

DIN EN 62541-100

ICS 35.200; 35.240.50

Einsprüche bis 2014-09-25

Entwurf

**OPC Unified Architecture Specification –
Teil 100: Schnittstelle zu Geräten
(IEC 65E/372/CDV:2014);
Englische Fassung FprEN 62541-100:2014**

OPC Unified Architecture Specification –
Part 100: Device Interface
(IEC 65E/372/CDV:2014);
English version FprEN 62541-100:2014

Spécification de l'Architecture Unifiée OPC –
Partie 100: Interface d'appareils
(CEI 65E/372/CDV:2014);
Version anglaise FprEN 62541-100:2014

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2014-07-25 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 61 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die Englische Fassung des europäischen Dokuments FprEN 62541-100:2014 „OPC Unified Architecture Specification – Teil 100: Schnittstelle zu Geräten“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 65E/372/CDV:2014 „OPC Unified Architecture Specification – Part 100: Device Interface“ unverändert in den Entwurf FprEN 62541-100:2014 übernommen worden.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 931 „Systemaspekte“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für den vorliegenden Normentwurf erfüllt.

Da sich die Benutzer des vorliegenden Normentwurfs der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 62541-100 veröffentlicht. Zu deren Abschnitt 3, Symbole und Abkürzungen, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Normentwurfs hinzugefügt. Dieser Normentwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. ISO [und/oder] IEC sind nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

Nationaler Anhang NA (informativ)

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC/TR 62541-1, IEC 62541-3 und IEC 62541-8 sowie die folgenden Begriffe.

3.1.1

Block

(en: **Block**)

in dieser Spezifikation bezieht sich die Benennung *Block* auf eine funktionale Gruppierungseinheit für *Parameter*. Sie könnte auf einen Funktionsblock (siehe [IEC 61131]) oder die Ressourcenparameter des Gerätes selbst abgebildet werden

3.1.2

BlockModus

(en: **BlockMode**)

der Blockmodus legt den Betriebsmodus für einen *Block* fest (Zielmodus, zulässige Modi, aktueller Modus und Normalmodus). Weitere Einzelheiten zu *Blockmodi* werden von Normungsorganisationen festgelegt

3.1.3

Kommunikationsprofil

(en: **Communication Profile**)

festgelegter Satz von Abbildungsregeln für eine eindeutige Interoperabilität zwischen Geräten bzw. Anwendungen

ANMERKUNG Beispiele für derartige Profile sind die „kabellosen Kommunikationsnetze und Kommunikationsprofile für WirelessHART“ in IEC 62591 und die Protokollabbildungen für OPC UA in Teil 6.

3.1.4

Verbindungspunkt

(en: **Connection Point**)

logische Darstellung der Schnittstelle zwischen einem *Gerät* und einem *Netz*

3.1.5

Gerät

(en: **Device**)

unabhängige physische Einheit, die fähig ist, eine oder mehrere spezifische Funktionen in einem speziellen Kontext auszuführen und die durch ihre Schnittstellen abgegrenzt ist, siehe IEC 61499-1, Ausgabe 1.0 (Januar 2005)

ANMERKUNG Typische Geräte stellen Funktionen zur Abtastung, Schaltung, Kommunikation und Steuerung bereit. Beispiele sind Messumformer, Ventilsteuergeräte, Getriebe, Motorregler, SPS und Kommunikations-Gateways.

3.1.6

Geräteintegrationshost

(en: **Device Integration Host**)

Server, der für die Integration mehrerer *Geräte* in ein Automatisierungssystem sorgt

3.1.7

Gerätetopologie

(en: **Device Topology**)

Anordnung von *Netzen* und *Geräten*, die eine Kommunikationstopologie bildet

3.1.8**Feldbus****(en: Fieldbus)**

Kommunikationssystem basierend auf serieller Datenübertragung und eingesetzt in der industriellen Automatisierung oder in Prozessleitsystemen, siehe IEC 61784-3, Ausgabe 2.0 (Juni 2010)

ANMERKUNG Bezeichnet den Kommunikationsbus, der von einem Gerät verwendet wird.

