

DIN EN IEC 62769-2



ICS 25.040.40; 35.240.50

Ersatz für
DIN EN 62769-2:2016-03
Siehe Anwendungsbeginn

**Feldgeräteintegration (FDI) –
Teil 2: FDI-Client
(IEC 62769-2:2021);
Englische Fassung EN IEC 62769-2:2021**

Field Device Integration (FDI) –
Part 2: FDI Client
(IEC 62769-2:2021);
English version EN IEC 62769-2:2021

Intégration des appareils de terrain (FDI) –
Partie 2: Client FDI
(IEC 62769-2:2021);
Version anglaise EN IEC 62769-2:2021

Gesamtumfang 154 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2021-03-12 angenommene Europäische Norm als DIN-Norm ist 2022-09-01.

Für DIN EN 62769-2:2016-03 besteht eine Übergangsfrist bis 2024-03-12.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 62769-2:2019-05

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Industrielle Kommunikation“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DKE (www.dke.de) und DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

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

Das IEC-Komitee hat entschieden, dass der Inhalt dieses Dokuments bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu diesem Dokument angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees das Dokument

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

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

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

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten 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 die vorliegende Norm erfüllt.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN IEC 62769-2 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt und Abschnitt 3, der die Begriffe, Abkürzungen und Konventionen festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für die meisten der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

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.

Änderungen

Gegenüber DIN EN 62769-2:2016-03 wurden folgende Änderungen vorgenommen:

- a) Ausführen von UIPs in einer Sandbox.

Frühere Ausgaben

DIN EN 62769-2: 2016-03

Nationaler Anhang NA (informativ)

Auszugsweise Übersetzung

Reihenfolge und Inhalt der folgenden Abschnitte sind identisch mit denen in den jeweiligen Abschnitten der Englischen Fassung.

1 Anwendungsbereich

Dieser Teil von IEC 62769 legt den FDI-Client fest. Die gesamtheitliche FDI-Architektur ist in Bild 1 dargestellt. Die im Anwendungsbereich dieses Dokuments liegenden Architekturkomponenten sind in diesem Bild hervorgehoben.

