">
1257     <SimpleDatatype xsi:type="BooleanT"/>
1258     <Name textId="TN_Signal1"/>
1259   </RecordItem>
1260 </Datatype>

```

1261

RecordItem	Subindex	Datentyp	bitLength	bitOffset	Value
1	1	BooleanT	–	0	false
2	2	BooleanT	–	1	true

1262

1263	Octet 0	Octet 1
	14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
	0 0 0 0 0 0 0	0 0 0 0 0 0 1 0
1264	0x00	0x02

7.5.3.3 Process data union data types

1266 The types ProcessDataInUnionT and ProcessDataOutUnionT are restricted to the description
 1267 of the process data standard variables (Index 40 and 41) in IODD-StandardDefinitions1.1.xml
 1268 and thus are not allowed in a normal IODD. The IO-Link Tool shall take the data type of the
 1269 appropriate ProcessDataIn / ProcessDataOut element. If more than one ProcessDataIn /
 1270 ProcessDataOut element is given, it is necessary to select the currently valid element by
 1271 evaluating the Condition elements.

7.5.3.3.1 ProcessDataInUnionT

1272 A ProcessDataInUnionT corresponds to the data type used in ProcessDataCollection/
 1273 ProcessData/ProcessDataIn.
 1274

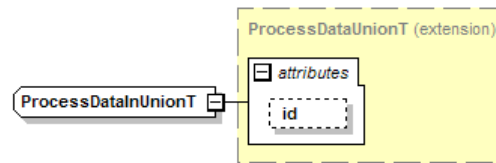


Figure 23 – ProcessDataInUnionT

7.5.3.3.2 ProcessDataOutUnionT

A ProcessDataOutUnionT corresponds to the data type used in ProcessDataCollection/ProcessData/ProcessDataOut.

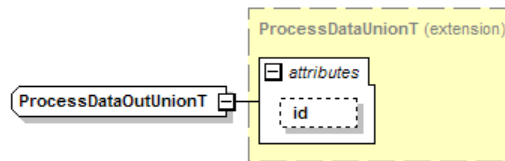


Figure 24 – ProcessDataOutUnionT

7.5.4 Variable collection

All parameters of the device are included here. Standard parameters are defined in IODD-StandardDefinitions1.1.xml and are referenced by StdVariableRef. DirectParameterOverlay allows defining a Record which is being layed over the DirectParameterPage 2 (DirectParameters 16 – 31). All other device-specific variables are named under 'Variable'.

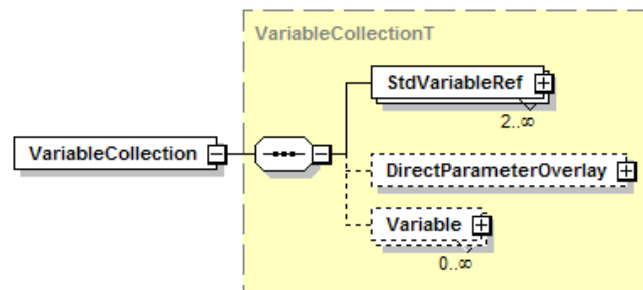


Figure 25 – VariableCollection element

All the variables that the device supports (i.e. the referenced standard variables, the direct parameter overlay and the vendor specific variables) shall have unique Names **within each** supported language.

7.5.4.1 StdVariableRef

Here it is described, which of the standard variables are used. They are referenced here by an explicit key. Because direct parameter page 1 is mandatory, the variable V_DirectParameters_1 shall always be referenced.

It is highly recommended not to use V_DirectParameters_2 and ISDU in the same Device. Use the V_DirectParameter_2 only if your Device does not support ISDUs at all.

V_DirectParameters_2 and DirectParameterOverlay may only be present both or none. If V_DirectParameters_2 and DirectParameterOverlay are present, V_DirectParameters_2 shall not be referenced and DirectParameterOverlay shall be referenced in menu.

All standard ISDU variables marked with the attribute mandatory="true" in the IODD-StandardDefinitions1.1.xml shall be referenced if the device supports ISDU access. The optional standard variables V_ProcessDataInput and V_ProcessDataOutput shall only be

1305 referenced if there is at least one ProcessDataIn / ProcessDataOut element in the
1306 ProcessDataCollection.

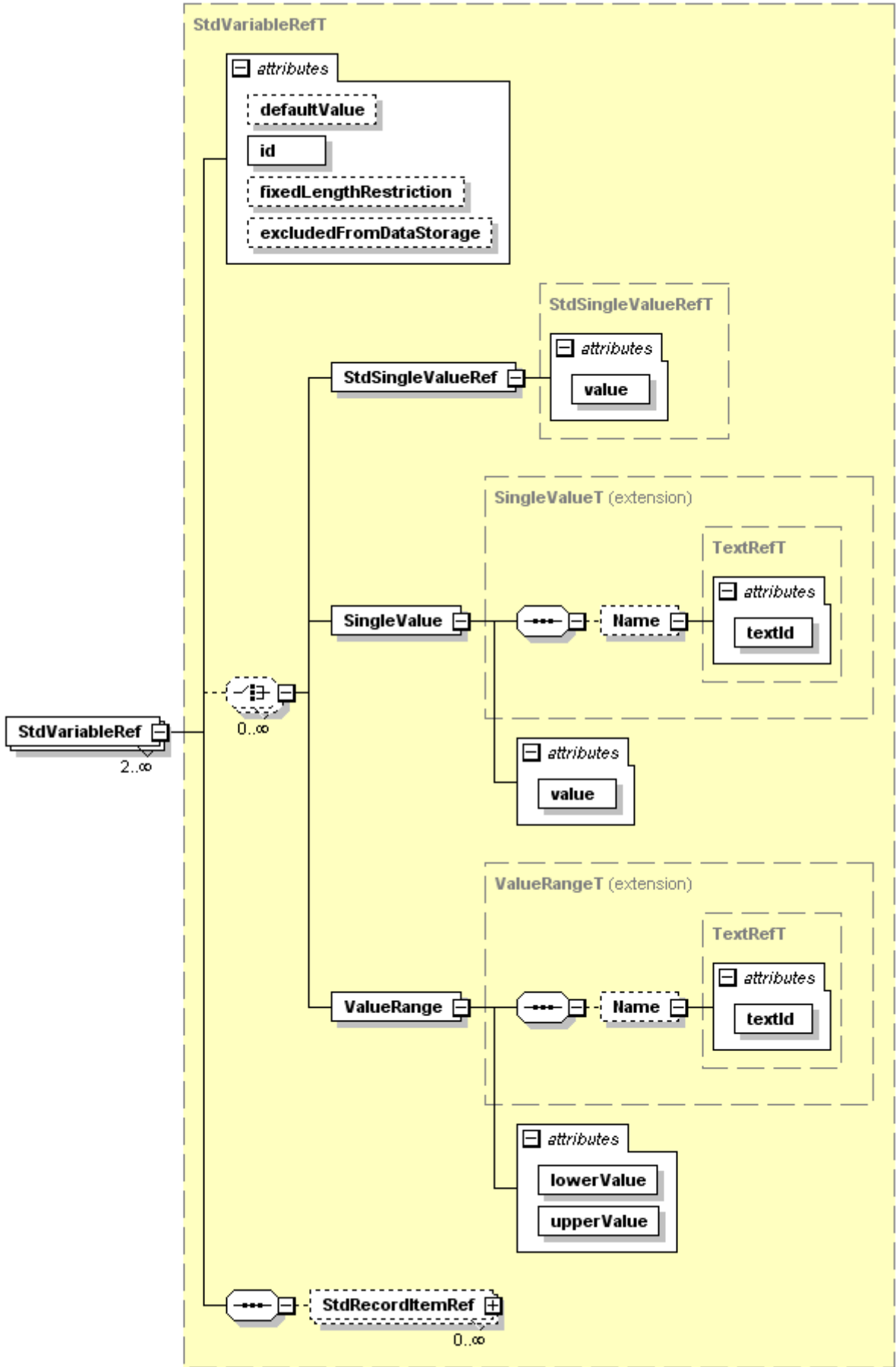


Figure 26 – StdVariableRef element

1309 id (m, IdT)

1310 This id is special since it can be both starting and end point of a referencing process. As end
1311 point of the referencing process, it contains the key of those variables within the IODD. As
1312 starting point, it references to a standard variable.

1313 defaultValue (o, anySimpleType)

1314 The defaultValue shall conform to the data type of the standard variable. Offline default value;
1315 it always refers to the complete variable. If the variable is a record, use StdRecordItemRef
1316 element(s) to specify default values for individual record items. On a variable of type array, the
1317 specified defaultValue shall be applied to all array members. For references to
1318 V_ProcessDataInput or V_ProcessDataOutput this attribute shall not be specified.

1319 For references to V_ProductID this attribute shall not be specified if more than one
1320 DeviceVariant is defined. If there is only one DeviceVariant, then the value of
1321 DeviceVariant/@productId shall be used as default value of V_ProductID.

1322 If more than one device variant is available, the defaultValue of V_ProductName should not be
1323 specified. Alternatively a substitute text covering all Device variants is allowed to be used as
1324 defaultValue. A specific name of any of the variants shall not be used.

1325 fixedLengthRestriction (o, SubindexT)

1326 Only applicable to standard variables of type string, octet string or array. Standard variables of
1327 types string or octet string have a 'fixedLength' attribute describing the maximum length that
1328 the IO-Link standard allows. A Device may implement standard variables with (octet) string
1329 shorter than what the IO-Link standard allows.

1330 Standard Variables of type array have a 'count' attribute describing the maximum size that the
1331 IO-Link standard allows. A Device may implement standard variables with arrays shorter than
1332 what the IO-Link standard allows.

1333 'fixedLengthRestriction' shall be less or equal to the 'fixedLength' (on standard variables of type
1334 string or octetstring) or 'count' (on standard variables of type array). If 'fixedLengthRestriction'
1335 is used with V_ApplicationSpecificTag, it shall be greater or equal to 16.

1336 excludedFromDataStorage (o, boolean)

1337 If set to "true", indicates that the contents of the standard variable are not stored with the data
1338 storage mechanism. This attribute may only be set to "true" for standard variables with
1339 accessRights = "rw". Within StdVariableRefs, excludedFromDataStorage can only be applied
1340 to V_ApplicationSpecificTag, V_DeviceAccessLocks and V_OffsetTime. The default is "false".

1341

1342 Allowed values:

1343 Only applicable to the standard variable V_SystemCommand and V_OffsetTime.

1344 StdSingleValueRef (o)

1345 Specifies a single supported standard value. The 'value' attribute must match the 'value'
1346 attribute of a SingleValue defined at the standard variable.

1347 SingleValue (o)

1348 Specifies a single supported vendor-specific value with an optional name.

1349 ValueRange (o)

1350 Specifies a supported vendor-specific value range with an optional name.

1351 In addition to the above, the following rules shall apply for referencing for standard variables
1352 with StdVariableRef or StdRecordItemRef:

- 1353
- 1354 • When neither SingleValue nor ValueRange nor StdSingleValueRef elements are given,
1355 the standard variable's value range as defined in IODD-StandardDefinitions1.1.xml is
taken. This rule is not valid for V_SystemCommand.
 - 1356 • When SingleValue(s) or ValueRange(s) or StdSingleValueRef(s) are given, only these
1357 values are allowed.

- SingleValues and ValueRanges shall not overlap with SingleValues at the standard variable, no matter whether these are referenced by StdSingleValueRef or not (i.e. standard values can't be redefined in a vendor specific way).

StdRecordItemRef (o)

Used to specify additional information for RecordItems of standard variables. At least one of the optional attributes and elements shall be present. For StdVariableRef id="V_DeviceAccessLocks", StdRecordItemRef is only allowed for those subindices which refer to an access lock that is supported, i.e. where the respective attribute in Features/SupportedAccessLocks is set to "true".

With V_DeviceAccessLocks StdRecordItemRef is only needed if you want to specify a defaultValue for a specific lock. It is not suited to indicate which of the access locks are supported.

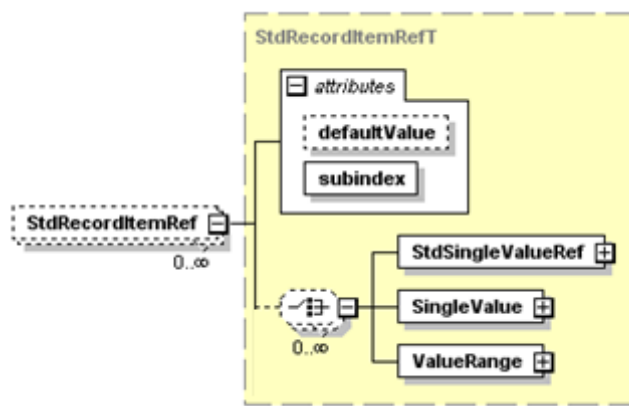


Figure 27 – StdRecordItemRef element

subindex (m, SubindexT)

Used for addressing the record item within the record. The subindex shall be unique within the StdRecordItemRefs of a StdVariableRef.

defaultValue (o, anySimpleType)

The defaultValue shall conform to the data type of the RecordItem. Offline default value.

Specifying allowed values for a RecordItem using StdSingleValueRef, SingleValue and / or ValueRange is only applicable to the standard variable V_DirectParameters_1, subindex 16 (system command for devices without ISDU support). The meaning and the rules regarding these elements shall be the same as with the StdVariableRef element shown above.

7.5.4.2 DirectParameterOverlay

This element corresponds to the device-specific data within the DirectParameter page. If the DirectParameterOverlay is used, TextRedefines should also be added to provide names for each used DirectParameter octet (see chapter 8).

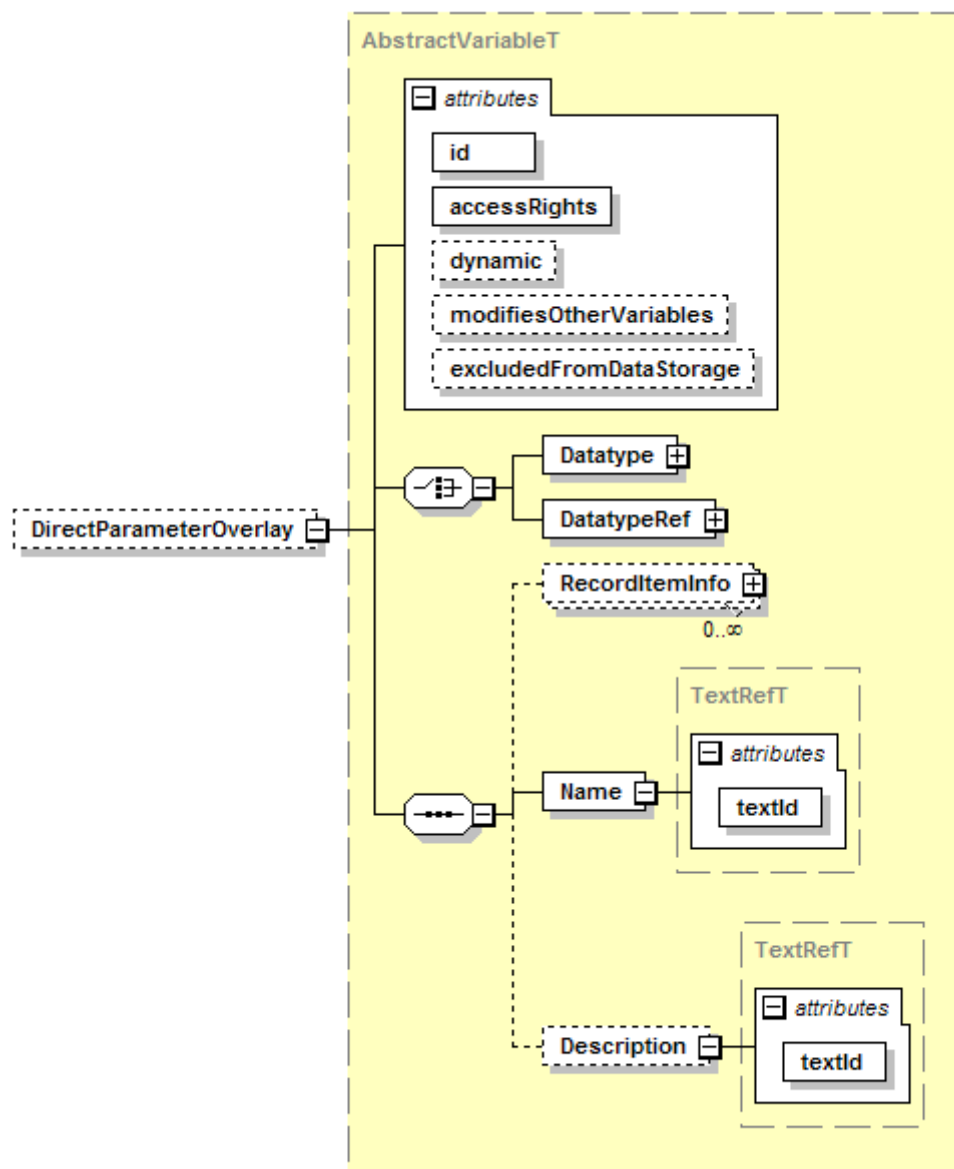


Figure 28 – DirectParameterOverlay element

id (m, IdT)

As the end point of a referencing process, it contains the key of the variable within the IODD. The id of any standard variable shall not be used as id for the DirectParameterOverlay, even if the standard variable is not referenced from this IODD.