3.1.9**Parameter****(en: Parameter)**

ein Parameter ist eine Variable des Gerätes, die für Konfigurations-, Überwachungs- und Steuerungszwecke verwendet werden kann. Im Informationsmodell ist er gleichbedeutend mit einer OPC UA-*DatenVariable*

3.1.10**Netz****(en: Network)**

stellt ein Mittel zur Kommunikation mit einem bestimmten Protokoll dar

CONTENTS

1		
2		
3		
4		
5	1	Scope..... 1
6	2	Reference documents..... 1
7	3	Terms, definitions, and abbreviations 2
8	3.1	Terms and definitions 2
9	3.2	Abbreviations and symbols 3
10	3.3	Used data types 3
11	4	Fundamentals..... 4
12	4.1	OPC UA 4
13	4.2	Conventions used in this document 4
14	4.2.1	Conventions for Node descriptions 4
15	4.2.2	NodeIds and BrowseNames..... 6
16	5	Device Model 7
17	5.1	General 7
18	5.2	TopologyElementType 8
19	5.3	FunctionalGroupType 9
20	5.4	Identification Functional Group 12
21	5.5	UIElement Type..... 12
22	5.6	DeviceType 13
23	5.7	Device Support Information 15
24	5.7.1	Device Type Image..... 16
25	5.7.2	Documentation 16
26	5.7.3	Protocol Support Files 16
27	5.7.4	Images 17
28	5.8	DeviceSet Entry Point 17
29	5.9	ProtocolType 19
30	5.10	BlockType 20
31	5.11	Configurable Components 21
32	5.11.1	General Pattern 21
33	5.11.2	ConfigurableObjectType 22
34	6	Device Communication Model..... 23
35	6.1	General 23
36	6.2	Network..... 24
37	6.3	ConnectionPoint..... 25
38	6.4	ConnectsTo and ConnectsToParent ReferenceTypes 26
39	6.5	NetworkSet Object (Mandatory)..... 28
40	7	Device Integration Host Model..... 29
41	7.1	General 29
42	7.2	DeviceTopology Object 30
43	7.3	Online / Offline 31
44	7.3.1	General 31
45	7.3.2	IsOnline ReferenceType 32
46	8	AddIn Capabilities 34
47	8.1	Overview 34
48	8.2	Offline-Online Data Transfer..... 34

49	8.2.1	Definition	34
50	8.2.2	TransferServices Type.....	35
51	8.2.3	TransferServices Object	36
52	8.2.4	TransferToDevice Method.....	36
53	8.2.5	TransferFromDevice Method.....	37
54	8.2.6	FetchTransferResultData Method	37
55	8.3	Locking	40
56	8.3.1	Overview	40
57	8.3.2	LockingServices Type.....	41
58	8.3.3	LockingServices Object	42
59	8.3.4	MaxInactiveLockTime Property	42
60	8.3.5	InitLock Method	42
61	8.3.6	ExitLock Method	43
62	8.3.7	RenewLock Method	44
63	8.3.8	BreakLock Method	44
64	9	Specialized Topology Elements	45
65	9.1	General	45
66	9.2	Block Devices (BlockOriented DeviceType)	45
67	9.3	Modular Devices.....	46
68	10	Profiles.....	47
69	10.1.1	Device Server Facets	47
70	10.1.2	Device Client Facets.....	48
71	Annex A (normative):	Namespace and Mappings	50
72			

FIGURES

73		
74		
75	Figure 1 – <i>Device Model</i> Overview	7
76	Figure 2 – Components of the <i>TopologyElementType</i>	8
77	Figure 3 – <i>FunctionalGroupType</i>	10
78	Figure 4 – Analyser Device use for FunctionalGroups ([UA Companion ADI])	11
79	Figure 5 – PLCopen use for FunctionalGroups ([UA Companion PLCopen])	11
80	Figure 6 – Example of an <i>Identification Functional Group</i>	12
81	Figure 7 – <i>DeviceType</i>	13
82	Figure 8 – Integration of support information within a <i>DeviceType</i>	15
83	Figure 9 – Standard Entry Point for <i>Devices</i>	18
84	Figure 10 – Example of a <i>ProtocolType</i> Hierarchy with instances that represent	
85	specific <i>Communication Profiles</i>	19
86	Figure 11 – <i>BlockType</i> Hierarchy	20
87	Figure 12 – <i>Configurable Component Pattern</i>	21
88	Figure 13 – <i>ConfigurableObjectType</i>	22
89	Figure 14 – Initial Example of a Communication Topology	23
90	Figure 15 – <i>NetworkType</i>	24
91	Figure 16 – Example of <i>ConnectionPointType</i> Hierarchy	25
92	Figure 17 – <i>ConnectionPointType</i>	25
93	Figure 18 – <i>ConnectionPoint</i> Usage	26
94	Figure 19 – Type Hierarchy for <i>ConnectsTo</i> and <i>ConnectsToParent</i> References	27
95	Figure 20 – Example with <i>ConnectsTo</i> and <i>ConnectsToParent</i> References	27
96	Figure 21 – Example of an Automation System	29
97	Figure 22 – Example of a Device Topology	30
98	Figure 23 – Online Component for Access to Device Data	31
99	Figure 24 – Type Hierarchy for <i>IsOnline</i> Reference	32
100	Figure 25 – <i>TransferServicesType</i>	35
101	Figure 26 – Transfer Services	36
102	Figure 27 – <i>LockingServicesType</i>	41
103	Figure 28 – Locking Services	42
104	Figure 29 – Block-Oriented Device Structure Example	45
105	Figure 30 – Modular Device Structure Example	46