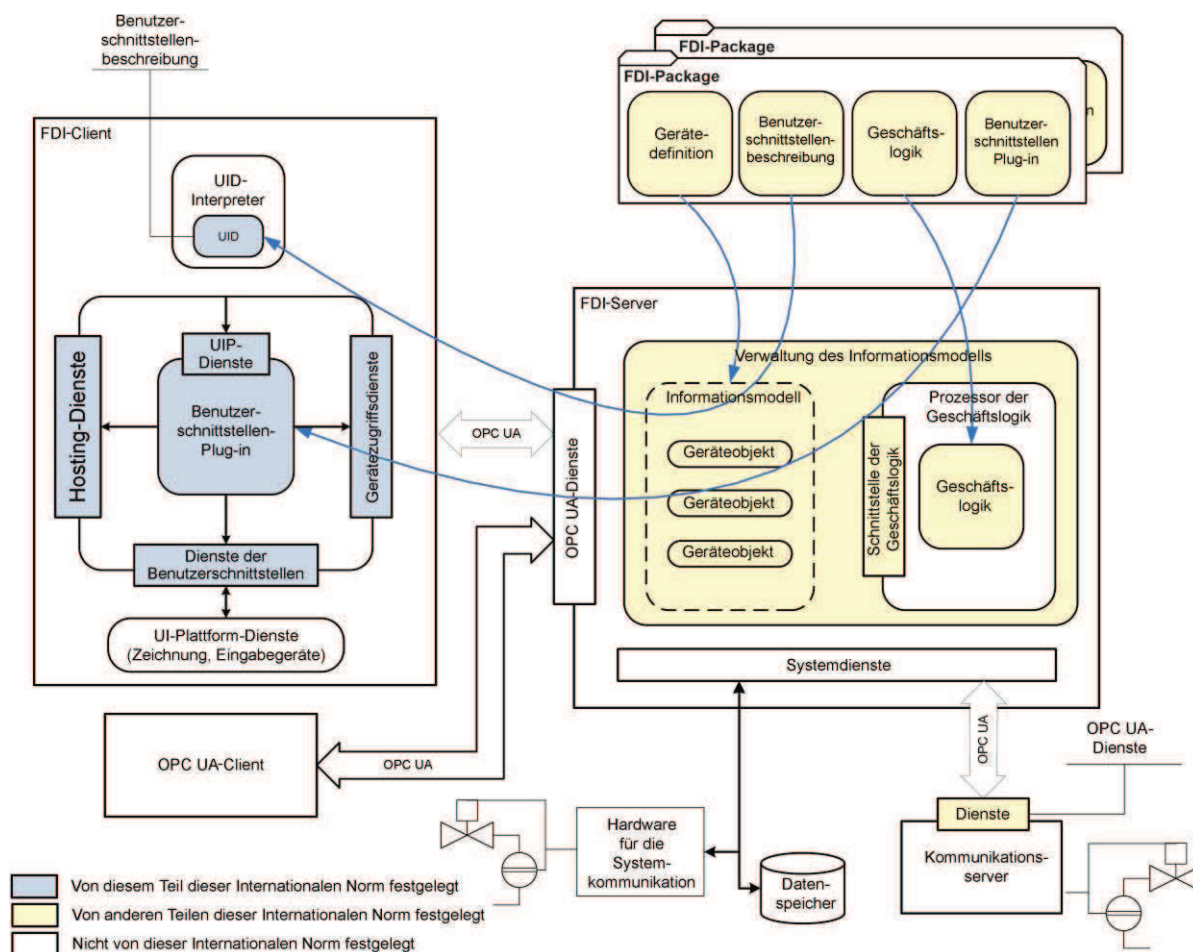


Bild 1 – FDI-Architektur

3 Begriffe, Abkürzungen und Konventionen

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Anwendung in der Normung unter den folgenden Adressen bereit:

- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>
- ISO Online Browsing Platform: verfügbar unter <http://www.iso.org/obp>

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 62769-1 und die folgenden Begriffe.

3.1.1 Begriffe für Dienste

3.1.1.1

Sperrdienste

en **Locking Services**

Menge von Diensten, über die der Zugriff auf ein Gerät gesteuert wird

3.1.1.2

Gerätemodelldienste

en **Device Model Services**

Untermenge der Gerätezugriffsdienste, über die ein UIP auf die Informationen eines Gerätes zugreifen kann

3.1.1.3

Direktzugriffsdienste

en **Direct Access Services**

Untermenge der Gerätezugriffsdienste, über die ein UIP direkt auf ein Gerät zugreifen kann

3.1.2 Begriffe für Gerätezugriffsdienste

3.1.2.1

Attribut

en **Attribute**

Informationselement eines Knotens

Anmerkung 1 zum Begriff: Einige Attribute sind für alle Knotenklassen (en: NodeClasses) vorhanden, andere sind für eine bestimmte NodeClass spezifisch.

Anmerkung 2 zum Begriff: Ersetzt die in IEC TR 62541-1 angegebene Definition.

3.1.2.2

Gerätemodell

en **Device Model**

Hierarchie von Knoten, die ein vorhandenes Gerät darstellen

3.1.2.3

Knoten

en **Node**

Element im Gerätemodell, das über die Gerätezugriffsdienste adressiert werden kann

Anmerkung 1 zum Begriff: Ersetzt die in IEC TR 62541-1 angegebene Definition.

3.1.2.4

Knotenklasse

en **NodeClass**

entweder ein Objekt oder eine Variable

Anmerkung 1 zum Begriff: Ersetzt die in IEC TR 62541-1 angegebene Definition.

3.1.2.5

Objekt

en **Object**

Instanz des Knotenklasse-Objektes

Anmerkung 1 zum Begriff: Ersetzt die in IEC 62769-1 angegebene Definition.

3.1.2.6

Variable

en **Variable**

Instanz der Knotenklasse-Variable

Anmerkung 1 zum Begriff: Ersetzt die in IEC 62769-1 angegebene Definition.

3.2 Abkürzungen

Für die Anwendung dieses Dokuments gelten die Abkürzungen nach IEC 62769-1 und die folgenden Abkürzungen.

UTC koordinierte Weltzeit (en: Coordinated Universal Time)

XML erweiterbare Auszeichnungssprache (en: Extensible Mark-up Language)

3.3 Konventionen

Die Konventionen für die Dienstdefinitionen sind die gleichen wie in IEC 62541-4.