accessRights (m, AccessRightsT)

“ro” read-only,
“wo”, write-only,
“rw”, read-write

dynamic (o, boolean)

Serves as information, whether the variable is autonomously changed by the device. This attribute may only be set to “true” for DirectParameterOverlay with accessRights = “rw” or “ro”. The default is “false”.

modifiesOtherVariables (o, boolean)

It is highly recommended to omit this attribute or set it to “false”.

excludedFromDataStorage (o, boolean)

It is highly recommended to omit this attribute or set it to “false”.

1405 Datatype (c)

1406 Directly given data type (see Note below)

1407 DatatypeRef (c)

1408 Reference to a data type that was defined in the DatatypeCollection (see Note below)

1409 RecordItemInfo (o)

1410 Contains additional information for record items. See 8.

1411 Name (m)

1412 textId (m, RefT)

1413 Contains the name of the variable

1414 Description (o)

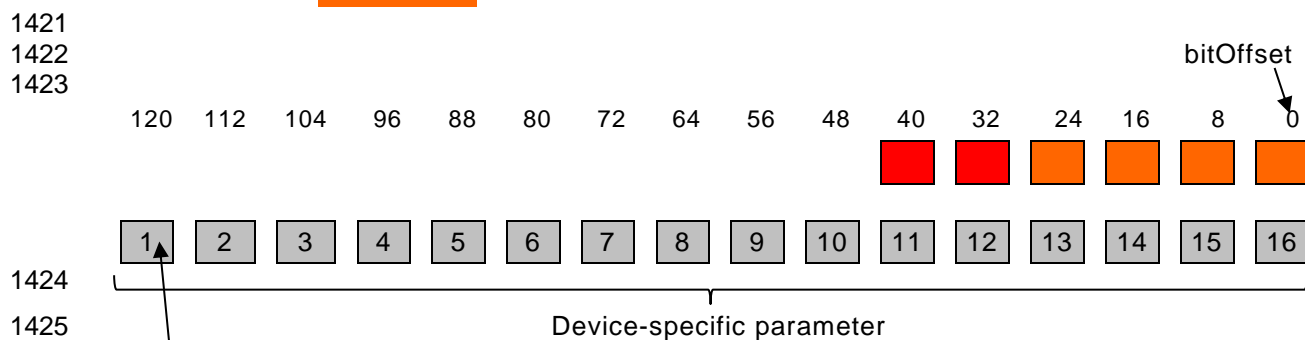
1415 textId (m, RefT)

1416 Contains a description of the variable (e.g. information text, help, etc.)

1417 Note: The data type shall be a record with a minimum length of 1 bit and a maximum length of
1418 128 bits. The last octet of this record is mapped to the last octet of the direct parameter page
1419 2.

1420 Example

RecordItem	Subindex	Datatype	bitLength	bitOffset
1	1	UIntegerT	16	32
2	2	UIntegerT	32	0



1424

1425

1426 Subindex in Index 1 (V_DirectParameters_2)

1427 **Figure 29 – Direct parameter overlay**

1428 Note: The communication of direct parameters is octet oriented.

- 1429 • For record items, which cross an octet boundary the consistency cannot be
1430 guaranteed.
- 1431 • If an octet contains more than one record item, the subindex access will influence all
1432 contained record items or parts.
- 1433 • For record items, which cross an octet boundary, the device cannot rely on the order of
1434 the single accesses. This means, the device shall tolerate intermediate values that may
1435 exceed the allowed value range.

1436 Recommendation: Use DirectParameterOverlay only for devices that do not support ISDU
1437 access.

1438 7.5.4.3 Variable

1439 Contains the description of a device parameter.

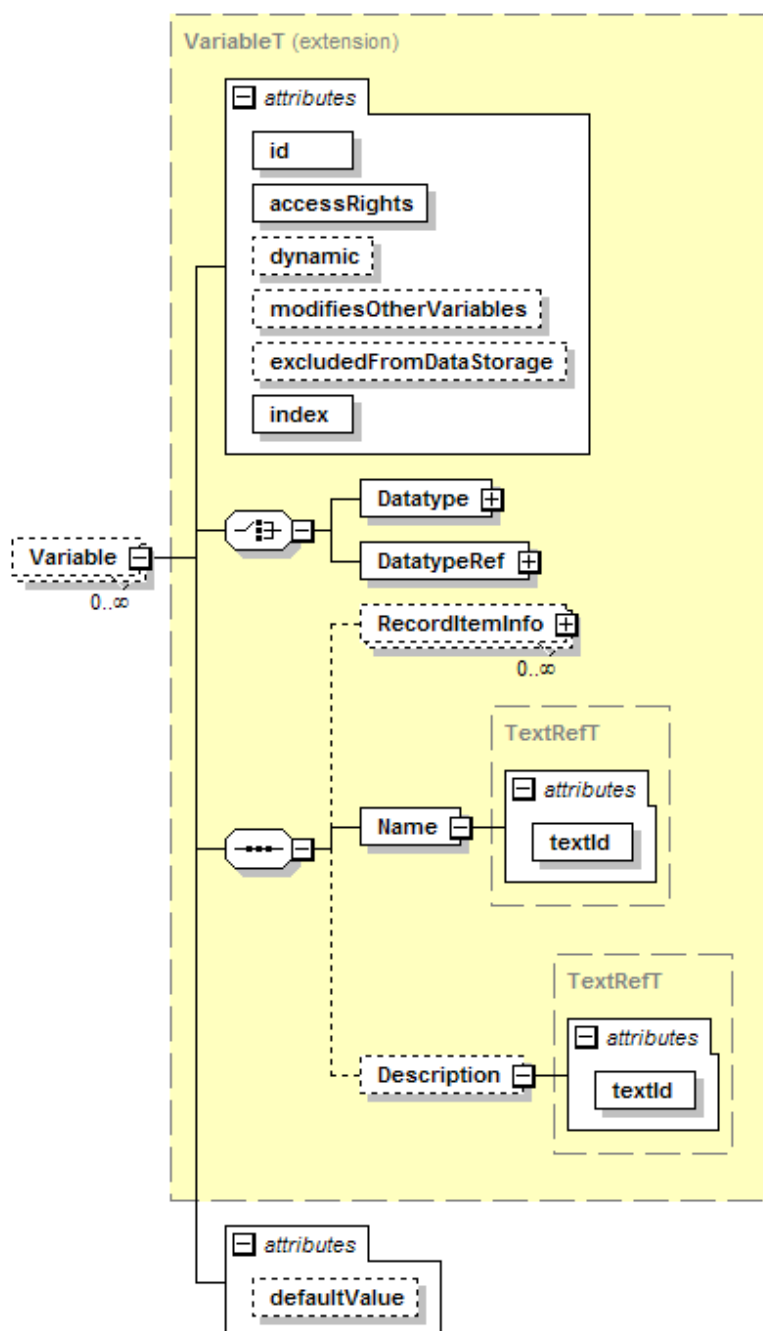


Figure 30 – Variable element

id (m, IdT)

As the end point of a referencing process, it contains the key of the variable within the IODD. The id of any standard variable shall not be used as id for the Variable, even if the standard variable is not referenced from this IODD.

accessRights (m, AccessRightsT)

“ro” read-only,
 “wo”, write-only,
 “rw”, read-write

For records and arrays the attribute accessRights='wo' is not permitted.

dynamic (o, boolean)

Serves as information, whether the variable is autonomously changed by the device. **This attribute may only be set to “true” for variables with accessRights = “rw” or “ro”.** The default is “false”.

modifiesOtherVariables (o, boolean)

If set to “true”, indicates that a write access to this variable (or to any of its subindices) may change the value of other variables. IO-Link Tools should re-load the variables of the device after a write access to this variable. **This attribute may only be set to “true” for variables with accessRights = “rw” or “wo”.** The default is “false”.

excludedFromDataStorage (o, boolean)

If set to “true”, indicates that this variable’s contents are not stored with the data storage mechanism. **This attribute may only be set to “true” for variables with accessRights = “rw”.** The default is “false”.

index (m, unsignedShort)

Index for the addressing of a variable. Shall be in the range for vendor specific indices or profile specific indices that are not already described as standard variables in IODD-StandardDefinitions1.1.xml.

defaultValue (o, anySimpleType)

The defaultValue shall conform to the data type of the variable. Offline default value; it always refers to the complete variable. If the variable is a record, use RecordItemInfo element(s) to specify default values for individual record items. On a variable of type array, the specified defaultValue shall be applied to all array members.

Datatype (c)

Directly given data type

DatatypeRef (c)

Reference to a data type that was defined in the DatatypeCollection

RecordItemInfo (o)

Only applicable if the variable is of type record. Contains additional information for record items. See 8.

Name (m)**textId (m, RefT)**

Contains the name of the variable

Description (o)**textId (m, RefT)**

Contains a description of the variable (e.g. information text, help, etc.)

7.5.4.4 RecordItemInfo

For variables of type RecordT, contains optional attributes for a RecordItem addressed by the subindex. At least one of the optional attributes shall be present.

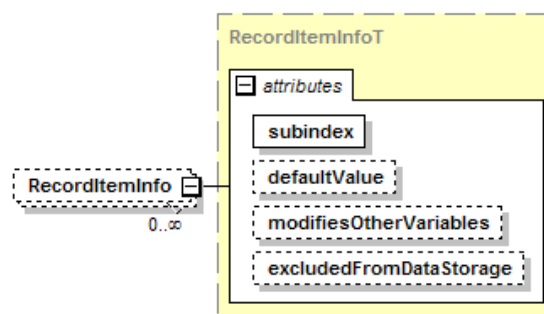


Figure 31 – RecordItemInfo element

subindex (m, SubindexT)

Used for addressing the record item within the record.

defaultValue (o, anySimpleType)

The defaultValue shall conform to the data type of the record item. Contains the default value for the RecordItem.

modifiesOtherVariables (o, boolean)

If set to "true", indicates that a write access to this subindex may change the value of other variables. IO-Link Tools should re-load the variables of the device after a write access to this subindex. **This attribute may only be set to "true" for record items with accessRights = "rw" or "wo".** This attribute shall not be specified both on the DirectParameterOverlay/Variable element and a subordinated RecordItemInfo element. The default is "false".

excludedFromDataStorage (o, boolean)

If set to "true", indicates that this subindex's contents are not stored with the data storage mechanism. **This attribute may only be set to "true" for record items with accessRights = "rw".** This attribute shall not be specified both on the DirectParameterOverlay/Variable element and a subordinated RecordItemInfo element. The default is "false".

7.5.5 Process data collection

Contains all process data of the device

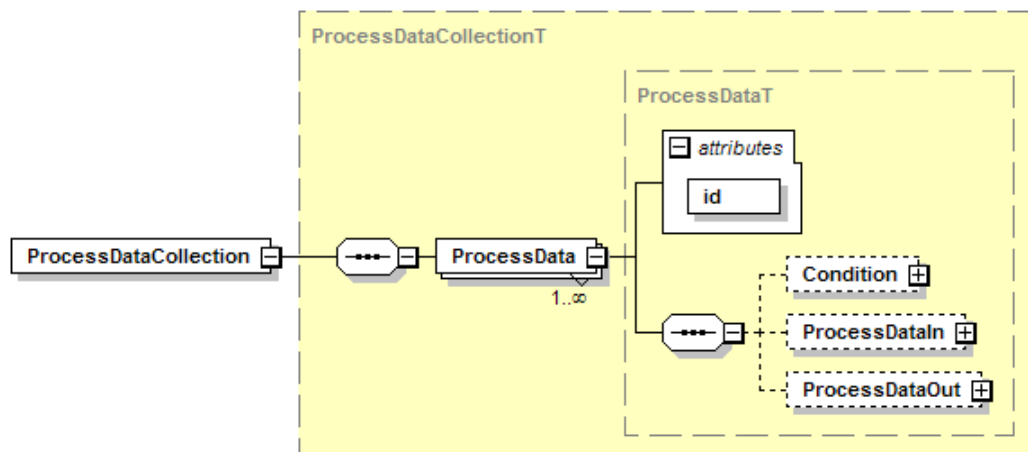


Figure 32 – ProcessDataCollection element

The element ProcessData may occur multiple times in the collection. If ProcessData occurs more than once,

- all the ProcessData elements shall contain a Condition element
- for each user role there shall be a menu (could be the same) that contains at least a read-only reference to the Variable / RecordItem used in the ProcessData/Condition elements
- the attributes 'variableId' and 'subindex' in the Condition elements shall be the same (there shall only be exactly one variable / record item used for the switching of the process data)
- the attribute 'value' in the Condition elements shall be unique within the ProcessData elements
- the attribute 'bitLength' in the ProcessDataIn elements shall be the same for all ProcessData
- the attribute 'bitLength' in the ProcessDataOut elements shall be the same for all ProcessData

- the variable / record item referenced in the Condition elements selects the currently valid ProcessData element when its value matches the 'value' attribute of the Condition element

The attribute 'id' shall be unique within all the elements ProcessData, ProcessDataIn and ProcessDataOut.

ProcessData (m)

id (m, IdT)

Explicit id of the ProcessData

Condition (o)

Serves to switch between different ProcessData.

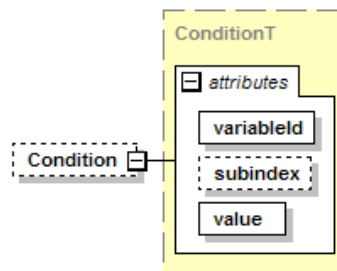


Figure 33 – Condition element

variableId (m, RefT)

References a variable. The variable shall be of data type BooleanT, IntegerT, UIntegerT or RecordT. The variable shall have a default value if it is not of type RecordT.

subindex (c, SubindexT)

This attribute shall be given if and only if the referenced variable is of type RecordT. Used for addressing the record item within the record. The record item shall be of data type BooleanT, IntegerT or UIntegerT and shall have a default value.

value (m, unsignedByte)

Must be a valid value for the variable / record item. This attribute can only hold values 0..255, thus limiting the possible IntegerT and UIntegerT values. Also, BooleanT condition values shall be entered as "0" for "false" and "1" for "true".

ProcessDataIn (o)

Description of the input process data

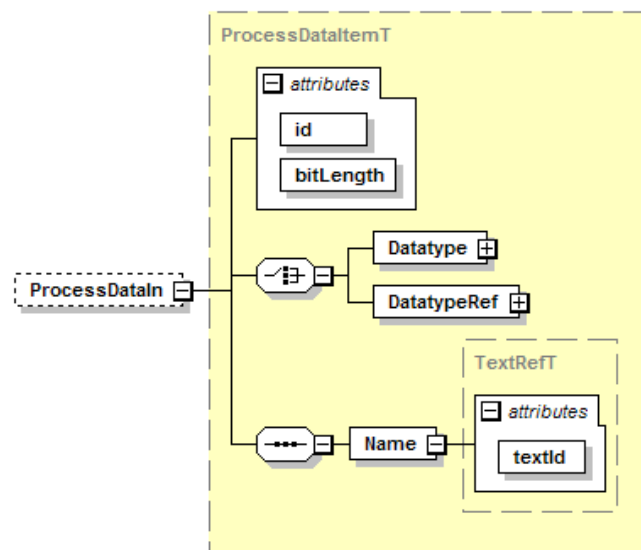


Figure 34 – ProcessDataIn element

1549

1550

1551 **id (m, IdT)**

1552 Explicit id of the ProcessDataIn description

1553 **bitLength (m, BitCountT)**

1554 Length of the input process data (in bits). The allowed value range is 1..256.

1555 It shall represent the underlying ProcessDataIn data type in a bit granular manner. For record
1556 data types this bitLength shall equal the 'bitLength' attribute of the record.