TABLES

107		
108	Table 1 –DataTypes defined in IEC 62541-3	3
109	Table 2 – Type Definition Table	5
110	Table 3 – Examples of DataTypes	5
111	Table 4 – <i>TopologyElementType</i> Definition	9
112	Table 5 – <i>ParameterSet</i> Definition	9
113	Table 6 – <i>MethodSet</i> Definition	9
114	Table 7 – <i>FunctionalGroupType</i> Definition	10
115	Table 8 – <i>UIElementType</i> Definition	13
116	Table 9 – <i>DeviceType</i> Definition	13

117	Table 10 – DeviceHealth Values	15
118	Table 11 – DeviceTypeImage Definition	16
119	Table 12 – Documentation Definition	16
120	Table 13 – ProtocolSupport Definition	17
121	Table 14 – ImageSet Definition	17
122	Table 15 – DeviceSet definition	18
123	Table 16 – <i>ProtocolType</i> Definition	19
124	Table 17 – <i>BlockType</i> Definition.....	20
125	Table 18 – <i>ConfigurableObjectType</i> Definition	22
126	Table 19 – NetworkType Definition	24
127	Table 20 – ConnectionPointType Definition.....	25
128	Table 21 – ConnectsTo ReferenceType	27
129	Table 22 – ConnectsToParent ReferenceType	27
130	Table 23 – NetworkSet definition	28
131	Table 24 – DeviceTopology definition	31
132	Table 25 – IsOnline ReferenceType	33
133	Table 26 – TransferServicesType Definition.....	35
134	Table 27 – TransferToDevice Method Arguments.....	37
135	Table 28 – TransferToDevice Method AddressSpace Definition	37
136	Table 29 – TransferFromDevice Method Arguments.....	37
137	Table 30 – TransferFromDevice Method AddressSpace Definition	37
138	Table 31 –FetchTransferResultData Method Arguments	38
139	Table 32 – FetchTransferResultData Method AddressSpace Definition	39
140	Table 33 – FetchResult DataType Structure.....	39
141	Table 34 – TransferResultError DataType Structure.....	39
142	Table 35 – TransferResultData DataType Structure	40
143	Table 34 – LockingServicesType Definition.....	41
144	Table 35 – MaxInactiveLockTime Property Definition	42
145	Table 36 – InitLock Method Arguments	43
146	Table 37 – InitLock Method AddressSpace Definition.....	43
147	Table 38 – ExitLock Method Arguments	44
148	Table 39 – ExitLock Method AddressSpace Definition.....	44
149	Table 40 – RenewLock Method Arguments	44
150	Table 41 – RenewLock Method AddressSpace Definition	44
151	Table 42 – BreakLock Method Arguments	45
152	Table 43 – BreakLock Method AddressSpace Definition	45
153	Table 44 – <i>BaseDevice_Server_Facet</i> Definition.....	47
154	Table 45 – <i>DeviceIdentification_Server_Facet</i> Definition.....	47
155	Table 46 – <i>BlockDevice_Server_Facet</i> Definition	47
156	Table 47 – Locking_Server_Facet Definition	47
157	Table 48 – <i>DeviceCommunication_Server_Facet</i> Definition	48
158	Table 49 – <i>DeviceIntegrationHost_Server_Facet</i> Definition.....	48
159	Table 50 – <i>BaseDevice_Client_Facet</i> Definition.....	48

160 Table 51 – *DeviceIdentification_Client_Facet* Definition 48

161 Table 52 – *BlockDevice_Client_Facet* Definition 48

162 Table 53 – *Locking_Client_Facet* Definition 49

163 Table 54 – *DeviceCommunication_Client_Facet* Definition 49

164 Table 55 – *DeviceIntegrationHost_Client_Facet* Definition 49

165

166

INTERNATIONAL ELECTROTECHNICAL COMMISSION

OPC Unified Architecture –

OPC UA for Devices

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62541-100 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this standard is based on the following documents:

FDIS	Report on voting
65E/XX/FDIS	65E/XX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 62541 series, published under the general title *OPC Unified Architecture*, can be found on the IEC website.