Grunddatentypen werden in IEC 62541-3 festgelegt.

„Parameter“ ist immer ein Informationsmodellparameter. „parameter“ ist der allgemeine Gebrauch des Worts. Bei Uneindeutigkeit wird zusätzlicher Kontext angegeben.

EUROPEAN STANDARD

EN IEC 62769-2

NORME EUROPÉENNE

EUROPÄISCHE NORM

March 2021

ICS 25.040.40; 35.100.05

Supersedes EN 62769-2:2015 and all of its amendments and corrigenda (if any)

English version

**Field Device Integration (FDI) – Part 2: FDI Client
(IEC 62769-2:2021)**

Intégration des appareils de
terrain (FDI) – Partie 2: Client FDI
(IEC 62769-2:2021)

Feldgeräteintegration (FDI) – Teil 2: FDI-Client
(IEC 62769-2:2021)

This European Standard was approved by CENELEC on 2021-03-12. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brüssel

European foreword

The text of document 65E/759/FDIS, future edition 2 of IEC 62769-2, prepared by SC 65E "Devices and integration in enterprise systems" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN IEC 62769-2:2021.

The following dates are fixed:

- latest date by which the document has to be implemented at national level (dop) 2021-12-12
by publication of an identical national standard or by endorsement
- latest date by which the national standards conflicting with the document (dow) 2024-03-12
have to be withdrawn

This document supersedes EN 62769-2:2015 and all of its amendments and corrigenda (if any).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 62769-2:2021 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61804-3 NOTE Harmonized as EN IEC 61804-3.

IEC 61804-4 NOTE Harmonized as EN IEC 61804-4.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

NOTE 1 Where an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

Publication	Year	Title	EN/HD	Year
IEC 62443-3-3	2013	Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels	EN IEC 62443-3-3	2019
IEC 62541-3	–	OPC Unified Architecture – Part 3: Address Space Model	EN IEC 62541-3	–
IEC 62541-4	–	OPC Unified Architecture – Part 4: Services	EN IEC 62541-4	–
IEC 62769-1	–	Field Device Integration (FDI) – Part 1: Overview	EN 62769-1	–
IEC 62769-3	–	Field Device Integration (FDI) – Part 3: Server	EN 62769-3	–
IEC 62769-4	–	Field Device Integration (FDI) – Part 4: FDI Packages	EN 62769-4	–
IEC 62769-5	–	Field Device Integration (FDI) – Part 5: Information Model	EN 62769-5	–
IEC 62769-6	–	Field Device Integration (FDI) – Part 6: Technology Mapping	EN 62769-6	–
ISO 639	–	Code for the representation of names of languages	–	–
ISO 3166	–	Codes for the representation of names of countries	EN 23166	–
ISO/IEC 15948	–	Information technology – Computer graphics and image processing – Portable Network Graphics (PNG) – Functional specification	–	–
IETF RFC 3066	–	Tags for the Identification of Languages	–	–
XMLSchema-1	–	XML Schema: Structures (available at http://www.w3.org/TR/xmlschema-1/)	–	–
XMLSchema-2	–	XML Schema: Datatypes (available at http://www.w3.org/TR/xmlschema-2/)	–	–

CONTENTS

	Page
European foreword	8
Annex ZA (normative) Normative references to international publications with their corresponding European publications	9
INTRODUCTION	19
1 Scope	20
2 Normative references	20
3 Terms, definitions, abbreviated terms and conventions	21
3.1 Terms and definitions	21
3.1.1 Terms used for Services	21
3.1.2 Terms used for Device Access Services	21
3.2 Abbreviated terms	22
3.3 Conventions	22
4 Overview	22
5 FDI Client	24
5.1 Device Access Services	24
5.1.1 General	24
5.1.2 Device Model	24
5.1.3 Node model	26
5.1.4 Services	31
5.1.5 Base Property Services	35
5.1.6 Device Model Services	36
5.1.7 Locking Services	49
5.1.8 Direct Access Services	51
5.1.9 Data types	53
5.2 Hosting Services	59
5.2.1 General	59
5.2.2 Services	59
5.2.3 Parameter Type Definitions	70
6 UIP	72
6.1 UIP Services	72
6.1.1 Services	72
6.1.2 Parameter type definitions	75
6.2 UIP instantiation rules	77
6.3 UIP state machine	77
6.3.1 States	77
6.3.2 State transitions	78
6.4 UIP permissions and restrictions	78
6.4.1 Introduction	78