1557 The value of the DirectParameterPage 1, subindex 5 (Process Data In), shall be calculated from
1558 the 'bitLength' attribute value by the following formula:

1559 If bitLength <= 16 then

1560 ProcessDataIn = bitLength

1561 Else

1562 ProcessDataIn = bitLength rounded up to the next multiple of 8

1563 End If

1564 **Name (m)**

1565 **textId (m, RefT)**

1566 Name specification of the input process data

1567 **ProcessDataOut (o)**

1568 Description of the output process data

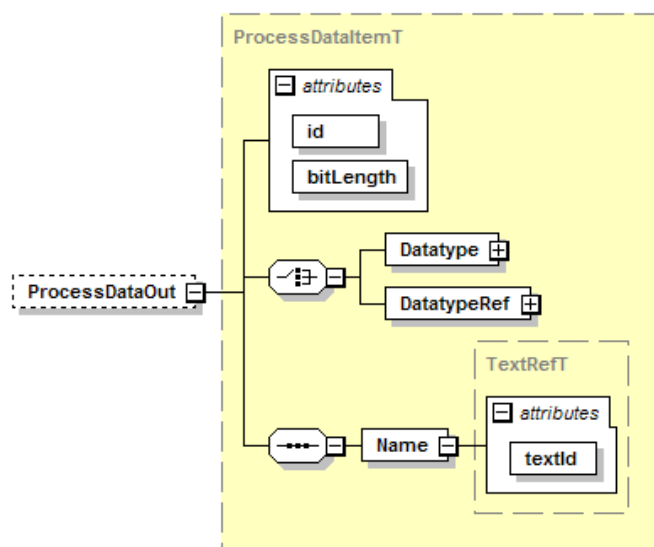


Figure 35 – ProcessDataOut element

1569

1570

1571 **id (m)**

1572 Explicit id of the ProcessDataOut description

1573 **bitLength (m, BitCountT)**

1574 Length of the output process data (in bits). The allowed value range is 1..256.

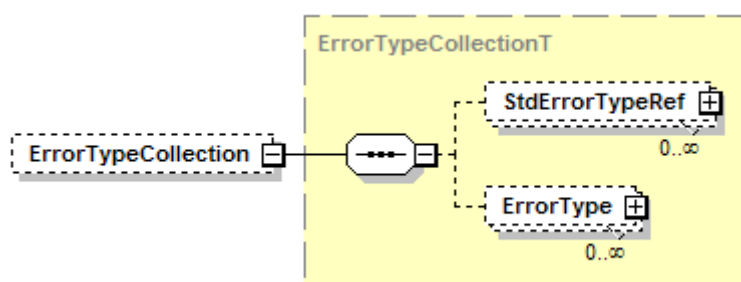
1575 The description of the 'bitLength' attribute at the ProcessDataIn element above also applies
 1576 here, but for Process Data Out (DirectParameterPage 1, subindex 6).

1577 **Name (m)**1578 **textId (m, RefT)**

1579 Name specification of the output process data

1580 **7.5.6 Error type collection**

1581 All error types that the device may return are collected here. There are system defined error
 1582 types (code=128) and vendor specific error types (code=129), see **IO-Link Interface and System**
 1583 **Specification Version 1.1.3**, annex C. The system defined error types are described in IODD-
 1584 StandardDefinitions1.1.xml and referenced by 'StdErrorTypeRef', while the vendor specific
 1585 error types are specified with 'ErrorType'.



1586

1587

Figure 36 – ErrorTypeCollection element

1588 **StdErrorTypeRef (o)**

1589 Standard error types are referenced by their 'additionalCode'.

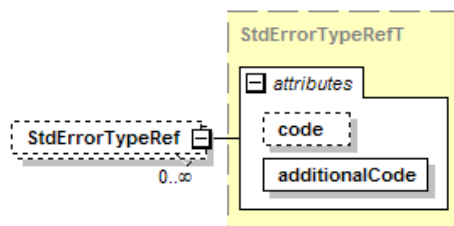


Figure 37 – StdErrorTypeRef element

code (o, unsignedByte)

Fixed to 128 by the IO-Link specification.

additionalCode (m, unsignedByte)

The additional code. Shall be unique within the 'StdErrorTypeRef' elements, and shall reference one of the error types defined in IODD-StandardDefinitions1.1.xml.

ErrorType (o)

Vendor specific error type, identified by its 'additionalCode'.

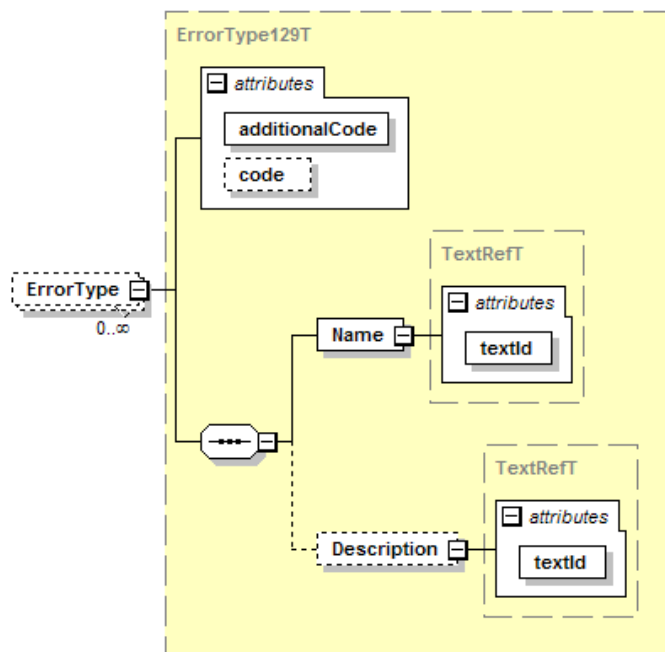


Figure 38 – ErrorType element

code (o, unsignedByte)

Fixed to 129 by the IO-Link specification.

additionalCode (m, unsignedByte)

The additional code. Shall be unique within the 'ErrorType' elements.

Name (m)

textId (m, RefT)

Use this text for the error message.

Description (o)

textId (m, RefT)

Use this text for the possible cause of the error and the remedy.

Error Types shall be handled by IO-Link Tools.

Since a device can respond to an ISDU with an ErrorType, IO-Link Tools shall list all incoming ErrorTypes during an up- or download. Up- or downloads shall not be interrupted. If a tool

supports device replication and the device responds with a single ErrorType, this replication shall be interrupted.

7.5.7 Event collection

All events that the device may return are collected here. There are system defined events and vendor specific events, see [IO-Link Interface and System Specification Version 1.1.3](#), annex D. The system defined events are described in IODD-StandardDefinitions1.1.xml and referenced by 'StdEventRef', while the vendor specific events are specified with 'Event'.

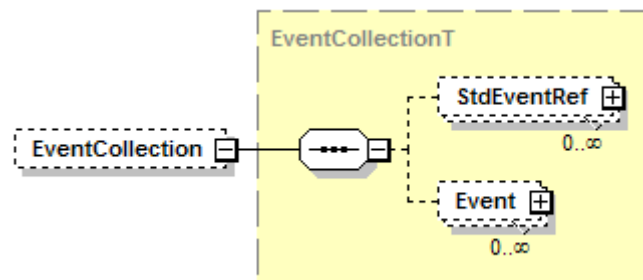


Figure 39 – EventCollection element

StdEventRef (o)

Indicates that the device may return the standard event identified by the 'code'.

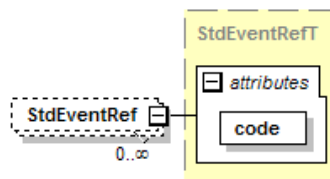


Figure 40 – StdEventRef element

code (m, unsignedShort)

The event code that identifies the standard event described in IODD-StandardDefinitions1.1.xml.

Event (o)

Describes a vendor specific event.

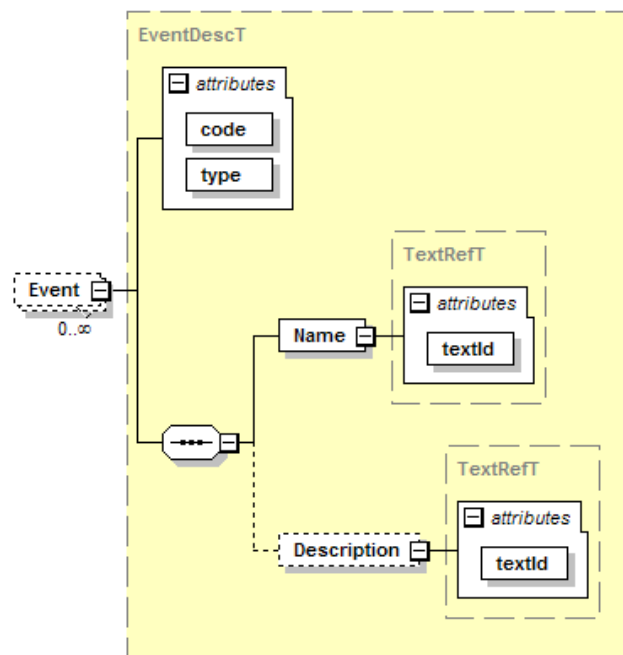


Figure 41 – Event element

code (m, unsignedShort)

The event code that identifies the vendor specific event. Shall be in the range for vendor specific or profile specific event codes.

type (m, string)

The TYPE part of the EventQualifier (see [IO-Link Interface and System Specification Version 1.1.3](#), chapter A.6.4). One of “Notification”, “Warning” or “Error”.

Note that the MODE depends on the TYPE, SOURCE is always 0 (device application) and INSTANCE is always 4 (application).

Name (m)**textId (m, RefT)**

Use this text for the event message.

Description (o)**textId (m, RefT)**

Use this text for the possible cause of the event and the remedy.

7.5.8 User interface

Contains the menus of the device

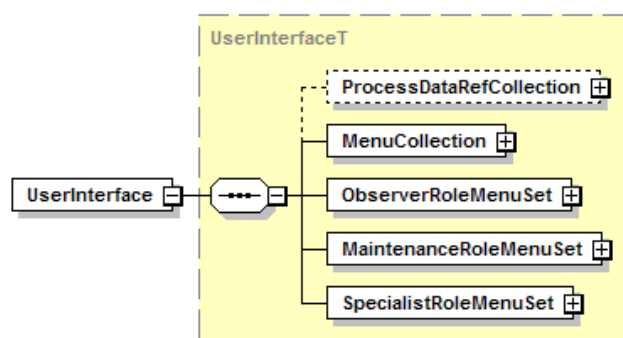


Figure 42 – UserInterface element

7.5.8.1 ProcessDataRef collection (o)

Some IO-Link masters support read access to the process data **independently** of the device. Process data read **this** way are shown in a separate menu by the IO-Link Tools for these masters. This collection allows defining how the process data read via the master are to be displayed.

On the other hand, a device may support the optional standard variables V_ProcessDataInput / V_ProcessDataOutput for access to the process data. Usually these variables are then referenced from the Observation or Diagnosis menu. The 'VariableRef' or 'RecordItemRef' controls how the process data read from the device are to be displayed.

Even if the device supports V_ProcessDataInput / V_ProcessDataOutput, when the 'VariableRef' or 'RecordItemRef' uses attributes to control how the process data is displayed, the ProcessDataRefCollection should be entered using the same attribute values.

If an IO-Link Tool wants to display process data and the IODD does not contain a corresponding ProcessDataRef for it, the tool shall display the process data just according to its data type. The tool shall not try to find the display format by walking the menus searching for V_ProcessDataInput / V_ProcessDataOutput references.

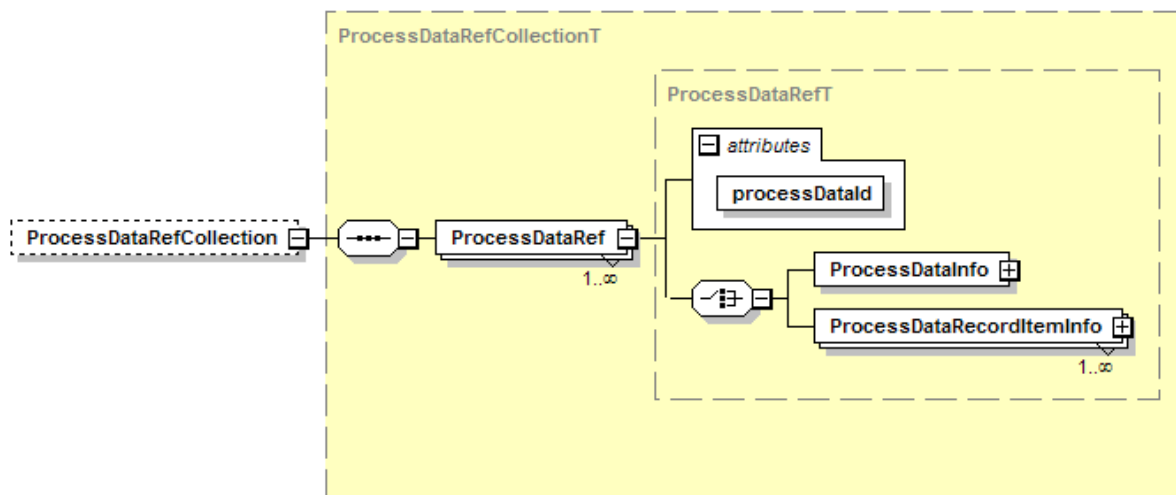


Figure 43 – ProcessDataRefCollection element

processDataId (m, RefT)

Refers to DeviceFunction/ProcessData/ProcessDataIn or ProcessDataOut.

ProcessDataInfo (c)

Corresponds to the 'VariableRef' element of a menu except for 'accessRightRestriction' and 'Button' which are not applicable (see 9).

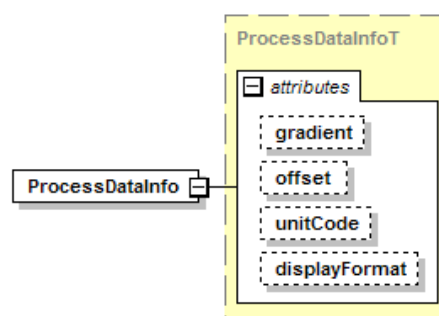
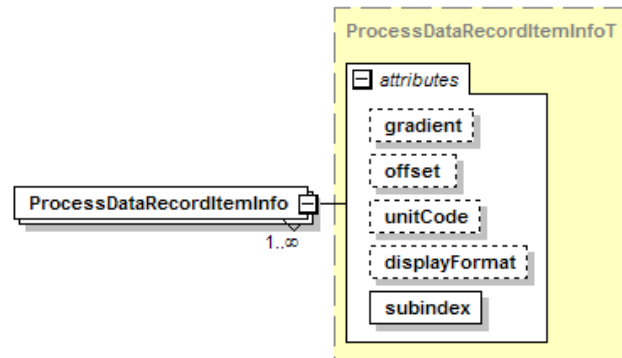


Figure 44 – ProcessDataInfo element

Use this for process data which is not of type record.

1679 **ProcessDataRecordItemInfo (c)**

1680 Corresponds to the 'RecordItemRef' element of a menu except for 'accessRightRestriction' and
 1681 'Button' which are not applicable (see 9).



1682
 1683 **Figure 45 – ProcessDataRecordItemInfo element**

1684 Use this for process data which is of type record.

1685 **7.5.8.2 MenuSets (m)**1686 **User Roles**

1687 A **user interface shall** be divided into three user roles. It is up to the vendor how the roles are
 1688 organized. The IO-Link Tool must assign the entered UserLevel to the respective menu. At most
 1689 three menu levels below the role assignment are acceptable.

1690 Example:

1691 ObservationRoleMenuSet

1692 → IdentificationMenu

1693 → Menu1

1694 → MenuRef1

1695 → Menu2

1696 → MenuRef1

1697 MaintenanceRoleMenuSet

1698 → ObservationMenu

1699 → MenuX

1700 → MenuRefY

1701 IO-Link Tools shall upload or download only the variables of the current user role. If the tool
 1702 supports a special function to replicate an IO-Link device, this function shall use the variables
 1703 of the specialist role and can be available in all user roles.