220 The committee has decided that the contents of this publication will remain unchanged until
221 the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data
222 related to the specific publication. At this date, the publication will be

- 223 • reconfirmed,
224 • withdrawn,
225 • replaced by a revised edition, or
226 • amended.

227

228 The National Committees are requested to note that for this publication the stability date
229 is 2017.

230 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE
231 DELETED AT THE PUBLICATION STAGE.

232

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

233

OPC Unified Architecture

OPC Unified Architecture for Devices

1 Scope

In today's automation systems devices from many different manufacturers have to be integrated resulting in effort for installation, version management and device operation. This challenge can be faced best with an open and standardized device model.

This specification is an extension of the overall OPC Unified Architecture specification series and defines the information model associated with *Devices*. This specification describes three models which build upon each other as follows:

- The (base) Device Model is intended to provide a unified view of devices irrespective of the underlying device protocols.
- The Device Communication Model adds Network and Connection information elements so that communication topologies can be created.
- The Device Integration Host Model finally adds additional elements and rules required for host systems to manage integration for a complete system. It allows reflecting the topology of the automation system with the devices as well as the connecting communication networks.

2 Reference documents

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

- IEC/TR 62541-1, *OPC Unified Architecture – Part 1: Concepts*
IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*
IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

IEC 62541-5, *OPC Unified Architecture – Part 5: Information Model*
IEC 62541-6, *OPC Unified Architecture – Part 6: Mappings*
IEC 62541-7, *OPC Unified Architecture – Part 7: Profiles*
IEC 62541-8, *OPC Unified Architecture – Part 8: Data Access*

IEC 62769, *Field Device Integration (FDI)*

NAMUR Recommendation NE107: Self-monitoring and diagnosis of field devices

Additional external references used to provide information model suggestions for this document:

- [IEC 61131] IEC standard for Programmable Logic Controllers (PLCs).
[UA Companion ADI] OPC UA Companion Specification for Analyser Devices.

271 [UA Companion PLCopen] OPC UA Companion Specification for PLCopen

272 **3 Terms, definitions, and abbreviations**

273 **3.1 Terms and definitions**

274 For the purposes of this document, the terms and definitions given in IEC/TR 62541-1, IEC 62541-
275 3, and IEC 62541-8 as well as the following apply.

276 **3.1.1**

277 **Block**

278 in this specification the term Block refers to a functional *Parameter* grouping entity. It could map to
279 a function block (see [IEC 61131]) or to the resource parameters of the device itself.

280 **3.1.2**

281 **BlockMode**

282 Block Mode specifies the mode of operation (target mode, permitted modes, actual mode, and
283 normal mode) for a *Block*. Further details about *Block* modes are defined by standard organisations.

284 **3.1.3**

285 **Communication Profile**

286 a fixed set of mapping rules to allow unambiguous interoperability between Devices or Applications,
287 respectively

288 NOTE: Examples of such profiles are the “Wireless communication network and communication profiles for
289 WirelessHART” in IEC 62591 and the Protocol Mappings for OPC UA in IEC 62541-6.

290 **3.1.4**

291 **Connection Point**

292 logical representation of the interface between a *Device* and a *Network*

293 **3.1.5**

294 **Device**

295 independent physical entity capable of performing one or more specified functions in a particular
296 context and delimited by its interfaces, see IEC 61499-1, ed. 1.0 (2005-01)

297 NOTE: Typical devices provide sensing, actuating, communication, and/or control functionality. Examples include
298 transmitters, valve controllers, drives, motor controllers, PLCs, and communication gateways.

299 **3.1.6**

300 **Device Integration Host**

301 a *Server* that manages integration of multiple *Devices* in an automation system

302 **3.1.7**

303 **Device Topology**

304 arrangement of *Networks* and *Devices* that constitute a communication topology

305 **3.1.8**

306 **Fieldbus**

307 communication system based on serial data transfer and used in industrial automation or process
308 control applications, see IEC 61784-3, ed. 2.0 (2010-06)

309 NOTE: Designates the communication bus used by a *Device*.

310 **3.1.9**

311 **Parameter**

312 a parameter is a variable of the *Device* that can be used for configuration, monitoring or control
313 purposes. In the information model it is synonymous to an OPC UA *DataVariable*.