6.4.2	Access to local file system.....	78
6.4.3	Export/Import of files	79
6.4.4	Inter-Process Communication (IPC)	79
6.4.5	Open files based on MIME Type.....	79
6.4.6	Access to resources	79
6.5	UIP deployment	79
6.5.1	UIP downloads from FDI Server	79
6.5.2	UIP management on FDI Client.....	81
7	Actions.....	81
7.1	General.....	81
7.2	Sequence diagram	82
7.3	FDI Action schema definition	85
8	User Interface Description (UID).....	86
8.1	Overview.....	86
8.2	UID execution	88
	Annex A (normative) XML schema	91
A.1	General.....	91
A.2	AbortRequestT.....	91
A.3	AccessT.....	91
A.4	AcknowledgementRequestT.....	92
A.5	ActionListT	92
A.6	AbortingNotificationT	92
A.7	ActionRequestT	93
A.8	ActionResponseT	94
A.9	ActionT	95
A.10	AxisListT	95
A.11	AxisT	96
A.12	BitEnumerationItemListT	97
A.13	BitEnumerationItemT.....	97
A.14	ButtonListT.....	97
A.15	ChartT.....	98
A.16	ChartTypeT.....	99
A.17	ColorNameT	100
A.18	ColorT.....	101
A.19	ColorValueT	101
A.20	ColumnBreakT	101
A.21	DateTimeDataT.....	101
A.22	DelayMessageRequestT	102
A.23	DiagramLineT	103

A.24	EnumerationItemListT	104
A.25	EnumerationItemT	104
A.26	FormatSpecifierT	104
A.27	GraphT	105
A.28	GridT	105
A.29	HandlingT	106
A.30	ImageT	106
A.31	InfoRequestT	107
A.32	InputRequestT	108
A.33	InputResponseT	108
A.34	InputValueT	109
A.35	InputValueTypeT.....	109
A.36	LabelHelpT	110
A.37	LabelT	110
A.38	LineTypeT.....	111
A.39	MenuT	112
A.40	MenuReferenceT	113
A.41	MenuStyleT	114
A.42	NumericDataT.....	114
A.43	NumericTemplateT	115
A.44	OptionListT	115
A.45	OrientationT	116
A.46	ParameterInputRequestT	116
A.47	ParameterListT	117
A.48	ParameterT.....	117
A.49	PluginT	119
A.50	RangeListT	119
A.51	RangeT.....	120
A.52	ResponseT	120
A.53	RowBreakT.....	120
A.54	ScalingT	120
A.55	SelectionRequestT.....	121
A.56	SelectionResponseT	121
A.57	SeparatorT.....	122
A.58	SizeT	122
A.59	ParameterClassT	123
A.60	ActionClassT.....	124
A.61	SourceListT	126
A.62	SourceT.....	127

A.63	StringDataT	127
A.64	StringTemplateT	128
A.65	StringOptionListT	128
A.66	StringOptionT	129
A.67	StringT	129
A.68	TimeScaleT	130
A.69	UidLayoutInformation	130
A.70	UidRequestT	131
A.71	UidResponseT	131
A.72	UiElementSizeableT	132
A.73	UiElementT	132
A.74	UiTemplateT	133
A.75	VariantT	134
A.76	VariantOptionListT	134
A.77	VariantOptionT	135
A.78	VectorListT	135
A.79	VectorT	136
A.80	WaveformListT	136
A.81	WaveformT	137
A.82	WaveformTypeT	137
A.83	WaveformTypeHorizontalT	137
A.84	WaveformTypeVerticalT	138
A.85	WaveformTypeYTT	138
A.86	WaveformTypeXYT	139
A.87	WaveformKeyPointListT	140
A.88	WaveformVectorT	140
A.89	WaveformVectorElementListT	141
A.90	WaveformVectorElementT	141
Annex B (informative) Action example		143
Annex C (informative) Typical FDI Client use cases		152
C.1	General	152
C.2	Bulk operations	152
C.3	Progress bar support	152
Bibliography		154
Figures		
Figure 1 – FDI architecture diagram		20
Figure 2 – Overall structure of a Device		25
Figure 3 – Structure of Blocks		25
Figure 4 – Device Model NodeClasses		26