1704 **ObserverRoleMenuSet (m)**

1705 This menu is designed for users who may not carry out any modifications on the device.

1706 **The role name in English: "Operator".**

1707 **MaintenanceRoleMenuSet (m)**

1708 This menu is designed for observers and to undertake "uncritical" editing. It is up to the vendor
 1709 to assess that.

1710 **The role name in English: "Maintenance".**

1711 **SpecialistRoleMenuSet (m)**

1712 If the user is logged in as a specialist, he/she has total access to the device. Again, the vendor
 1713 can decide which parameters may be edited.

1714 **The role name in English: "Specialist".**

1715

1716 For each role, there is a set of fixed top-level menus given.

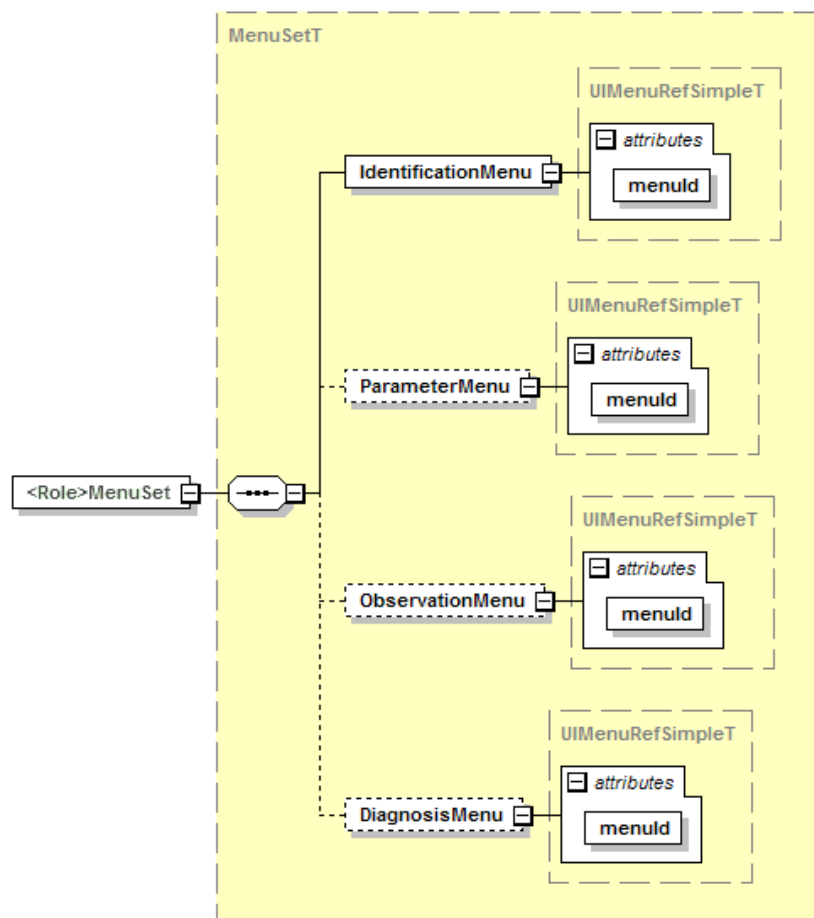


Figure 46 – <Role>MenuSet element

IdentificationMenu (m)

The attribute 'menuld' references a menu from the MenuCollection. This menu should contain variables which serve the identification of the device.

The menu name in English: "Identification".

ParameterMenu (o)

The attribute 'menuld' references a menu from the MenuCollection. This menu should contain variables which serve the parameterization of the device.

The menu name in English: "Parameter".

ObservationMenu (o)

The attribute 'menuld' references a menu from the MenuCollection. This menu should contain variables which serve the observation of the device (process data, dynamic variables, etc.).

The menu name in English: "Observation".

DiagnosisMenu (o)

The attribute 'menuld' references a menu from the MenuCollection. This menu should contain variables which serve the diagnosis of the device (events, etc.).

The menu name in English: "Diagnosis".

The English 'name' definition within this chapter is translated into common languages, and the text templates are delivered within this package. If a tool supports one of those common languages, it shall apply only those translations.

7.5.8.3 Menu collection

The names of top level menus, like IdentificationMenu, ParameterMenu, ObservationMenu or DiagnosisMenu are given from tooling. If a name is specified, it shall be ignored by tooling.

In underlying menus, a menu name shall be given by IODD.

MenuCollection (m)

All menu entries of the device are collected in the MenuCollection. These menu entries may be referenced by different roles (ObserverRole, MaintenanceRole, and SpecialistRole). There shall be no unreferenced Menu elements.

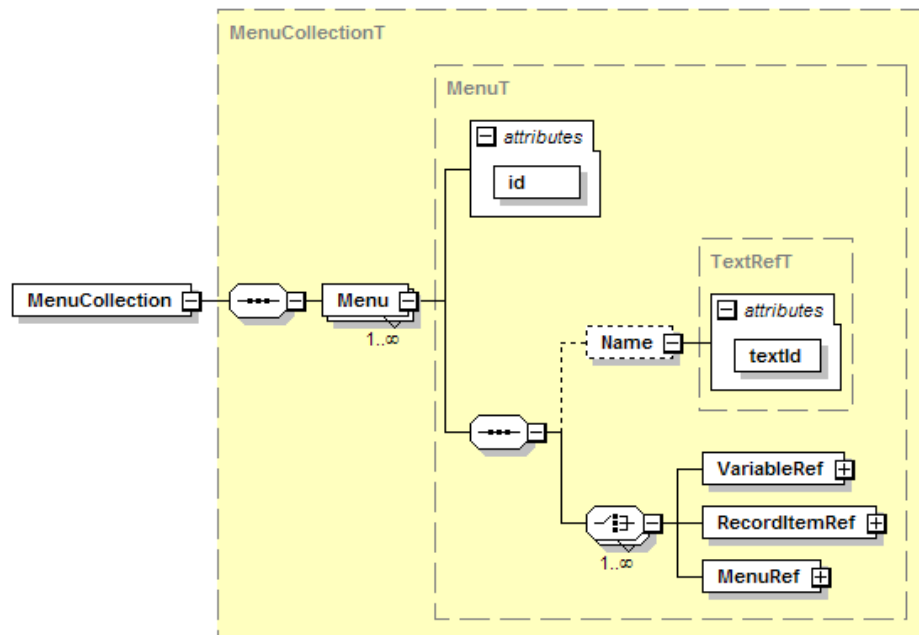


Figure 47 – MenuCollection element

Menu (m)

Variables, RecordItems and other menus may be referenced here.

id (m, IdT)

Explicit id of the menu.

Name (c)

textId (m, RefT)

Name of the menu. Top-level menus (i.e. those referenced from one of the MenuSets) may have a Name element, but it shall be ignored by IO-Link Tools. Instead, hard-coded names shall be used by the tools. Nested menus shall have a Name element which is shown by the tools.

7.5.8.4 VariableRef

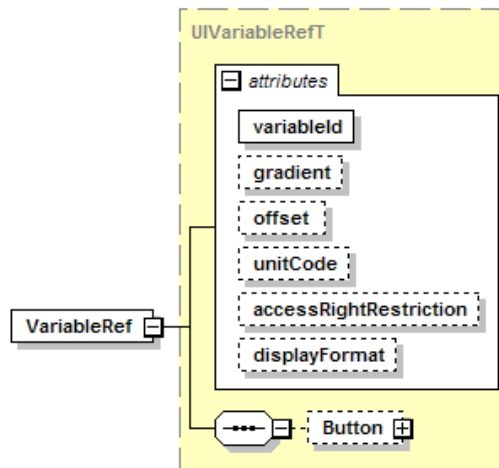


Figure 48 – VariableRef element

variableId (m, RefT)

Referenced variable

Regardless of the type of the referenced Variable or RecordItem, if gradient and / or offset are given, they shall be specified as floating point values.

Displayed value = (value read from the Device x gradient) + offset

When applying gradient and / or offset to convert the Variable or RecordItem value to the displayed value, the value will be **implicitly** converted to a floating point value. Consequently, the only allowed displayFormat on such values shall be "Dec". (The displayFormat "Hex", "Bin", ... does not force a conversion back to the original type of the Variable or RecordItem.)

When applying gradient and / or offset to convert an entered value back to the new value of a Variable or RecordItem, the resulting floating point value will be rounded to the nearest possible value of the type of the Variable or RecordItem.

Single array members can't be referenced with RecordItemRef. If you need to access a single member, you have to define a record instead of an array.

A variable of type array can only be referenced as a whole, i.e. with VariableRef. All the elements and attributes in VariableRef (gradient, offset, unitCode, accessRightRestriction, displayFormat and Button) apply to each of the array members.

gradient (o, decimal)

Gradient of the indicated variables. The value shall not be zero. When offset is specified and gradient is not specified, a value of 1.0 shall be used.

offset (o, decimal)

Zero-offset of the indicated variables. When gradient is specified and offset is not specified, a value of 0.0 shall be used.

unitCode (o, positiveInteger)

Unit code to which the indicated variable refers. For valid unit codes see IODD-StandardUnitDefinitions1.1.xml.

unitCode shall only be used with datatypes UIntegerT, IntegerT and Float32T.

unitCode shall only be used without displayFormat or with displayFormat Dec and Dec.x.

accessRightRestriction (o, AccessRightsT)

For certain UserRoles, the access rights may be limited here.

1795 The attribute `accessRightRestriction` shall not be set to "wo".

1796 **displayFormat (o, string with pattern)**

1797 Specifies how an IO-Link Tool shall display the value in the menu. The values of the attribute
1798 'displayFormat' shall follow the regular expression pattern:

1799 "Bin|Hex|Dec(\.\d)?"

1800 Meaning of the values:

1801 Bin: Binary notation with postfix "b", e.g. 0101 1010 1010 0101b
1802 Hex: Hexadecimal notation with postfix "h", e.g. 5AA5h
1803 Dec: Decimal notation without postfix, e.g. 23205
1804 Dec.2: Decimal notation with given precision (number of digits after the
1805 decimal point) e.g. 23.00

1806 The following table shows the valid combinations of the data type of the referenced Variable /
1807 RecordItem and the displayFormat, gradient and offset. Combinations not listed here shall not
1808 be used.

1809 **Table 2 – Allowed combinations of datatype, displayFormat, gradient and offset**

datatype	allowed displayFormat	gradient and/or offset allowed	IO-Link Tool behaviour
BooleanT	Dec	No	Display as "0" for "false" and "1" for "true".
	<i>default</i>	No	Display as "false" or "true".
UIntegerT	Bin	No	Display as e.g. "0101 1010 1010 0101b". Show 8, 16, 32 or 64 binary digits.
	Hex	No	Display as e.g. "5AA5h". Show 2, 4, 8 or 16 hexadecimal digits.
	Dec	Yes	Without gradient and/or offset: Display as e.g. "23205". Do not show leading zeroes. With gradient and/or offset: See Float32T, displayFormat=Dec
	Dec.x	Yes	Without gradient and/or offset: Display as e.g. "23205.00". Do not show leading zeroes. With gradient and/or offset: See Float32T, displayFormat=Dec.x
	<i>default</i>	Yes	Same as Dec.
IntegerT	Bin	No	Display as e.g. "1111 1011 0010 1110b". Show 8, 16, 32 or 64 binary digits. Show negative values as two's complement.
	Hex	No	Display as e.g. "FB2Eh". Show 2, 4, 8 or 16 hexadecimal digits. Show negative values as two's complement.

	Dec	Yes	Without gradient and/or offset: Display as e.g. "-1234". Do not show leading zeroes. With gradient and/or offset: See Float32T, displayFormat=Dec
	Dec.x	Yes	Without gradient and/or offset: Display as e.g. "-1234.00". Do not show leading zeroes. With gradient and/or offset: See Float32T, displayFormat=Dec.x
	<i>default</i>	Yes	Same as Dec.
Float32T	Dec	Yes	Display digits after the decimal point as needed (up to an implementation-defined maximum).
	Dec.x	Yes	Display exactly "x" digits after the decimal point (also in exponential representation). Rounding shall be done with midpoint rounding away-from-zero (e.g. with "Dec.3" 23.3455 gets rounded to 23.346, and -23.3455 gets rounded to -23.346).
	<i>default</i>	Yes	Same as Dec.
StringT	<i>default</i>	No	Display just the string.
OctetStringT	<i>default</i>	No	Display as e.g. 0x00,0x56,0x78.
TimeT	<i>default</i>	No	Display as yyyy-mm-dd hh:mm:ss.fff where yyyy is the year, mm is the month, dd is the day, hh is the hour, mm is the minute, ss is the second and fff is the milliseconds.
TimeSpanT	<i>default</i>	No	Display as [+][d]hh:mm:ss.fff where d is the days (optional, one or more digits), hh is the hour, mm is the minute, ss is the second and fff is the milliseconds.
ArrayT	Display all array elements. Button is not allowed. The allowed displayFormat, gradient/offset and unitCode is determined by the data type of the array elements.		
RecordT	<i>default</i>	No	Display all RecordItems in the order in which they appear in the Record definition, i.e. with ascending subindices, with their default display format. Button and / or unitCode are not allowed.

1810 Some standard variables have complex types which are not modelled as special types in
1811 IODD because:

- 1812 • The types cannot be used with other variables in a meaningful way.
- 1813 • The types are difficult to describe in XML, increasing the complexity of the IODD.
- 1814 • The types are IO-Link specific, complicating the general use of the IODD.

1815 These standard variables could be displayed as bare numbers the way they are described in
1816 IODD-StandardDefinitions1.1.xml, but it is recommended that IO-Link Tools identify them by
1817 their name or index and display them specially, as described in the following table:

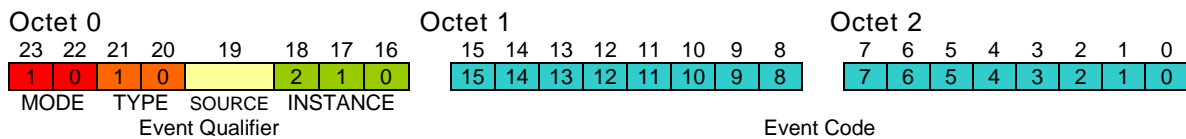
1818 **Table 3 – Standard variables with special display**

Standard Variable / RecordItem	Special display
V_DirectParameters_1 (index 0), MasterCycleTime (subindex 2)	The octet consists of a Time Base in bits 7 to 6 and a Multiplier in bits 5 to 0. The IO-Link Tool shall calculate the time according chapter B.1.3 of the IO-Link Interface and System Specification Version 1.1.3 and display it as a decimal number with the unit milliseconds (ms).
V_DirectParameters_1 (index 0), MinCycleTime (subindex 3)	The octet consists of a Time Base in bits 7 to 6 and a Multiplier in bits 5 to 0. The IO-Link Tool shall calculate the time according chapter B.1.3 of the IO-Link Interface and System Specification Version 1.1.3 and display it as a decimal number with the unit milliseconds (ms).
V_DirectParameters_1 (index 0), M-sequence Capability (subindex 4)	The octet consists of a PREOPERATE M-sequence type in bits 5 to 4, an OPERATE M-sequence type in bits 3 to 1, and ISDU in bit 0. The IO-Link Tool shall decode this into text according to chapter B.1.4 of the IO-Link Interface and System Specification Version 1.1.3 .
V_DirectParameters_1 (index 0), RevisionID (subindex 5)	The octet consists of a MajorRev in bits 7 to 4 and a MinorRev in bits 3 to 0. It shall be displayed as Vx.y, where x is the MajorRev and y is the MinorRev.
V_DirectParameters_1 (index 0), ProcessDataIn (subindex 6)	The octet consists of BYTE in bit 7, SIO in bit 6 and Length in bits 4 to 0. The IO-Link Tool shall display whether SIO is supported and the length in bits / octets according chapter B.1.6 of the IO-Link Interface and System Specification Version 1.1.3 .
V_DirectParameters_1 (index 0), ProcessDataOut (subindex 7)	The octet consists of BYTE in bit 7, SIO in bit 6 and Length in bits 4 to 0. The IO-Link Tool shall display whether SIO is supported and the length in bits / octets according chapter B.1.7 of the IO-Link Interface and System Specification Version 1.1.3 .
V_DetailedDeviceStatus (index 37)	Each array element shall be treated as an event data structure (see IO-Link Interface and System Specification Version 1.1.3 , chapter A.6. It shall be decoded and displayed to text using the EventCollection in the IODD.