314 **3.1.10**
315 **Network**
316 represents a means used to communicate with one specific protocol

317

318 **3.2 Abbreviations and symbols**

319 ADI Analyser Device Integration
320 CP Communication Processor (hardware module)
321 CPU Central Processing Unit (of a *Device*)
322 DA Data Access
323 DI Device Integration (the short name for this specification)
324 UA Unified Architecture
325 UML Unified Modelling Language
326 XML Extensible Mark-up Language
327

328 **3.3 Used data types**

329 Table 1 describes the *DataTypes* that are used throughout this document.

330 **Table 1 –DataTypes defined in IEC 62541-3**

Parameter Type
Argument
Boolean
Duration
LocalizedText
String
Int32

331
332
333

334 4 Fundamentals

335 4.1 OPC UA

336 The main use case for OPC standards is the online data exchange between devices and HMI or
337 SCADA systems using Data Access functionality. In this use case the device data is provided by an
338 OPC server and is consumed by an OPC client integrated into the HMI or SCADA system. OPC DA
339 provides functionality to browse through a hierarchical namespaces containing data items and to
340 read, write and to monitor these items for data changes. The classic OPC standards are based on
341 Microsoft COM/DCOM technology for the communication between software components from
342 different vendors. Therefore classic OPC server and clients are restricted to Windows PC based
343 automation systems.

344 OPC UA incorporates all features of classic OPC standards like OPC DA, A&E and HDA but defines
345 platform independent communication mechanisms and generic, extensible and object-oriented
346 modelling capabilities for the information a system wants to expose.

347 The OPC UA network communication part defines different mechanisms optimized for different use
348 cases. The current version of OPC UA is defining an optimized binary protocol for high performance
349 intranet communication as well as Web Services. It allows adding new protocols in the future.
350 Features like security, access control and reliability are directly built into the transport mechanisms.
351 Based on the platform independence of the protocols, OPC UA servers and clients can be directly
352 integrated into devices and controllers.

353 The OPC UA *Information Model* provides a standard way for *Servers* to expose *Objects* to *Clients*.
354 *Objects* in OPC UA terms are composed of other *Objects*, *Variables* and *Methods*. OPC UA also
355 allows relationships to other *Objects* to be expressed.

356 The set of *Objects* and related information that an OPC UA *Server* makes available to *Clients* is
357 referred to as its *AddressSpace*. The elements of the OPC UA *Object Model* are represented in the
358 *AddressSpace* as a set of *Nodes* described by *Attributes* and interconnected by *References*. OPC
359 UA defines eight classes of *Nodes* to represent *AddressSpace* components. The classes are
360 *Object*, *Variable*, *Method*, *ObjectType*, *VariableType*, *DataType*, *ReferenceType* and *View*. Each
361 *NodeClass* has a defined set of *Attributes*.

362 This specification makes use of almost all OPC UA *NodeClasses*.

363 *Objects* are used to represent real-world entities such as *Devices* and (communication) *Networks* as
364 well as software entities such as *Blocks*. An *Object* is associated to a corresponding *ObjectType*
365 that provides definitions for that *Object*.

366 *Variables* are used to represent values. Two categories of *Variables* are defined, *Properties* and
367 *DataVariables*.

368 *Properties* are *Server*-defined characteristics of *Objects*, *DataVariables* and other *Nodes*.
369 *Properties* are not allowed to have *Properties* defined for them. Examples for *Properties* of *Objects*
370 are the device serial number and the block tag.

371 *DataVariables* represent the contents of an *Object*. *DataVariables* may have component
372 *DataVariables*. This is typically used by *Servers* to expose individual elements of arrays and
373 structures. This specification uses *DataVariables* to represent the *Parameters* of both *Blocks* and
374 *Devices*.

375 4.2 Conventions used in this document

376 4.2.1 Conventions for Node descriptions

377 *Node* definitions are specified using tables (See Table 2)

Table 2 – Type Definition Table

Attribute	Value				
Attribute name	Attribute value. If it is an optional Attribute that is not set “--” will be used.				
References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
<i>ReferenceType</i> name	<i>NodeClass</i> of the <i>TargetNode</i> .	<i>BrowseName</i> of the target <i>Node</i> . If the <i>Reference</i> is to be instantiated by the server, then the value of the target <i>Node</i> ’s <i>BrowseName</i> is “--”.	<i>Attributes</i> of the referenced <i>Node</i> , only applicable for <i>Variables</i> and <i>Objects</i> .		Referenced <i>ModellingRule</i> of the referenced <i>Object</i> .
Notes – Notes referencing footnotes of the table content.					

Attributes are defined by providing the Attribute name and a value, or a description of the value.

References are defined by providing the ReferenceType name, the BrowseName of the TargetNode and its NodeClass.