Figure 5 – Example: Variable hierarchy representing a RECORD.....	29
Figure 6 – Variable hierarchy representing a VALUE_ARRAY of RECORDs	30
Figure 7 – UIP state machine.....	77
Figure 8 – FDI Action sequence diagram	83
Figure 9 – User Interface Descriptions	87
Figure 10 – User Interface Description sequence diagram.....	89
Figure B.1 – Action example (step 1)	146
Figure B.2 – Action example (step 2)	147
Figure B.3 – Action example (step 3)	148
Figure B.4 – Action example (step 4)	149
Figure B.5 – Action example (step 5)	150
Figure B.6 – Action example (step 6)	151
Figure C.1 – Progress bar support	153

Tables

Table 1 – BaseNodeClass Attributes.....	26
Table 2 – Object NodeClass Attributes.....	26
Table 3 – Variable NodeClass Attributes.....	27
Table 4 – Parsing of the initial bytes.....	29
Table 5 – Service Definition Table.....	31
Table 6 – StatusCode Bit Assignments	33
Table 7 – DataValue InfoBits.....	33
Table 8 – Service result codes	34
Table 9 – Operation level result codes	34
Table 10 – GetDeviceAccessInterfaceVersion Service parameters.....	36
Table 11 – GetOnlineAccessAvailability Service parameters	36
Table 12 – Browse Service parameters.....	37
Table 13 – CancelBrowse Service parameters.....	38
Table 14 – Read Service parameters	39
Table 15 – Read Service result codes.....	39
Table 16 – Read operation result codes	40
Table 17 – CancelRead Service parameters	41
Table 18 – Write Service parameters	42
Table 19 – Write operation result codes	42
Table 20 – CancelWrite Service parameters	43
Table 21 – CreateSubscription Service parameters.....	44
Table 22 – CreateSubscription Service result codes.....	44
Table 23 – Subscribe Service parameters.....	45
Table 24 – Subscribe operation result codes.....	47
Table 25 – Unsubscribe Service Parameters.....	47

Table 26 – Unsubscribe operation result codes	48
Table 27 – DeleteSubscription Service parameters	48
Table 28 – DataChangeCallback Service parameters.....	49
Table 29 – DataChangeCallback result codes	49
Table 30 – InitLock Service parameters.....	50
Table 31 – InitLock Service result codes	50
Table 32 – ExitLock Service parameters	50
Table 33 – ExitLock Service result codes	51
Table 34 – InitDirectAccess Service parameters###.....	52
Table 35 – InitDirectAccess Service result codes	52
Table 36 – ExitDirectAccess Service parameters	52
Table 37 – ExitDirectAccess Service result codes	52
Table 38 – Transfer Service parameters.....	53
Table 39 – Transfer Service result codes	53
Table 40 – Base data types.....	54
Table 41 – Identifiers assigned to Attributes	54
Table 42 – NodeSpecifier	55
Table 43 – DataValue	55
Table 44 – InnerErrorInfo	56
Table 45 – LocalizedText Definition.....	56
Table 46 – LocaleId Examples	57
Table 47 – Range Data Type Structure	58
Table 48 – EUInformation Data Type Structure	58
Table 49 – EnumValueType Definition	58
Table 50 – GetClientTechnologyVersion Service parameters	59
Table 51 – OpenUserInterface Service parameters	60
Table 52 – LogAuditTrailMessage Service parameters.....	60
Table 53 – SaveUserSettings Service parameters.....	61
Table 54 – LoadUserSettings Service parameters.....	61
Table 55 – Trace Service parameters.....	62
Table 56 – ShowMessageBox Service parameters.....	62
Table 57 – ShowProgressBar Service parameters.....	63
Table 58 – UpdateShowProgressBar Service parameters	63
Table 59 – EndShowProgressBar Service parameters	63
Table 60 – StandardUIActionItemsChange Service parameters.....	64
Table 61 – SpecificUIActionItemsChange Service parameters.....	64
Table 62 – InitExportFile Service parameters	65
Table 63 – WriteExportFile Service parameters.....	66
Table 64 – FinishExportFile Service parameters.....	66