V_OffsetTime (index 48)	The octet consists of a Time Base in bits 7 to 6 and a Multiplier in bits 5 to 0. The IO-Link Tool shall calculate the time according chapter B.2.24 of the IO-Link Interface and System Specification Version 1.1.3 and display it as a decimal number with the unit milliseconds (ms).
V_DeviceAccessLocks (index 12)	V_DeviceAccessLocks shall only be referenced via RecordItemRef to apply to the supported subindices, see chapter 7.5.8.5.

1819

1820



1821

Figure 49 – Event data structure

1822

Button (o)

1823

Buttons are intended for implementing a command interface to the device. Several commands can be implemented on the same variable / record item using different values to be written.

1824

1825

If this element is given, the IO-Link Tool shall display a button instead of a value. The attributes 'gradient', 'offset', 'unitCode' and 'displayFormat' shall not be used when the element 'Button' is present.

1826

1827

1828

The button shall be labelled with the Name that is given to the SingleValue at the data type of the referenced Variable or RecordItem whose 'value' corresponds to the 'buttonValue'.

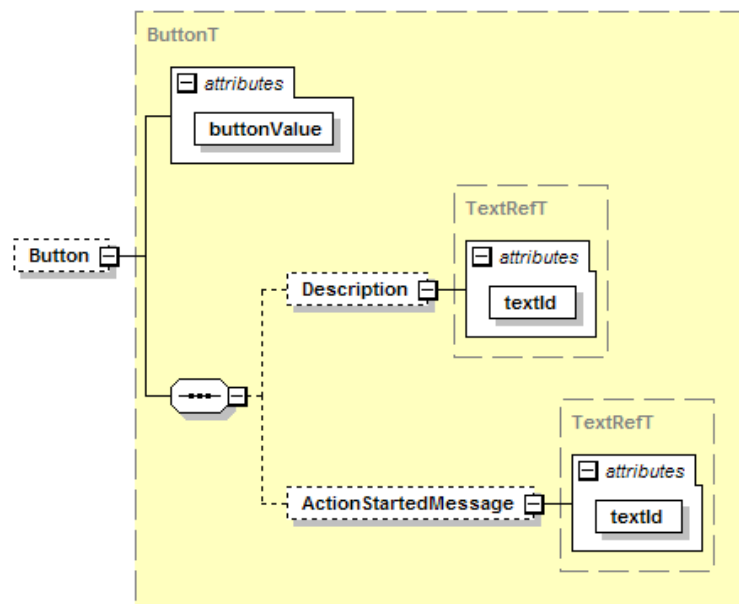
1829

1830

Use of this element is restricted to the data types BooleanT, UIntegerT and IntegerT, but it is highly recommended to use data type UIntegerT with bitLength="8" Note that this does not include arrays of these data types.

1831

1832



1833

1834

Figure 50 – Button element

1835

A variable referenced as "Button" shall have accessRights "wo"

1836

1837 shall only be displayed as a button
 1838 shall not be used as a condition variable, to switch menus or process data.

1839 The buttonValue
 1840 will be sent to the device immediately by pushing the button.
 1841 shall not be part of the block-download sequence.

1842 **buttonValue (m, union of boolean, unsignedLong and long)**

1843 This value must correspond to a SingleValue/@value of the data type of the referenced Variable
 1844 or RecordItem. It is sent to the device when the button is clicked.

1845 **Description (c)**

1846 **textId (m, RefT)**

1847 A text that explains the action that will be started by pressing the button.

1848 • For button values described as StdSingleValueRef, the Description Element is mandatory.
 1849 The definition in template IODD-SystemCommandDefinitions_V113.xml shall be used.

1850 • For button values described as SingleValue the Description Element is optional.

1852 **ActionStartedMessage (o)**

1853 **textId(m, RefT)**

1854
 1855 A text that is shown after the button value was successfully sent to the device. Use this as a
 1856 feedback to the user for actions that may take a while to complete or that require some user
 1857 action to complete.

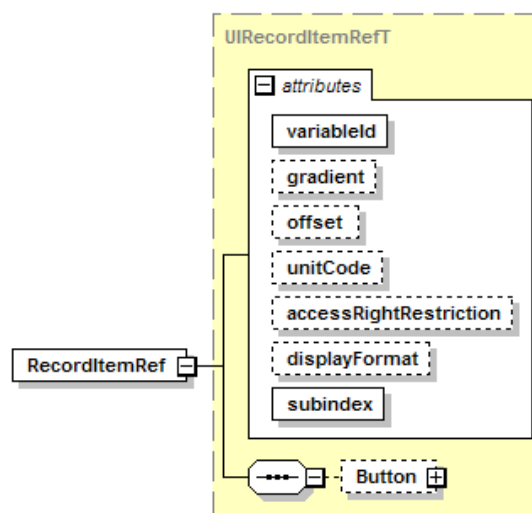
1858 ActionStartedMessages shall be omitted for buttons referencing StdSingleValueRefs
 1859 [@id="128" or "129" or "130" or "131"].

1860

1861 **7.5.8.5 RecordItemRef**

1862 Corresponds to VariableRef with an additional subindex. The variable referenced by variableId
 1863 shall be of type record. If 'Button' is specified, the referenced variable shall support subindex
 1864 access. A RecordItemRef with variableId="V_DeviceAccessLocks" is only allowed for those
 1865 subindices which refer to an access lock that is supported, i.e. where the respective attribute
 1866 in Features/SupportedAccessLocks is set to "true".

1867



1868

1869

Figure 51 – RecordItemRef element

1870 **subindex (m, SubindexT)**

1871 Addresses the record item of a variable of type record.

For the other attributes and the element 'Button', see VariableRef above.

7.5.8.6 MenuRef

Reference to a (sub)menu nested inside this menu.

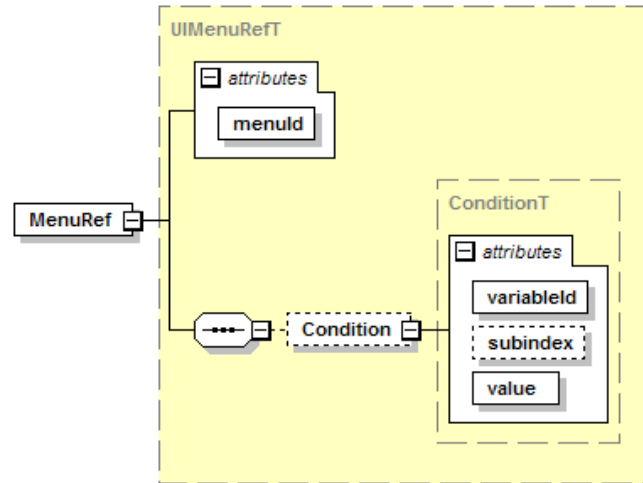


Figure 52 – MenuRef element

menuId (m, RefT)

References the (sub)menu from the MenuCollection.

Condition (o)

Condition for the display of this menu; an IO-Link Tool shall show the referenced menu only if the value of the referenced variable / record item equals the value of the attribute 'value'.

variableId (m, RefT)

References a variable. The variable shall be of data type BooleanT, IntegerT, UIntegerT or RecordT. The variable shall have a default value if it is not of type RecordT.

subindex (c, SubindexT)

This attribute shall be given if and only if the referenced variable is of type RecordT. Used for addressing the record item within the record. The record item shall be of data type BooleanT, IntegerT or UIntegerT and shall have a default value.

value (m, unsignedByte)

Must be a valid value for the variable / record item. This attribute can only hold values 0..255, thus limiting the possible IntegerT and UIntegerT values. Also, BooleanT condition values shall be entered as 0 for "false" and 1 for "true".

Conditions may be used in all menu levels.

Condition variables shall appear as VariableRef or RecordItemRef at least in a read-only way in a menu which is referenced in the same user role.

If there is more than one ProcessData element, selected by conditions, and the variable V_ProcessDataInput or V_ProcessDataOutput is referenced in a menu, one of the following shall hold:

- The type of reference (VariableRef / RecordItemRef) and the gradient, offset, unitCode and displayFormat fit to each of the ProcessData elements.
- The menu is conditioned in the same way as one of the ProcessData elements, and the type of reference (VariableRef / RecordItemRef) and the gradient, offset, unitCode and displayFormat fit to this particular ProcessData element.

"Conditioned in the same way" means that this or one of the parent menus has the same condition (same variable, same subindex, same value).

7.5.9 Rules for write-only variables

For variables with accessRights="wo" (write-only) the following rules shall be considered. If not accompanied with attribute displayFormat="button":

- "wo"- Variables generally are handled as commands
- "wo"- Variables are edited like any other variable
- "wo"- Variables shall never be part of any download sequence
- "wo"- Variables shall always be handled as a single write request

7.6 Communication characteristics

CommNetworkProfile

Figure 53 – CommNetworkProfile element

Excursion on XML schema *abstract types*:

An abstract type can't be used itself. Only non-abstract types which are derived from an abstract type can be used. The instance selects the desired derived type with xsi:type="name of the derived type".

This technique is used here with the 'CommNetworkProfile' element to adapt the XML structure to the requirements of the specific communication. This allows easy extension of the IODD to non-IO-Link devices with different communication characteristics as long as the applicative concept remains the same (i.e. addressing via index/subindex, standardized variables).

For IO-Link, the following derived type IOLinkCommNetworkProfileT describes the communication characteristics of an IO-Link interface.

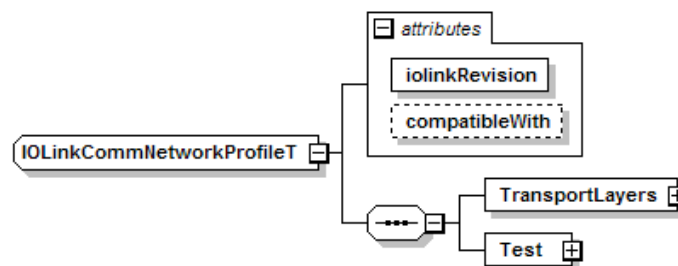


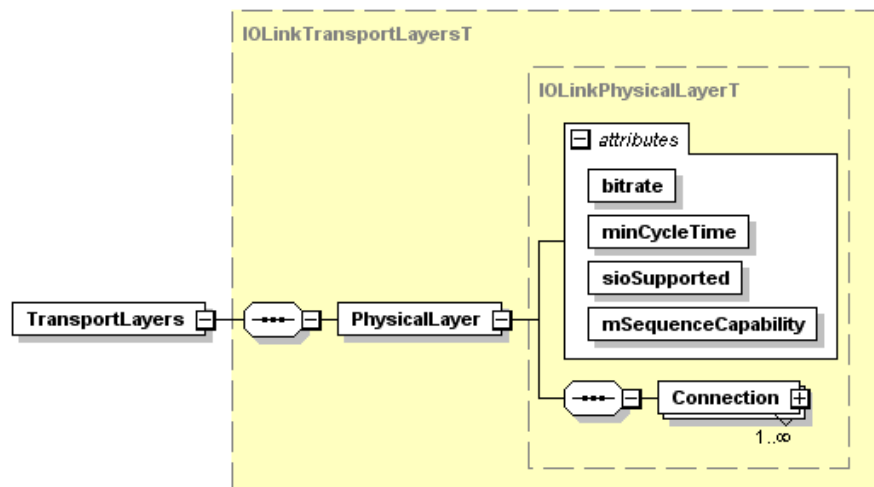
Figure 54 – CommNetworkProfile element – IO-Link variant

iolinkRevision (m, VersionT)

Implemented protocol version. Fixed to "V1.1".

compatibleWith (o, VersionT restricted to "V1.0")

Specify this attribute if the device is compatible with IO-Link revision 1.0, i.e. also runs on a V1.0 IO-Link Master. This requires an IODD V1.0.1 (or V1.0).

1934 **TransportLayers (m)**1935
1936 **Figure 55 – TransportLayers element – IO-Link variant**1937 **PhysicalLayer (m)**1938 **bitrate (m, string)**

1939 Allowed values are "COM1", "COM2" or "COM3".

1940 **minCycleTime (m, unsignedInt)**

1941 The minimum cycle time of the slave; specified in 1 microsecond (µs) units. E.g. the value
 1942 2300 represents 2.3 milliseconds (ms). The allowed value range is 0..6300 in steps of 100,
 1943 6400..31600 in steps of 400 and 32000..132800 in steps of 1600 (see **IO-Link Interface and**
 1944 **System Specification Version 1.1.3**, chapter B.1.3).

1945 **sioSupported (m, boolean)**

1946 Whether the fall-back to SIO mode is supported.

1947 **mSequenceCapability (m, unsignedByte)**

1948 Enter the content of V_DirectParameters_1, subindex 4 (M-sequence Capability) here as a
 1949 decimal number.

1950 **Test (m)**

1951 Contains information to enable automatic testing of the device.

1952 Enter appropriate data for the ISDU and event test configurations (see **IO-Link Test**
 1953 **Specification Version 1.1.3**).

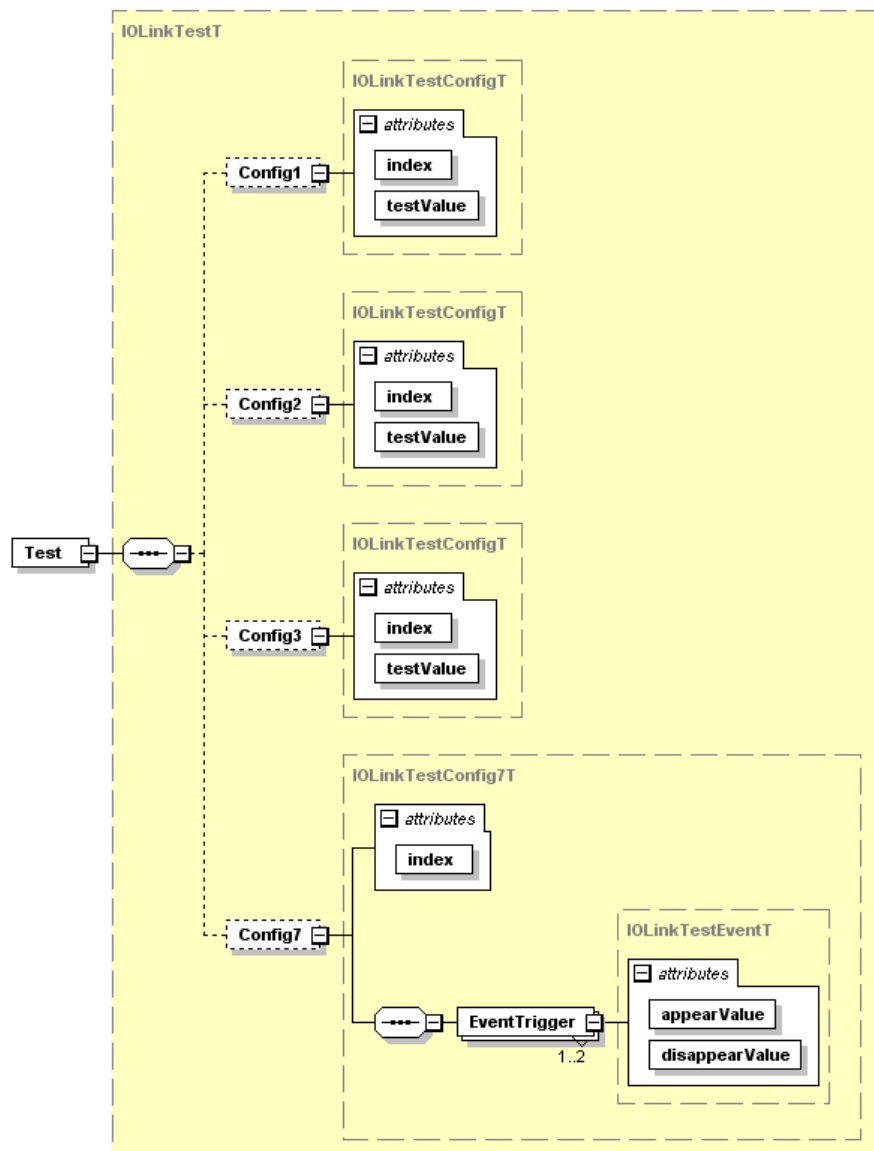


Figure 56 – Test element

Config1 (c)

Shall be present if the device supports ISDU access, and refers to a read-write 8 bit index. The testValue shall be small enough (≤ 12 octets) so that the ExtLength coding of the ISDU is not used.