- If the TargetNode is a component of the Node being defined in the table the Attributes of the composed Node are defined in the same row of the table..
- The DataType is only specified for Variables; “[<number>]” indicates a single-dimensional array, for multi-dimensional arrays the expression is repeated for each dimension (e.g. [2][3] for a two-dimensional array). For all arrays the ArrayDimensions is set as identified by <number> values. If no <number> is set, the corresponding dimension is set to 0, indicating an unknown size. If no number is provided at all the ArrayDimensions can be omitted. If no brackets are provided, it identifies a scalar DataType and the ValueRank is set to the corresponding value (see IEC 62541-3). In addition, ArrayDimensions is set to null or is omitted. If it can be Any or ScalarOrOneDimension, the value is put into “{<value>}”, so either “{Any}” or “{ScalarOrOneDimension}” and the ValueRank is set to the corresponding value (see IEC 62541-3) and the ArrayDimensions is set to null or is omitted. In Table 3 examples are given.

Table 3 – Examples of DataTypes

Notation	Data-Type	Value-Rank	Array-Dimensions	Description
Int32	Int32	-1	omitted or NULL	A scalar Int32
Int32[]	Int32	1	omitted or {0}	Single-dimensional array of Int32 with an unknown size
Int32[][]	Int32	2	omitted or {0,0}	Two-dimensional array of Int32 with unknown sizes for both dimensions
Int32[3][]	Int32	2	{3,0}	Two-dimensional array of Int32 with a size of 3 for the first dimension and an unknown size for the second dimension
Int32[5][3]	Int32	2	{5,3}	Two-dimensional array of Int32 with a size of 5 for the first dimension and a size of 3 for the second dimension
Int32{Any}	Int32	-2	omitted or NULL	An Int32 where it is unknown if it is scalar or array with any number of dimensions
Int32{ScalarOrOneDimension}	Int32	-3	omitted or NULL	An Int32 where it is either a single-dimensional array or a scalar

- 398 • The *TypeDefinition* is specified for *Objects* and *Variables*.
- 399 • The *TypeDefinition* column specifies a *NodeId* of a *TypeDefinitionNode*, i.e. the specified
400 *Node* points with a *HasTypeDefinition Reference* to the corresponding *TypeDefinitionNode*.
401 The symbolic name of the *NodeId* is used in the table.
- 402 • The *ModellingRule* of the referenced component is provided by specifying the symbolic
403 name of the rule in the *ModellingRule* column. In the *AddressSpace*, the *Node* shall use a
404 *HasModellingRule Reference* to point to the corresponding *ModellingRule Object*.

405 If the *NodeId* of a *DataType* is provided, the symbolic name of the *Node* representing the *DataType*
406 shall be used.

407 If no components are provided, the *DataType*, *TypeDefinition* and *ModellingRule* columns may be
408 omitted and only a *Comment* column is introduced to point to the *Node* definition.

409 Components of *Nodes* can be complex, i.e. containing components by themselves. The
410 *TypeDefinition*, *NodeClass*, *DataType* and *ModellingRule* can be derived from the type definitions,
411 and the symbolic name can be created as defined in 4.2.2.1. Therefore those containing
412 components are not explicitly specified; they are implicitly specified by the type definitions.

413 4.2.2 NodeIds and BrowseNames

414 4.2.2.1 NodeIds

415 The *NodeIds* of all *Nodes* described in this document are only symbolic names. Annex A defines the
416 actual *NodeIds*.

417 The symbolic name of each *Node* defined in this document is its *BrowseName*, or, when it is part of
418 another *Node*, the *BrowseName* of the other *Node*, a “.”, and the *BrowseName* of itself. In this case
419 “part of” means that the whole has a *HasProperty* or *HasComponent Reference* to its part. Since all
420 *Nodes* not being part of another *Node* have a unique name in this document, the symbolic name is
421 unique.

422 The namespace for this specification is defined in Annex A. The *NamespaceIndex* for all *NodeIds*
423 and *BrowseNames* defined in this specification is server specific and depends on the position of the
424 namespace URI in the server namespace table.

425 4.2.2.2 BrowseNames

426 The text part of the *BrowseNames* for all *Nodes* defined in this specification is specified in the
427 tables defining the *Nodes*. The *NamespaceIndex* for all *BrowseNames* defined in this specification
428 is server specific and depends on the position of the namespace URI in the server namespace
429 table.

5 Device Model

5.1 General

Figure 1 depicts the main *ObjectTypes* of base *Device* model and their relationship. The drawing is not intended to be complete. For the sake of simplicity only a few components and relations were captured so as to give a rough idea of the overall structure.