Table 65 – InitImportFile Service parameters	67
Table 66 – ReadImportFile Service parameters.....	67
Table 67 – FinishImportFile Service parameters.....	68
Table 68 – InitOpenDefaultApplication Service parameters	68
Table 69 – WriteOpenDefaultApplication Service parameters.....	69
Table 70 – FinishOpenDefaultApplication Service parameters.....	69
Table 71 – GetHostingProperties Service parameters	70
Table 72 – GetHostingProperties Key Value Pairs.....	70
Table 73 – DefaultResult definition.....	71
Table 74 – ButtonSet definition	71
Table 75 – AcknStyle definition	71
Table 76 – Activate Service parameters	72
Table 77 – Deactivate Service parameters	73
Table 78 – SetSystemLabel Service parameters	73
Table 79 – SetTraceLevel Service parameters	74
Table 80 – GetStandardUIActionItems Service parameters	74
Table 81 – GetSpecificUIActionItems Service parameters	74
Table 82 – InvokeStandardUIAction Service parameters	75
Table 83 – InvokeSpecificUIAction Service parameters.....	75
Table 84 – TraceLevel definition	75
Table 85 – StandardUIAction definition	76
Table 86 – StandardUIActionItem definition.....	76
Table 87 – SpecificUIActionItem definition.....	77
Table 88 – UIP states	78
Table 89 – UIP state transitions	78
Table A.1 – Elements of AbortRequestT	91
Table A.2 – Enumerations of AccessT.....	91
Table A.3 – Elements of AcknowledgementRequestT	92
Table A.4 – Elements of ActionListT.....	92
Table A.5 – Elements of ActionRequestT	93
Table A.6 – Elements of ActionResponseT	94
Table A.7 – Elements of ActionT	95
Table A.8 – Elements of AxisListT.....	95
Table A.9 – Attributes of AxisT	96
Table A.10 – Elements of AxisT	96
Table A.11 – Elements of BitEnumerationItemListT	97
Table A.12 – Elements of BitEnumerationItemT	97
Table A.13 – Elements of ButtonListT	98
Table A.14 – Elements of ChartT	98

Table A.15 – Enumerations of ChartTypeT	99
Table A.16 – Enumerations of ColorNameT	100
Table A.17 – Enumerations of DateTimeDataT	102
Table A.18 – Elements of DelayMessageRequestT	102
Table A.19 – Attributes of DiagramLineT	103
Table A.20 – Elements of DiagramLineT	103
Table A.21 – Elements of EnumerationItemListT	104
Table A.22 – Elements of EnumerationItemT	104
Table A.23 – Elements of GraphT	105
Table A.24 – Elements of GridT	106
Table A.25 – Enumerations of HandlingT	106
Table A.26 – Attributes of ImageT	107
Table A.27 – Elements of ImageT	107
Table A.28 – Elements of InfoRequestT	108
Table A.29 – Elements of InputRequestT	108
Table A.30 – Elements of InputResponseT	109
Table A.31 – Elements of InputValueT	109
Table A.32 – Elements of InputValueTypeT	110
Table A.33 – Elements of LabelHelpT	110
Table A.34 – Elements of LabelT	110
Table A.35 – Enumerations of LineTypeT	111
Table A.36 – Attributes of MenuT	112
Table A.37 – Elements of MenuT	113
Table A.38 – Attributes of MenuReferenceT	113
Table A.39 – Elements of MenuReferenceT	113
Table A.40 – Enumerations of MenuStyleT	114
Table A.41 – Enumerations of NumericDataT	115
Table A.42 – Elements of NumericTemplateT	115
Table A.43 – Elements of OptionListT	116
Table A.44 – Enumerations of OrientationT	116
Table A.45 – Elements of ParameterInputRequestT	116
Table A.46 – Elements of ParameterListT	117
Table A.47 – Elements of ParameterT	118
Table A.48 – Elements of PluginT	119
Table A.49 – Elements of RangeListT	119
Table A.50 – Elements of RangeT	120
Table A.51 – Enumerations of ScalingT	121
Table A.52 – Elements of SelectionRequestT	121
Table A.53 – Elements of SelectionResponseT	121