Config2 (c)

Shall be present if the device supports ISDU access, and refers to a read-write 16 bit index. If the device supports a read-write variable on such an index, the testValue shall be accepted by the device. If the device does not support such an index a valid ErrorType shall be returned (Index not available).

Config3 (c)

Shall be present if the device supports ISDU access, and refers to a read-write 8 bit index. The testValue shall have a length big enough (> 12 octets) to cause the ExtLength coding of the ISDU.

Config7 (c)

Shall refer to an index raising different types of events. For details, please refer to the IO-Link Test Specification.

index (m, RefT)

References an Index to be used for testing.

1974 **testValue (m, string with pattern: “(0x[0-9A-Fa-f][0-9A-Fa-f],)*0x[0-9A-Fa-f][0-9A-Fa-f]”)**

1975 Must be an acceptable octet string value for the index.

1976 **EventTrigger (m)**

1977 Trigger values for up to two events.

1978 **appearValue (m, unsignedByte)**

1979 The value that triggers an event when written to Config7/@index.

1980 **disappearValue (m, unsignedByte)**

1981 The value that quenches the event triggered by @appearValue when written to Config7/@index.

1982 **Connection (m)**

1983 Describes, how the device can be connected. A Connection may be used for multiple device
1984 variants, which are referenced by the ProductRef/@productid attributes.

1985 This element has the following XML abstract type:

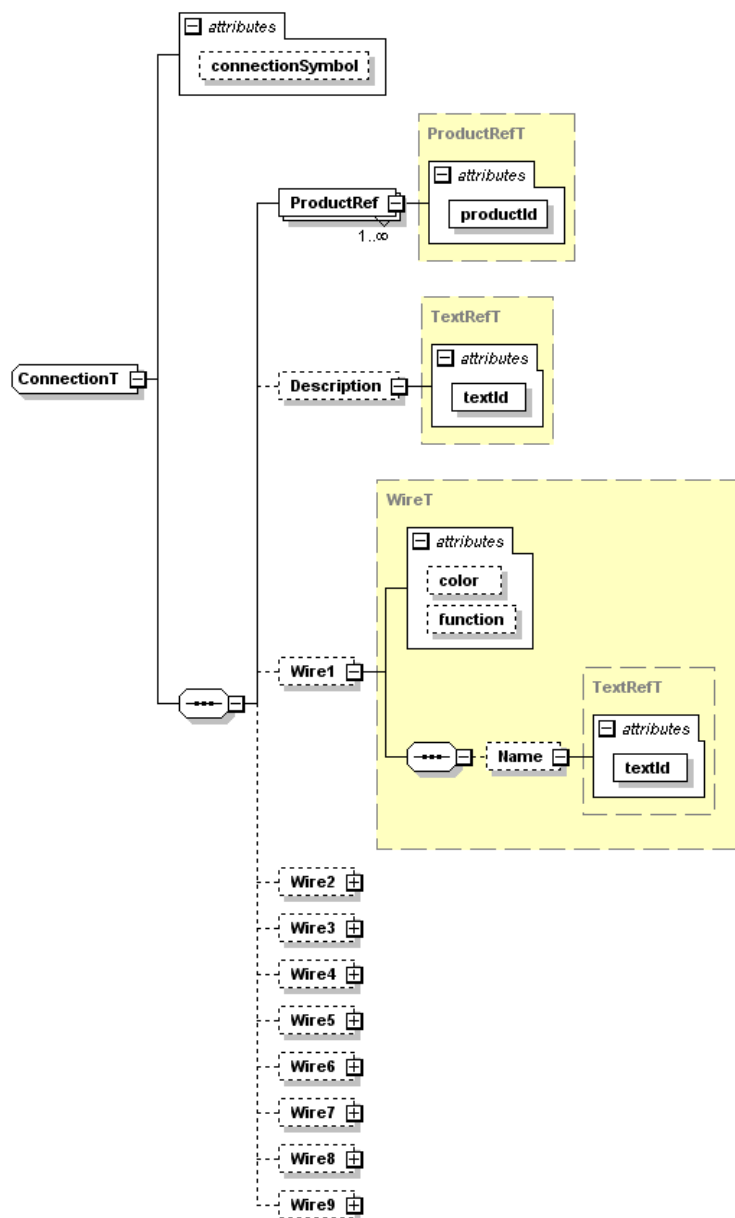


Figure 57 – ConnectionT abstract type

1988 **connectionSymbol (o, string with pattern “([\p{L}\d_#]+-)+con-pic\.**
 1989 File name of the connection symbol. If this attribute is used, the referenced image file shall be
 1990 present.

1991 **ProductRef (m)**
 1992 **productId (m, string)**
 1993 Selects the device variants that use this Connection. Must correspond to one of the
 1994 DeviceIdentity/DeviceVariantCollection/DeviceVariant/@productId values.

1995 **Description (o)**
 1996 **textId (m, RefT)**
 1997 Description of the connection.

1998 **Wire<X>**
 1999 Describes one of the wires. If the connection is some type of connector, the number <X> also
 2000 designates the pin / hole number.

2001 For OtherConnectionT and CableConnectionT, if the number of wires exceeds 9, the wires
 2002 relevant for IO-Link shall be described. In this case, the <X> does not necessarily equal the pin
 2003 / hole number. The real pin number should be described in the Wire<X>/Name element.

2004 **color (o, string)**
 2005 A color code according to IEC 60757:1983.

2006 **Table 4 – Wire colors**

Code	Color
“BK”	Black
“BN”	Brown
“RD”	Red
“OG”	Orange
“YE”	Yellow
“GN”	Green
“BU”	Blue (including light blue)
“VT”	Violet (purple)
“GY”	Grey (slate)
“WH”	White
“PK”	Pink
“GD”	Gold
“TQ”	Turquoise
“SR”	Silver

2007
 2008 **function (o, string)**
 2009 The function of the wire.

2010 **Table 5 – Wire functions**

Function	Description
“NC”	Not connected
“L+”	Power supply (+), pin 1, brown
“L-”	Power supply (-), pin 3, blue
“P24”	Extra power supply (+)
“N24”	Extra power supply (-)
“Other”	e.g. signal (DI, DO, analog) or power supply
“C/Q”	Communication signal, pin 4, black

2011
 2012 **Name (o)**
 2013 **textId (m, RefT)**
 2014 An additional textual description of the wire. Do not repeat the wire color or wire function in
 2015 textual form here.

2016 The allowed XML derived types are: M5ConnectionT, M8ConnectionT, M12-4ConnectionT,
 2017 M12-5ConnectionT, OtherConnectionT and CableConnectionT. They restrict the abstract type
 2018 accordingly.

2019 **M5ConnectionT**

2020 The M5 connector as specified in **IO-Link Interface and System Specification Version 1.1.3**,
 2021 chapter 5.5.

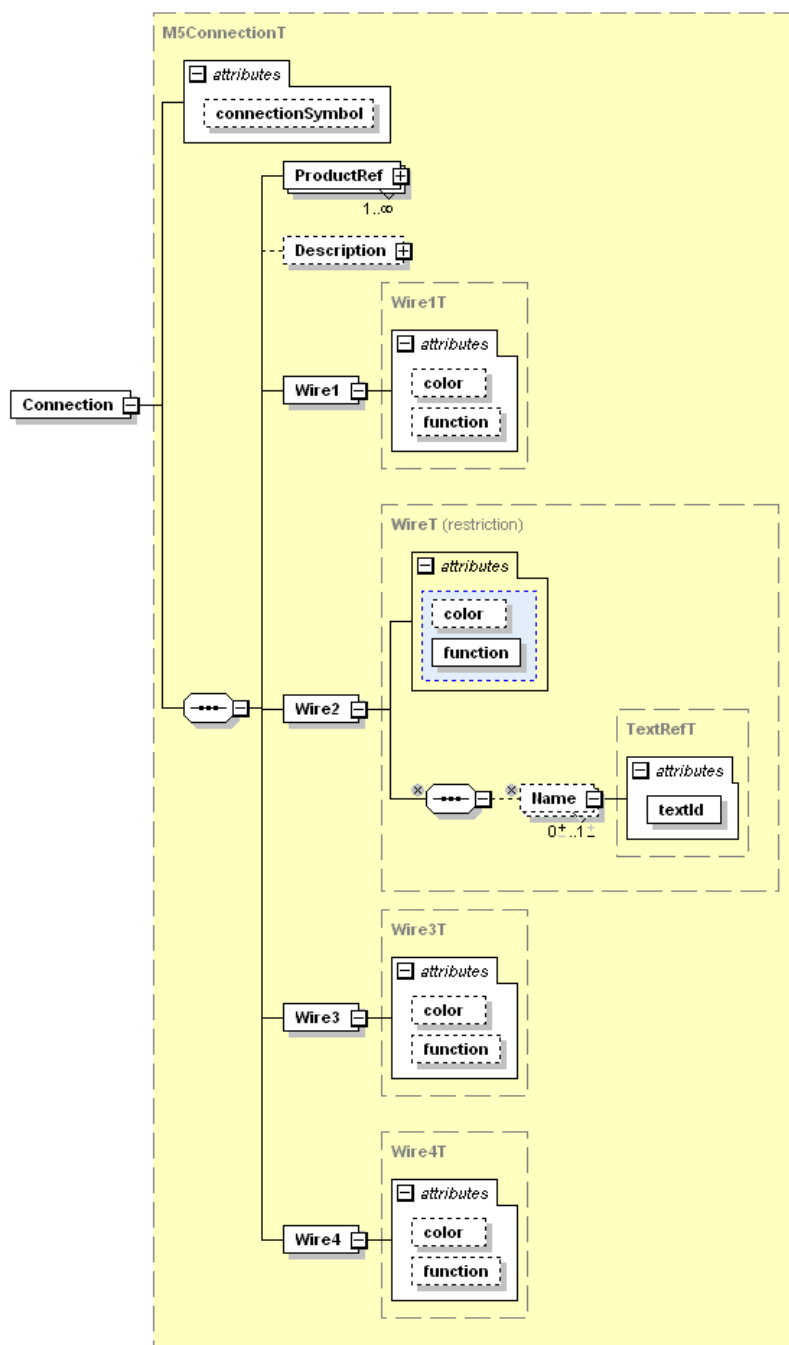


Figure 58 – Connection element – M5ConnectionT variant

2024 Wire1, Wire3 and Wire4 have fixed color and function. No Name is allowed.

2025 Wire 2 has a fixed color “WH” (white), and a function restricted to “NC” or “Other”. The function
 2026 attribute is mandatory.

2027 **M8ConnectionT and M12-4ConnectionT**

2028 Same as M5ConnectionT.

2029 **M12-5ConnectionT**

2030 The M12-5 connector as specified in **IO-Link Interface and System Specification Version**
 2031 **1.1.3**, chapter 5.5.

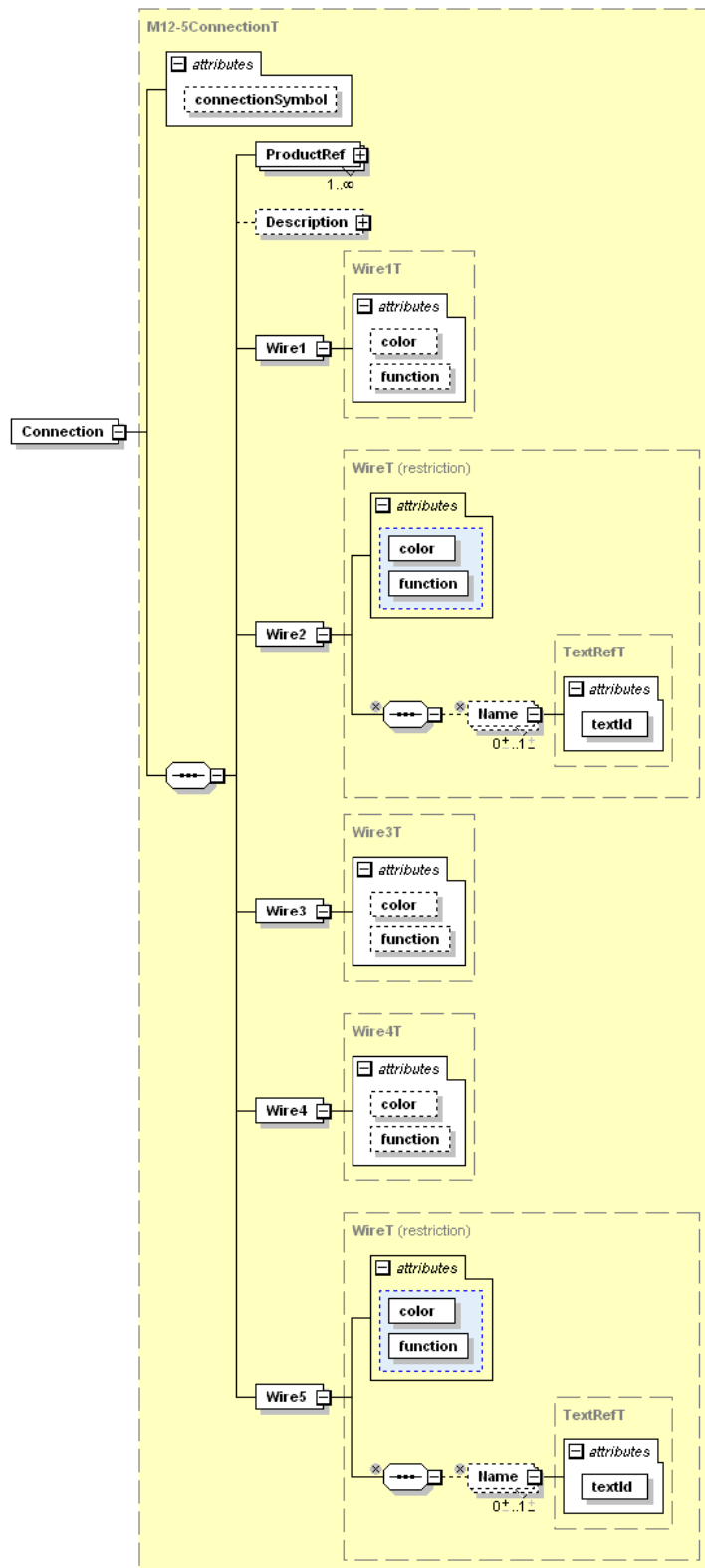


Figure 59 – Connection element – M12-5ConnectionT variant

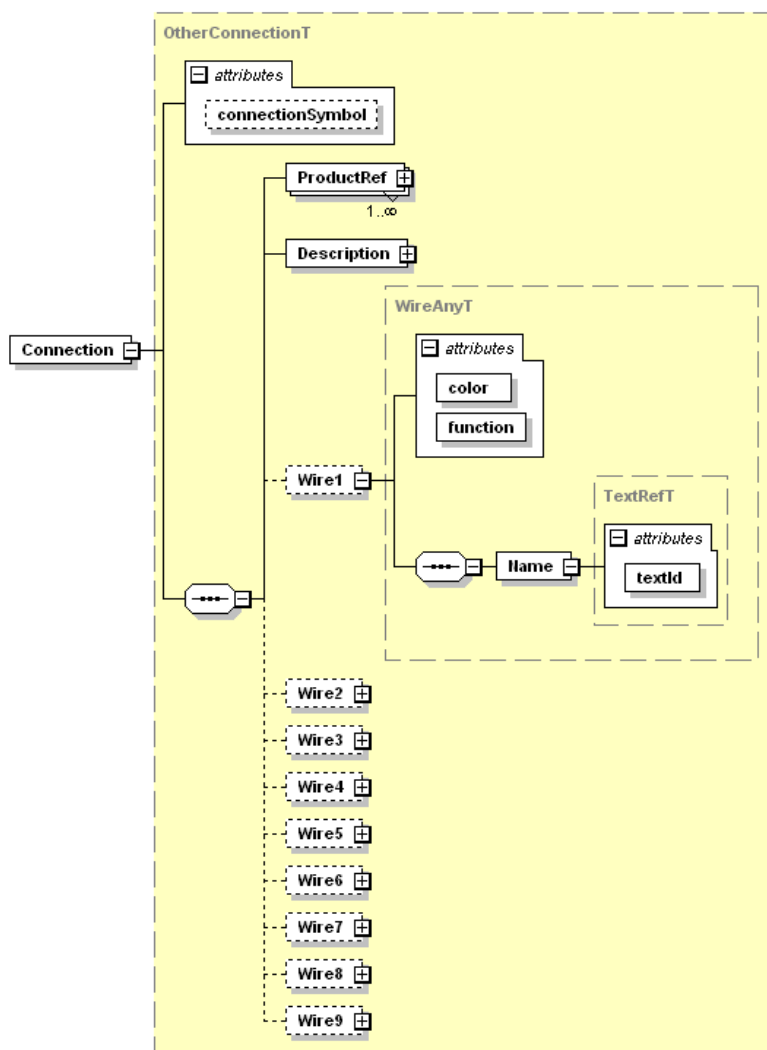
Wire1, Wire3 and Wire4 have fixed color and function. No Name is allowed.