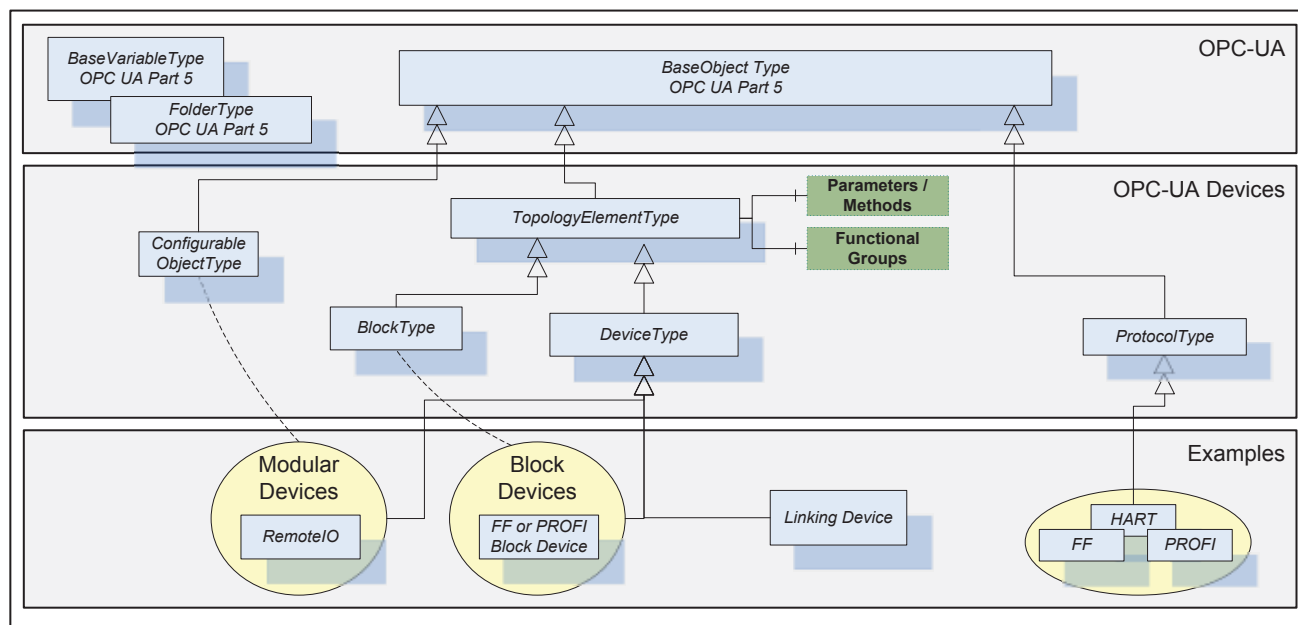


Figure 1 – Device Model Overview

The boxes in this drawing show the *ObjectTypes* used in this specification as well as some elements from other specifications that help understand some modelling decisions. The upper grey box shows the OPC UA core *ObjectTypes* from which the *TopologyElementType* and *ProtocolType* are derived. The grey box in the second level shows the main *ObjectTypes* that this specification introduces. The components of those *ObjectTypes* are illustrated only in an abstract way in this overall picture.

The grey box in the third level shows real-world examples as they will be used in products and plants. In general, such subtypes are defined by other organizations.

TopologyElementType is the base *ObjectType* for elements in a device topology. It introduces *Parameters* and *Methods*. This specification also defines a functional grouping concept to provide alternative viewpoints.

The *DeviceType* *ObjectType* provides a general type definition for any *Device*. *Devices* – in addition to *Parameters* and *Methods* – may support sub-devices and may support *Blocks*. *Blocks* are typically used as means to organise the functionality within a *Device*. Specific types of *Blocks* will be specified by the various field communication foundations.

A *ProtocolType* *ObjectType* represents a specific communication / *FieldBus* protocol implemented by a certain *TopologyElement*. Examples are *FFBusType*, *HARTBusType*, or *PROFIBUS/PROFINETType*.

The *ConfigurableObjectType* is used as a general means to create modular topology units. If needed an instance of this type will be added to the head object of the modular unit. Modular

459 *Devices*, for example, will use this *ObjectType* to organise their *Modules*. Block-oriented *Devices*
460 use it to expose and organise their *Blocks*.

461 5.2 TopologyElementType

462 This *ObjectType* defines the basic information components for all configurable elements in a device
463 topology. It introduces *ParameterSet*, *MethodSet*, and *FunctionalGroups*. Figure 2 shows the
464 *TopologyElementType*. It is formally defined in Table 4.