Table A.54 – Enumerations of SizeT	122
Table A.55 – Enumerations of ParameterClassT	124
Table A.56 – Enumerations of ActionClassT	126
Table A.57 – Elements of SourceListT	127
Table A.58 – Elements of SourceT	127
Table A.59 – Enumerations of StringDataT	128
Table A.60 – Elements of StringTemplateT	128
Table A.61 – Elements of StringOptionListT	129
Table A.62 – Elements of StringOptionT	129
Table A.63 – Elements of StringT	129
Table A.64 – Enumerations of TimeScaleT	130
Table A.65 – Elements of UidLayoutInformation	130
Table A.66 – Elements of UidRequestT	131
Table A.67 – Elements of UidResponseT	131
Table A.68 – Attributes of UiElementSizeableT	132
Table A.69 – Elements of UiElementSizeableT	132
Table A.70 – Elements of UiElementT	133
Table A.71 – Elements of UiTemplateT	133
Table A.72 – Elements of VariantT	134
Table A.73 – Elements of VariantOptionListT	135
Table A.74 – Elements of VariantOptionT	135
Table A.75 – Elements of VectorListT	135
Table A.76 – Elements of VectorT	136
Table A.77 – Elements of WaveformListT	136
Table A.78 – Elements of WaveformT	137
Table A.79 – Elements of WaveformTypeHorizontalT	138
Table A.80 – Elements of WaveformTypeVerticalT	138
Table A.81 – Elements of WaveformTypeYTT	139
Table A.82 – Elements of WaveformTypeXYT	139
Table A.83 – Elements of WaveformKeyPointListT	140
Table A.84 – Attributes of WaveformVectorT	140
Table A.85 – Elements of WaveformVectorT	141
Table A.86 – Elements of WaveformVectorElementListT	141
Table A.87 – Elements of WaveformVectorElementT	142

INTRODUCTION

The IEC 62769 series has the general title *Field Device Integration (FDI)* and the following parts:

- Part 1: Overview
- Part 2: FDI Client
- Part 3: FDI Server
- Part 4: FDI Packages
- Part 5: FDI Information Model
- Part 6: FDI Technology Mapping
- Part 7: FDI Communication Devices
- Part 100: Profiles – Generic Protocol Extensions
- Part 101-1: Profiles – Foundation Fieldbus H1
- Part 101-2: Profiles – Foundation Fieldbus HSE
- Part 103-1: Profiles – PROFIBUS
- Part 103-4: Profiles – PROFINET
- Part 109-1: Profiles – HART and WirelessHART
- Part 115-2: Profiles – Protocol-specific Definitions for Modbus RTU
- Part 150-1: Profiles – ISA 100.11a

FIELD DEVICE INTEGRATION (FDI) –

Part 2: FDI Client

1 Scope

This part of IEC 62769 specifies the FDI Client. The overall FDI architecture is illustrated in Figure 1. The architectural components that are within the scope of this document have been highlighted in this figure.