2035 Wire 2 has a function restricted to “NC”, “P24” or “Other”. If its function is I/Q, its color shall be
 2036 “WH” (white). The color and function attributes are mandatory.

2037 Wire 5 has a function restricted to “NC” or “N24”. The color and function attributes are
 2038 mandatory.

2039 **OtherConnectionT**

2040 Some non-standard connector.



2041
 2042 **Figure 60 – Connection element – OtherConnectionT variant**

2043 The Description is mandatory. For all given Wires, the element ‘Name’ and the attributes ‘color’
 2044 and ‘function’ are mandatory.

2045 Wires with functions “L+”, “L-” and “C/Q” shall be present.

2046 **CableConnectionT**

2047 Same as OtherConnectionT, but Description is not mandatory. The wire number in the
 2048 ‘Wire...’ element name does not designate a pin number here, but any arbitrary numbering of
 2049 the wires.

2050 **7.7 Language dependent description texts**

2051 All text components of the different languages are given in the ExternalTextCollection. There
 2052 may be one or more languages deposited. Additional languages may be stored in separate files.

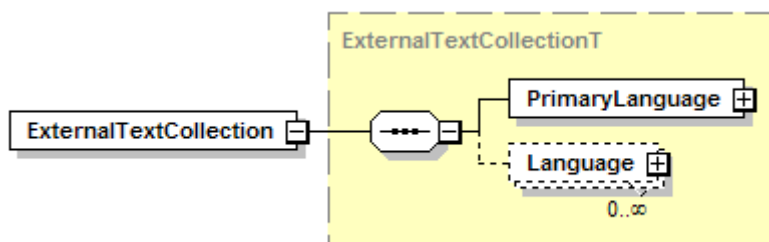


Figure 61 – ExternalTextCollection element

7.7.1 PrimaryLanguage (m)

Shall be in English.

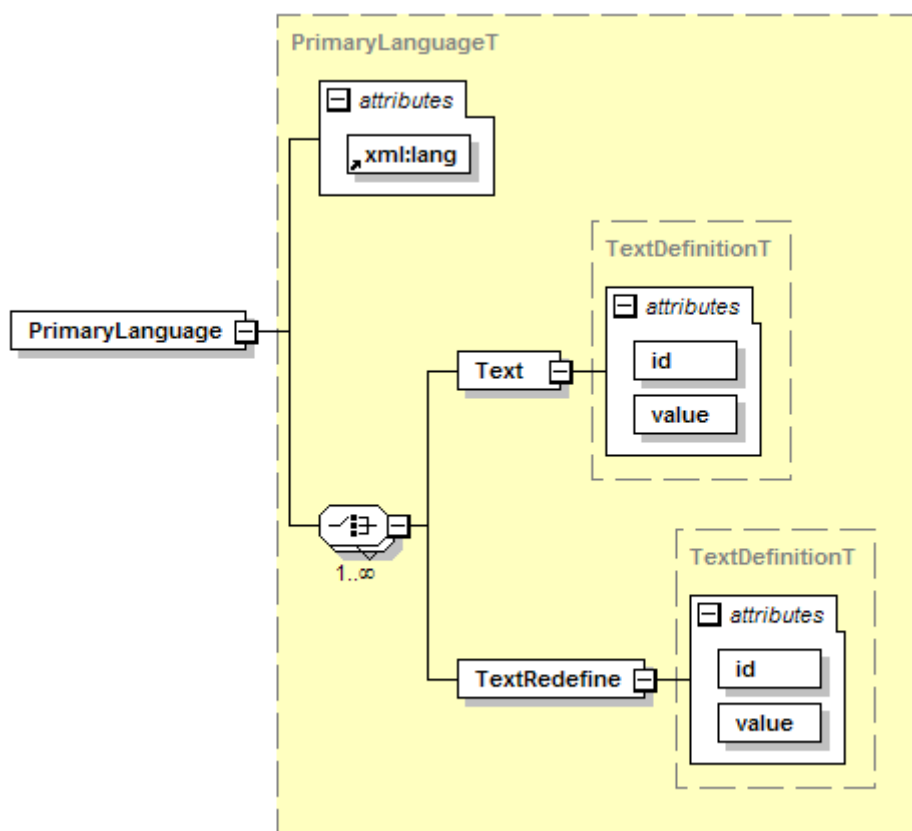


Figure 62 – PrimaryLanguage element

xml:lang (m, language)

The code for the language according to ISO 639-1:2002. Shall be “en” for English.

Text (m)

Language dependent text which is referenced by its id.

id (m, IdT)

Shall be referenced by other elements via their ‘textId’ attribute (there shall be no unreferenced Text elements)

value (m, string)

Text in the denoted language.

NOTE: Special characters shall be coded according to the XML syntax. See REC-xml-20081126, chapter 2.4 Character Data and Markup.

& → &

2071 ' → ' (only required if inside a string enclosed in ' characters)
2072 > → >
2073 < → <
2074 " → " (only required if inside a string enclosed in " characters)
2075 LF →

2076 Only the line feed is allowed for formatting the text.

2077 **TextRedefine (o)**

2078 Language dependent text which overrides a standard text. Only applicable for texts describing
2079 the octets of DirectParameter page 2.

2080 **id (m, IdT)**

2081 Shall be one of STD_TN_DeviceSpecific_1 to STD_TN_DeviceSpecific_16.

2082 **value (m, string)**

2083 Text in the denoted language. The **Note** at Text/@value also applies.

2084 **7.7.2 Language (o)**

2085 Optional specification of texts in another language. The attribute 'xml:lang' specifies the
2086 language (see ISO 639-1:2002). The structure of this element corresponds to the structure of
2087 the element PrimaryLanguage.

2088 **8 Compatibility**

2089 IO-Link devices conforming to the **IO-Link Interface and System Specification Version 1.1.3.**

2090 IO-Link devices conforming to the IO-Link Communication Specification Version 1.0 shall be
2091 described with an IODD according to the IO Device Description Specification Version 1.0.1 (or
2092 the legacy version 1.0).

2093 It is not possible to describe a V1.0 device with an IODD V1.1 or a V1.1 device with an IODD
2094 V1.0.1 or V1.0.

2095 Two IODDs having the same vendorId and deviceId, one being based on V1.0.1 (or V1.0) and
2096 the other being based on V1.1 are only allowed in the following use cases:

- 2097 • A V1.0 device exists and has a V1.0.1 (or V1.0) IODD. A new, compatible version of the
2098 device is built using the same vendorId and deviceId based on V1.1 needing a V1.1 IODD.
- 2099 • A V1.1 device is newly built with a new vendorId /deviceId pair. It is also V1.0 compatible
2100 and thus needs two IODDs.

2101 In both cases, the DeviceFamily and all DeviceVariant/Name (previously ProductName) for all
2102 languages, as well as all DeviceVariant/@productId must be same.

2103 **An IO-Link tool may show V1.1 as well as V1.0.1 (or V1.0) IODDs in its catalog. According**
2104 **compatibility feasibility of the IO-Link master, the appropriate IODD may be instantiated.**

2105 Annex A IODD schemas

2106 The following schemas and standard definition files are part of this specification:

2107 Schema files

- | | | |
|------|-------------------------------------|---|
| 2108 | • IODD1.1.xsd | main IODD schema |
| 2109 | • IODD-Primitives1.1.xsd | basic definitions |
| 2110 | • IODD-Datatypes1.1.xsd | data types |
| 2111 | • IODD-Events1.1.xsd | events |
| 2112 | • IODD-Variables1.1.xsd | variables |
| 2113 | • IODD-UserInterface1.1.xsd | user interface |
| 2114 | • IODD-Communication1.1.xsd | communication network profile |
| 2115 | • IODD-StandardDefinitions1.1.xsd | main schema for the standard definition files |
| 2116 | • IODD-WirelessCommunication1.1.xsd | wireless communication network profile |

2117 Standard definition files

- | | | |
|------|---------------------------------------|--|
| 2118 | • IODD-StandardDefinitions1.1.xml | list of standard variables, error types and |
| 2119 | | events + english texts |
| 2120 | • IODD-StandardDefinitions1.1-de.xml | german texts |
| 2121 | • IODD-StandardDefinitions1.1-es.xml | spanish texts |
| 2122 | • IODD-StandardDefinitions1.1-fr.xml | french texts |
| 2123 | • IODD-StandardDefinitions1.1-it.xml | italian texts |
| 2124 | • IODD-StandardDefinitions1.1-ja.xml | japanese texts |
| 2125 | • IODD-StandardDefinitions1.1-ko.xml | korean texts |
| 2126 | • IODD-StandardDefinitions1.1-pt.xml | portuguese texts |
| 2127 | • IODD-StandardDefinitions1.1-ru.xml | russian texts |
| 2128 | • IODD-StandardDefinitions1.1-zh.xml | chinese texts |
| 2129 | | |
| 2130 | • IODD-StandardUnitDefinitions1.1.xml | list of available unit codes + english texts |

Annex B Definitions of IODD quantity structure

The following table lists limits on the number and the length of elements of the IODD. IODDs shall not exceed these limits. IO-Link Tools shall accept all IODDs that do not exceed these limits.

Table 6 – IODD quantity structure

Element	Maximum Number/Length	Comments
Length of filename	255	
Number of DeviceVariants	255	
Number of ValueRanges	32	See note
Number of SingleValues	255	See note
Number of ProcessData Elements	16	
Number of Datatypes	255	See note
Number of Variables	1024	
Length of text which is referenced by DeviceVariant/Description/@textId in all languages	1024	Characters, not octets
For all others Description/@textId: Length of text which is referenced by Description/@textId in all languages	255	Characters, not octets
Length of URL which is referenced by VendorUrl/@textId in all languages	255	Characters, not octets
Length of @vendorName and all other texts referenced by a @textId (e.g. VendorText, DeviceName, DeviceFamily, Name, ...)	64	Characters, not octets
Length of DocumentInfo/@copyright text	255	Characters, not octets
Number of Menus in MenuCollection	255	
Number of elements per menu (VariableRef+RecordItemRef+MenuRef)	64	
Variable/@id length	64	Characters, not octets
Datatype/@id length	64	Characters, not octets
@textId length	64	Characters, not octets
Number of supported languages	see ISO 639-1	Currently 185
Number of Menu Levels	3	

Note: The entries "Number of <element name>" are not meant as the total number of XML elements with that name in the whole IODD. Instead, they are meant as "Number of XML elements of that name within the superordinated element".

"Number of Datatypes" is the number of Datatype elements within the DatatypeCollection element. This does not constitute the total number of data types which may appear in an IODD, because each Variable and ProcessDataIn/ProcessDataOut element may additionally use an embedded data type.

"Number of Variables" is the sum over all StdVariableRef, DirectParameterOverlay and Variable elements within the VariableCollection.

Annex C IODD test (normative)

1.1 General

An IODD shall conform to the underlying schema files and a set of business rules, defined in this specification.

1.2 Schema test via an IODD Checker tool

The IO-Link Community provides the IODD Checker for download free of charge from its web site. It is mandatory for each IODD associated with a Device to pass the test with this Checker. The Device's manufacturer declaration shall state the successful result of the test.

The latest released version of the IODD Checker shall be applied.

The requirements for the Checker consist of two main parts.

Within the first part the Checker uses the following parsers or their later versions or alternatives where applicable to test the schema consistency of a particular IODD:

- XmlReader class of Microsoft .NET Framework Version 2.0 with schema validation switched on, i.e. *Settings.ValidationType* set to *ValidationType.Schema*.

Microsoft .NET Framework Version 2.0 Redistributable Package,
date published: 22-Jan-2006,
<http://www.microsoft.com/downloads/details.aspx?familyid=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&displaylang=en>

Microsoft .NET Framework 2.0 Service Pack 2
date published: 16-Jan-2009,
<http://www.microsoft.com/downloads/details.aspx?FamilyID=5b2c0358-915b-4eb5-9b1d-10e506da9d0f&displaylang=en>

- Xerces-C++ Version 3.1.1,
<http://xerces.apache.org/xerces-c/>

Within the second part the Checker uses the following business rule set in Table 7.

Column 'Reference' indicates either the chapter within this document or with prefix [1] the chapter within the IO-Link Interface and System Specification Version 1.1.3.

An "X" in column "M" indicates relevance for the main IODD

An "X" in column "L" indicates relevance for the language file.

An "X" in column "W" indicates that this business rule check creates a warning.

An "X" in column "H" indicates 'highly recommendation'. Checker creates a warning, which cannot be disabled or made invisible.

An "X" in column "V" indicates validity of the check for legacy Devices (V1.0).