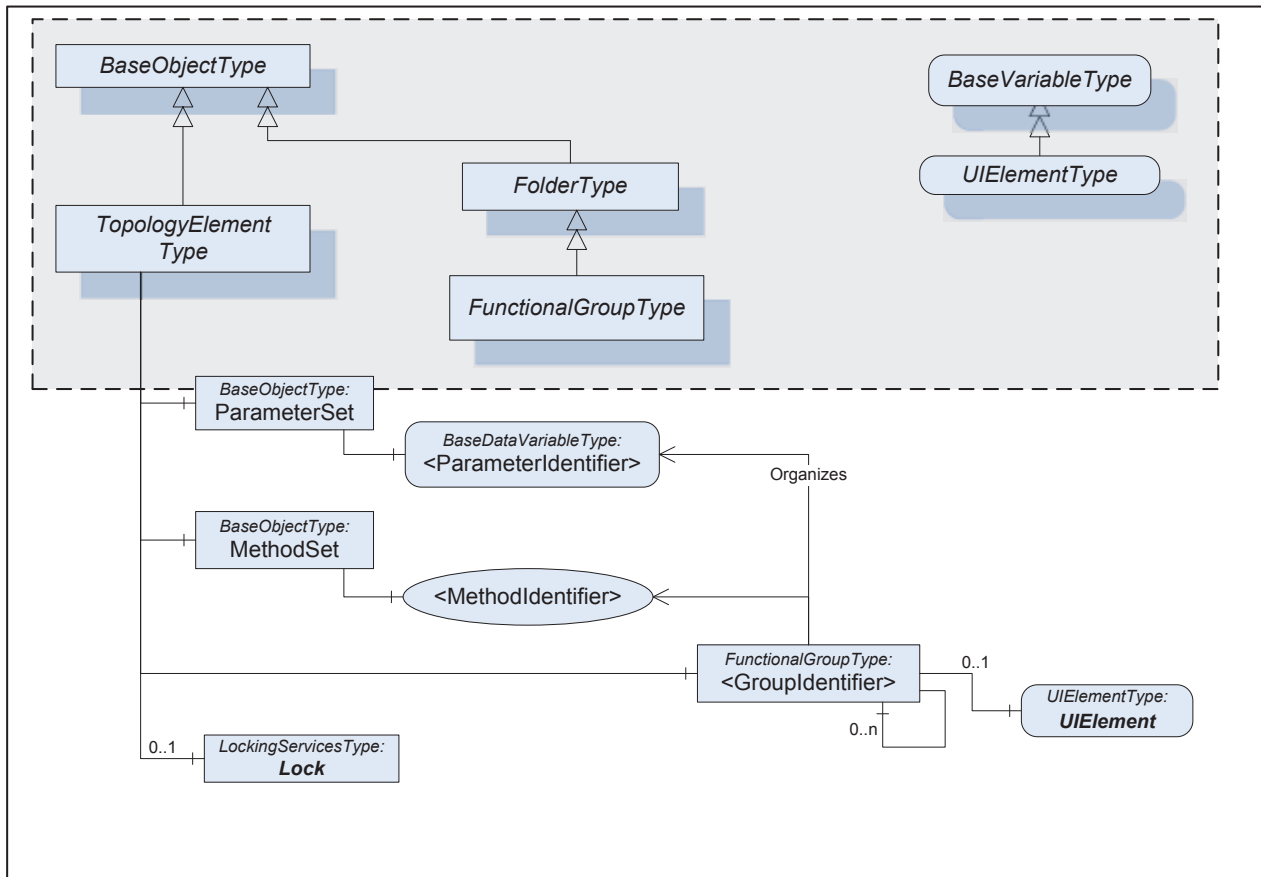


Figure 2 – Components of the *TopologyElementType*

465
466
467
468 All elements in a topology may have *Parameters* and *Methods*. If such an element has *Parameters*
469 they are kept in an *Object* called “*ParameterSet*” as a flat list of *Parameters*. If it has *Methods* they
470 are kept the same way in an *Object* called “*MethodSet*”. *FunctionalGroups* can be used to organise
471 the *Parameters* and *Methods* to reflect the structure of the *TopologyElement*. The same *Parameter*
472 or *Method* might be referenced from more than one *FunctionalGroup*. Each group may reference an
473 optional user interface for the represented functionality. *FunctionalGroups* are specified in clause
474 5.3.

475 Rule for all *Methods* organised in *FunctionalGroups*:

476 Since *Methods* are components of the *MethodSet* *Object*, *Clients* have to specify the *NodeId* of the
477 *MethodSet* instance in the *objectId* argument of the *Call Service*.

478 *Parameters* are modelled with OPC UA *DataVariable* nodes. A *Parameter* can be an instance of
479 *BaseDataVariableType* (defined in IEC 62541-5) or any sub-type defined by OPC UA or other
480 standard organisations. Some of the most common *VariableTypes* (*DataItem*, *AnalogItem*,
481 and *DiscreteItem*) are defined in IEC 62541-8. In this specification, the term *Parameter* is
482 used generically - regardless of *VariableType*.

483