Table 7 – Checker business rule set for IODDs

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0001	Encoding	5	Encoding shall be UTF-8	X	X			X
0002	NameSpace	5	http://www.w3.org/2001/XMLSchema-instance with the prefix "xsi" http://www.io-link.com/IODD/2010/10 with the prefix "iodd"	X	X			

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0003	AdditionalNameSpaces	5	No additional name spaces shall be included	X	X			X
0004	SchemaLocation	5	http://www.io-link.com/IODD/2010/10	X	X			
0005	SchemaMainIODD	5	IODD1.1.xsd	X				
0006	SchemaLanguageFile	5	IODD-Primitives1.1.xsd		X			
0007	FileNameConvention - IODD SpecialCharacters	5	Special characters are permitted in vendor name and device name part of filename: _, #, -	X	X			X
0008	FileNameConvention IODD-VendorNamePart	5	All files of the set of files belonging to a specific IODD shall have the same <vendor name> part in their file names.	X	X			X
0009	FileNameConvention MainIODD	5.1	<vendor name>-<device name>-<date of file creation>-IODD<schema version>.xml; IODD shall be done with upper case letters	X				X
0010	FileNameConvention LanguageFile	5.2	<vendor name>-<device name>-<date of file creation>-IODD<schema version>-<language>.xml; IODD shall be done with upper case letter		X			X
0011	LanguagePart - LanguageFileName-ISO	5.2	The "language" part follows ISO 639-1:2002.		X	X		X
0012	LanguagePart – LanguageFileName-2Letters	5.2	The "language" part consists of two letters		X			X
0013	LanguagePart – LanguageFileName-Unique	5.2	There shall be no additional language file for languages already covered in the main IODD file		X			X
0014	LanguagePart – LanguageFileName-Inside	5.2	The language part of the language file name shall be the same as the definition inside the language file		X			X
0015	LanguageStandard-Definitions	5.2	If an IODD contains a language, which is not existing for IODD-StandardDefinitions, the checker will show a warning	X	X	X		
0016	VendorLogo	5.3	160 x 90 pixel, landscape format	X				X
0017	DeviceIcon	5.3	48 x 48 pixel	X				X
0018	DevicePicture	5.3	Min. 160 x 160 pixel, max. 320 x 320, square	X				X
0019	ConnectionSymbol	5.3	Min. 160 x 160 pixel, max. 320 x 320, square	X				
0020	ImageFilesExist	5.3	If the attributes are used, the referenced image files shall be present	X				X
0021	ImageFileNameConvention	5	All files of the set of files belonging to a specific IODD shall have the same <vendor name> part in their file names	X				X
0022	DateIODD - FileName	7.3.1	The date information in the IODD file name shall correspond to the releaseDate attribute in the DocumentInfo element	X				X
0023	DateLanguage - FileName	7.3.1	The date information in the language file name shall correspond to the releaseDate attribute in the DocumentInfo element, if the DocumentInfo element exists		X			X
0024	ProfileHeader	7.3.2	It shall correspond exactly to the given values in the specification	X				X

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0025	Stamp	7.3.4	If no errors are detected during the checking process, the crc attribute is set to a CRC value calculated across the file contents. Otherwise, the crc attribute is set to an invalid value.	X	X			X
0026	Comments - InOr-AfterStamp	7.3.4	Comments shall not be included in or after the Stamp element.	X	X			X
0027	ProductId	7.4.1	ProductID in IODD corresponds to the ISDU standard parameter. Multiple device variants are only allowed, if StdVariableRef with id="V_ProductID" is present.	X				X
0028	ProductId-Length	7.4.1	The maximum length of ProductId shall not exceed 64 octets UTF-8 coded.	X				X
0029	Declarations-Data-types	7.5.2	There shall be no unreferenced data type elements.	X				X
0030	DatatypeId	7.5.2	For data types in the DatatypeCollection, the attribute id shall be specified.	X				X
0031	NoDatatypeId	7.5.2	For Datatypes outside the DatatypeCollection, the attribute id shall not be specified	X				X
0032	StdVariableRef	7.5.4.1	V_DirectParameters_1 shall always be referenced. All standard ISDU variables marked with the attribute mandatory="true" in the IODD-StandardDefinitions shall be referenced, if the Device supports ISDU access.	X				X
0033	V_DirectParameters_2 and DirectParameterOverlay	7.5.4.1	StdVariableRef with id="V_DirectParameters_2" and DirectParameterOverlay may only be present both or none.	X				
0034	DirectParameterOverlay reference in menu	7.5.4.1	If StdVariableRef with id="V_DirectParameters_2" and DirectParameterOverlay are present, V_DirectParameters_2 shall not be referenced and DirectParameterOverlay shall be referenced in menu.	X				
0035	StdVariableRef - ReservedIds	7.5.4.2 7.5.4.3 [9], B.2.1	The id of any standard variable shall not be used as id for the Variable or DirectParameterOverlay, even if the standard variable is not referenced from this IODD.	X				X
0036	StdVariableRef - DefaultValue	7.5.4.1	@defaultValue shall not be specified for references to V_ProcessDataInput or V_ProcessDataOutput	X				X
0037	FixedLength-Restriction	7.5.4.1	If referenced variable is of type OctetString or String, it shall be less or equal its fixedLength attribute. If referenced variable is of type Array, it shall be less or equal to its count attribute.	X				X
0038	V_Application-SpecificTag	[1], B.2.16	If fixedLengthRestriction is used, it shall be equal or greater than 16.	X				X
0039	StdSingleValueRef	7.5.4.1	Check if StdSingleValue exists as SingleValue in StdDefinitions1.1.xml.	X				X
0040	StdValueRangeRef	7.5.4.1	Check if StdValueRange exists as ValueRange in StdDefinitions1.1.xml.	X				X
0041	StdVariableRef - SingleValue	7.5.4.1	Check against overlapping with StdSingleValue and StdValueRange (even if they are optional and not referenced), other SingleValue or ValueRange.	X				X

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0042	StdVariableRef - ValueRange	7.5.4.1	Check against overlapping with StdSingleValue and StdValueRange (even if they are optional and not referenced), other SingleValue or ValueRange.	X				X
0043	StdRecordItemRef	7.5.4.1	Check if referenced RecordItem exists in StdDefinitions1.1.xml.	X				X
0044	DirectParameter-Overlay - Subindex-AccessSupported	7.5.3.2.3 [1], B.1.1	Statement subindexAccessSupported="false" shall not be used. Index 1 can only be accessed octet by octet.	X				
0045	DirectParameter-Overlay - Datatype	7.5.4.2	The data type shall be a record.	X				
0046	DirectParameter-Overlay - Usage	7.5.4.1	It is highly recommended not to use V_DirectParameters_2 and ISDU in the same Device.	X		X	X	
0047	VariableIndex	[1] B.2.1	Preferred and extended Index for Device or vendor specific variables are 64 to 254 and 256 to 16383. Indices 2 to 63, 255, 16384 to 65535 shall not be used as Device or vendor specific variable.	X				X
0048	VariableIndex - Profiles and extensions	[1] B.2.1	If indices from ranges reserved for profiles and extensions are used, the checker shall generate a warning.	X		X		X
0049	RecordItemInfo	7.5.4.3	RecordItem shall exist.	X				X
0050	RecordItemInfo - NoRecord	7.5.4.3	Only applicable if the variable is of type record.	X				X
0051	DefaultValue	7.5.4.1 7.5.4.3	The default value shall match the given datatype.	X				X
0052	DefaultValue - String	7.5.4.1 7.5.4.3	Check whether used letters are valid in respect to given encoding.	X				X
0053	ProcessData - Condition	7.5.5	If ProcessData occur more than once, the individual ProcessData elements shall be distinguished by the Condition element.	X				X
0054	ProcessData - BitLength	7.5.5	The attribute 'bitLength' shall represent the underlying ProcessDataIn (-Out) datatype in a bit by bit manner. For record data types this bitLength shall equal the bitLength attribute of the record.	X				X
0055	ProcessData - ConditionVariable	7.5.5	There shall only be exactly one variable used for the switching of process data. The referenced variable shall contain a default value. The process data length (of ProcessDataIn and ProcessDataOut respectively) shall be the same for all ProcessData.	X				X
0056	ProcessData - ConditionDatatype	7.5.5	Conditions shall only be of datatype IntegerT, UIntegerT and BooleanT.	X				X
0057	ProcessData - Condition – Menu	7.5.5	All Variable / RecordItem referenced in the ProcessData/Condition elements shall be referenced from at least one menu within each user role.	X				
0058	StdErrorTypeRef	7.5.6	Check whether referenced ErrorType exists in StdDefinitions1.1.xml.	X				
0059	StdEventRef	7.5.7	Check whether referenced Event exists in StdDefinitions1.1.xml.	X				
0060	EventCode	[1], D.2	Vendor or device specific codes are: 0x1800-0x18FF and 0x8CA0-0x8DFF.	X				

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0061	EventCode – Profiles and extensions	[1], D.2	If EventCodes from ranges reserved for profiles and extensions are used, the checker shall generate a warning.	X		X		
0062	ProcessDataInfo	7.5.8.1	Check whether displayFormat and Datatype are matching.	X				
0063	MenuLevel	7.5.8.2	At most three menu levels below the role assignment are acceptable.	X				X
0064	NotUsedMenus	7.5.8.3	There shall be no unreferenced Menu elements	X				
0065	MenuName	7.5.8.3	For underlying menus, a menu name shall be specified.	X				X
0066	GradientOffset	7.5.8.4	When applying gradient and/or offset to convert the Variable or RecordItem value into the displayed value, the value will be implicitly converted to a floating-point value. Consequently, the only allowed displayFormat for such values shall be "Dec" or "Dec.*".	X				X
0067	RecordItemRef - Array	7.5.8.4	Single array members cannot be referenced by RecordItemRef.	X				X
0068	UnitCode	7.5.8.4	Unit code to which the indicated variable refers. See IODD-StandardUnitDefinitions1.1.xml for valid unit codes.	X				X
0069	Menu - AccessRightRestriction	7.5.8.4	The accessRight of the referenced element shall include the accessRights given by accessRightRestriction.	X				X
0070	ButtonValue	7.5.8.4	The value of the 'buttonValue' attribute shall be defined as a 'SingleValue' of the Variable/-RecordItem	X				X
0071	ButtonValue - Datatype	7.5.8.4	It is highly recommended to use data type UIntegerT with bitLength="8"	X		X	X	
0072	ButtonReference	7.5.8.4	A variable referenced as "Button" shall have accessRights "wo", shall only be displayed as a button, and shall not be used as a condition variable to switch menus or processdata.	X				X
0073	RecordItemRef	7.5.8.5	The variable referenced by variableId shall be of type record. The subindex shall be defined in the referenced record.	X				X
0074	RecordItemRef - Button	7.5.8.5	If referenced as "Button", the referenced variable shall support subindex access.	X				X
0075	Button - NoDisplay-Format	7.5.8.5	If in menu subelement 'Button' exists for VariableRef or RecordItemRef, the attributes 'gradient', 'offset', 'unitCode' and 'displayFormat' shall not be used.	X				
0076	MenuRef - CircularReferences	7.5.8.6	Circular references to menus are not allowed.	X				X
0077	MenuCondition	7.5.8.6	Conditions shall only be of datatype IntegerT, UIntegerT, and BooleanT.	X				X
0078	MenuCondition - ProcessDataCondition	7.5.8.6	If there is more than one ProcessData element selected by conditions, and the variable V_ProcessDataIn or V_ProcessDataOut is referenced in a menu, one of the following shall hold: a) The type of reference (VariableRef/-RecordItemRef) and the gradient, offset,	X				X

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
			unitCode and displayFormat match each of the ProcessData elements. b) The menu is conditioned in the same way as one of the ProcessData elements, and the type of reference (VariableRef / RecordItemRef) and the gradient, offset, unitCode and displayFormat match this particular ProcessData element ("conditioned in the same way" means that this menu or one of the parent menus has the same condition: same variable, same subindex, same value).					
0079	MinCycleTime	7.6 [1], B.1.3	The minimum cycle time of the device; specified in units of 1 µs. For example, 2300 represents 2,3 ms.	X				X
0080	PrimaryLanguage	7.7.1	The primary language shall be English (the attribute xml:lang shall have the value "en").	X				X
0081	TextRedefine	7.7.1	Language dependent text overriding a standard text. Only applicable for texts describing the octets of DirectParameter page 2. It shall be one of STD_TN_DeviceSpecific_1 to STD_TN_DeviceSpecific_16.	X				
0082	NotUsedTexts	7.7.1	There shall be no unreferenced Text elements.	X	X			X
0083	NotTranslatedTexts	7.7.1	Check whether texts are not translated.	X	X	X		X
0084	UniqueDeviceVariant - NameText	7.4.1	Texts referenced by DeviceVariantCollection/DeviceVariant/Name/t extId shall be unique within each supported language.	X	X			
0085	UniqueVariable - NameText	7.5.4	Texts referenced by StandardVariables, DirectParameterOverlay or Variables shall be unique within each supported language.	X	X			
0086	OverlappingSingle-Value - ValueRange	7.5.3.1.1	SingleValues and ValueRanges shall not overlap	X				X
0087	Arrays	[1], F.2.3 [1], F.2.4	UIntegerT and IntegerT with a length of ≥ 58 bit and < 64 bit are not permitted.	X				X
0088	Array Alignment	7.5.3.2.2	Array elements with bitLength < 8 bit shall not cross octet boundaries. For array elements with bitLength ≥ 8 bit use bitLength as multiples of 8 bit.	X				
0089	Record - Subindices	[1], F.3.3	The Subindices within the IODD shall be listed in ascending order from 1 to n describing an octet sequence. Gaps within the list of Subindices are allowed.	X				X
0090	RecordItems - Alignment Data-types	[1], F.3.3	The following data types shall always be aligned with octet boundaries: Float32T, StringT, OctetStringT, TimeT, and TimeSpanT.	X				X
0091	RecordItems - Alignment - Integer58	[1], F.3.3	UIntegerT and IntegerT with a length of ≥ 58 bit shall always be aligned with one side of an octet boundary.	X				X
0092	RecordItems – Alignment - Integer10	[1], F.3.3	It is highly recommended for UIntegerT and IntegerT with a length of ≥ 8 bit to align always with one side of an octet boundary.	X		X		X
0093	RecordItems - Alignment - Integer6	[1], F.3.3	It is highly recommended for UIntegerT and IntegerT with a length of < 8 bit not to cross octet boundaries.	X		X		X

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0094	RecordItems - Alignment - Integer	7.5.3.2.3	For variables, UIntegerT and IntegerT shall begin on an octet boundary.	X				
0095	RecordItems - Alignment - Boolean	7.5.3.2.3	For variables, it is highly recommended to group BooleanT together into one or few octets.	X		X	X	
0096	RecordItems - AccessRightRestriction	7.5.3.2.3	The accessRightRestriction of the RecordItem shall include the accessRights of the variable.	X				X
0097	ProcessData - AccessRightRestriction	7.5.3.2.3	The attribute "accessRightRestriction" is only applicable for ISDU parameter, not for RecordItem for process data.	X				X
0098	BitOffset	[1], F.3.3	RecordItem shall not overlap	X				X
0099	Connection - MinDef	7.6	Connection: If OtherConnectionT is used, there shall be wires with the function L+, L- and C/Q.	X				
0100	MaxLength – Variables	[1], 4.4	The length of a variable shall not exceed 232 octets.	X				X
0101	MaxLength - ProcessData	[1], 4.4	The length of a ProcessDataIn/Out shall not exceed 32 octets.	X				X
0102	ExcludedFromData Storage	7.5.4.1	This attribute may only be set to "true" for variables with accessRights = "rw".	X				
0103	ModifiesOtherVariables	7.5.4.2	This attribute may only be set to "true" for variables with accessRights = "rw" or "wo".	X				
0104	Dynamic	7.5.4.2	This attribute may only be set to "true" for variables with accessRights = "rw" or "ro".	X				X
0105	Quantity	B	Quantities shall not be exceeded	X	X			
0106	DeviceAccessLocks - RecordItemRef	7.5.8.4	V_DeviceAccessLocks shall only be referenced via RecordItemRef.	X				
0107	DeviceAccessLocks - Features	7.5.8.5	A RecordItemRef with variableId="V_DeviceAccessLocks" is only allowed for those subindices which refer to an access lock that is supported, i.e. where the respective attribute in Features/SupportedAccessLocks is set to "true".	X				
0108	Features – DataStorage - DevSpecIDSU	7.5.1	If attribute Features/@dataStorage = "false" and Device specific ISDU variables with accessRights = "rw" are present, issue a warning: "Warning: Support of Data storage is highly recommended" Result: Warning	X		X	X	
0109	Features – DataStorage - StdIDSU	7.5.1	If attribute Features/@dataStorage = "false", all Standard Variables with accessRights = "rw" shall not be implemented. (StdVariableRef to V_ApplicationSpecificTag, V_DeviceAccessLocks, V_OffsetTime) Result: Error	X				
0110	Variable - StdVariableRef	7.5.4.1	For a StdVariableRef to V_DeviceAccessLocks, StdRecordItemRef elements may only be present for supported access locks. Supported means, that the corresponding attribute in Features/SupportedAccessLocks is set to "true".	X				

IODD_TC No	Rule name	Reference	Definition	M	L	W	H	V
0111	Variable - StdVariableRef-ProductID_1	7.5.4.1	If there is only one DeviceVariant present, the value of DeviceVariant/@productID shall be used as default value of V_ProductID.	X				
0112	Variable - StdVariableRef-ProductID_2	7.5.4.1	For references to V_ProductID the attribute 'defaultValue' shall not be specified if more than one DeviceVariant is defined.	X				
0113	Menu - VariableRef	7.5.8.4	For VariableRef, the attribute accessRightRestriction shall not be set to "wo".	X				
0114	Menu - RecordItemRef	7.5.8.5	For RecordItemRef, the attribute accessRightRestriction shall not be set to "wo".	X				
0115	Variables - ComplexDatatypes-accessRights	7.5.4.3	For records and arrays the attribute accessRights="wo" is not permitted.	X				
0116	Variables – Record-RecordItem	7.5.3.2.3	The attribute accessRightRestriction shall not be set to "wo". An exception to this rule is the V_DirectParameters_1.	X				
0117	PhysicalLayer - mSequenceCapability	[1], A.2.6	Checker shall warn if M-Sequence value leads to Type_0. Checker shall use following text: "Warning – It is highly recommended for Devices not to use TYPE_0 in Preoperate or Operate states"	X		X	X	
0118	VariableRef – Button - ActionStartedMessage	7.5.8.4	ActionStartedMessages shall be omitted for buttons referencing StdSingleValueRefs [@id="128" or "129" or "130" or "131"].	X	X			
0119	VariableRef – Button - Description	7.5.8.4	For button values described as StdSingleValueRefs [@id="128" or "129" or "130" or "131"], the Description element is mandatory. The definition in template IODD-SystemCommandDefinitions_V113.xml shall be used. Checker shall check IODD for correct content of button description within each supported language.	X	X			
0120	SupportedAccessLocks - Parameter	7.5.1 [1] B.2.4	It is highly recommended, that Features/SupportedAccessLocks/@parameter is set to "false".	X		X	X	
0121	SupportedAccessLocks - dataStorage	7.5.1 [1] B.2.4	It is highly recommended, that Features/SupportedAccessLocks/@dataStorage is set to "false".	X		X	X	
0122	RecordItemRef – DeviceAccessLocks - Parameter	7.5.1	IODD Menu shall not contain reference to V_DeviceAccessLocks subindex="1"	X				
0123	RecordItemRef – DeviceAccessLocks - DataStorage	7.5.1	IODD Menu shall not contain reference to V_DeviceAccessLocks subindex="2"	X				

© Copyright by:

IO-Link **Community**

Haid-und-Neu-Str. 7

76131 Karlsruhe

Germany

Phone: +49 (0) 721 / 96 58 590

Fax: +49 (0) 721 / 96 58 589

e-mail: info@io-link.com

<http://www.io-link.com/>



IO-Link