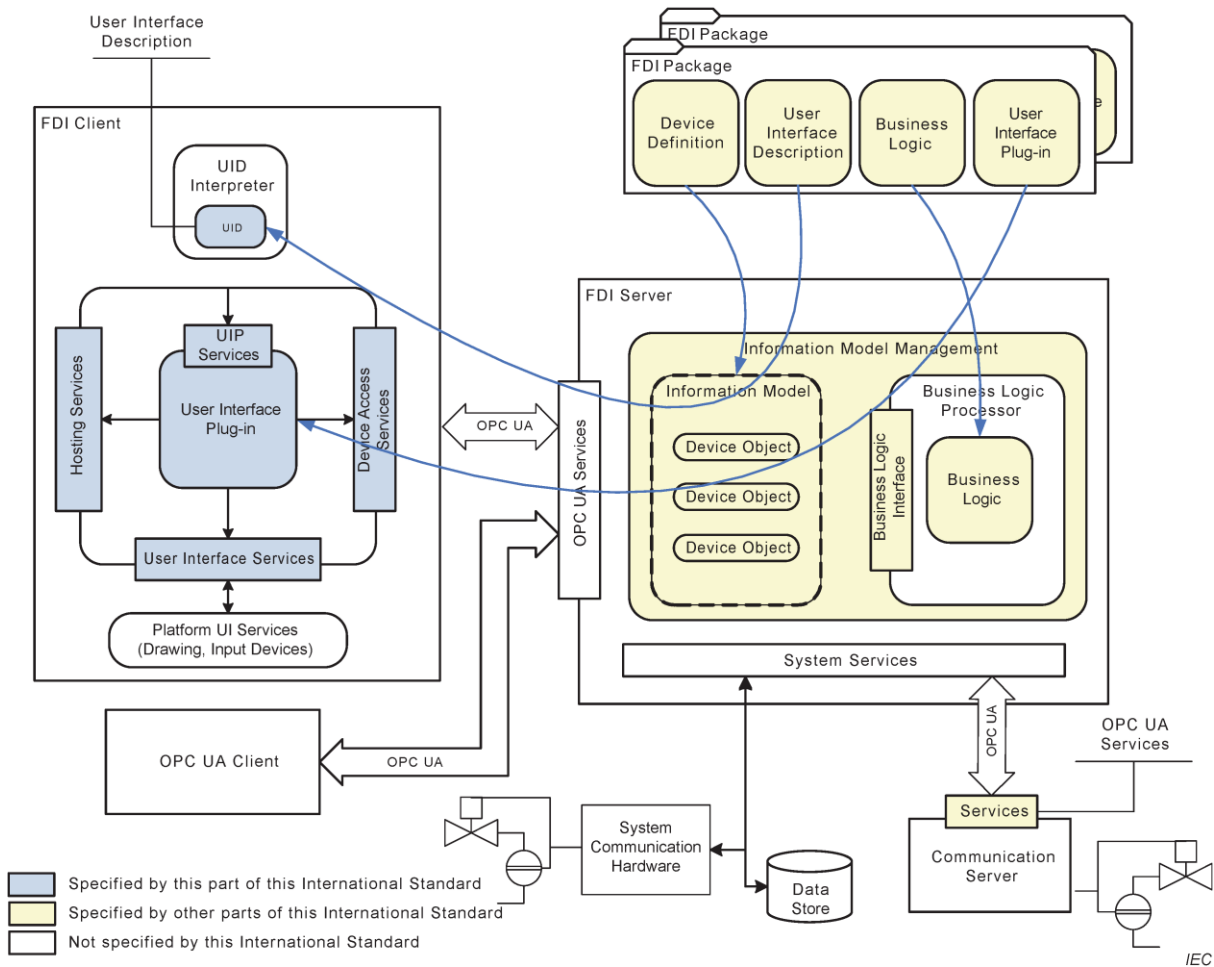


Figure 1 – FDI architecture diagram

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

IEC 62443-3-3:2013, *Industrial communication networks – Network and system security – Part 3-3: System security requirements and security levels*

IEC 62769-1, *Field Device Integration (FDI) – Part 1: Overview*IEC 62769-3, *Field Device Integration (FDI) – Part 3: FDI Server*IEC 62769-4, *Field Device Integration (FDI) – Part 4: FDI Packages*

IEC 62769-5, *Field Device Integration (FDI) – Part 5: FDI Information Model*

IEC 62769-6, *Field Device Integration (FDI) – Part 6: FDI Technology Mapping*

IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*

IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

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

ISO 639, *Codes for the representation of names of languages*

ISO 3166, *Codes for the representation of names of countries and their subdivisions*

IETF RFC 3066, *Tags for the Identification of Languages*

XMLSchema-1, *XML Schema: Structures* (available at <http://www.w3.org/TR/xmlschema-1/>)

XMLSchema-2, *XML Schema: Datatypes* (available at <http://www.w3.org/TR/xmlschema-2/>)

3 Terms, definitions, abbreviated terms and conventions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62769-1 as well as the following apply.

3.1.1 Terms used for Services

3.1.1.1

Locking Services

set of Services through which access to a Device is controlled

3.1.1.2

Device Model Services

sub-set of the Device Access Services through which a UIP can access the information of a Device

3.1.1.3

Direct Access Services

sub-set of the Device Access Services through which a UIP can directly access a Device

3.1.2 Terms used for Device Access Services

3.1.2.1

Attribute

information element of a Node

Note 1 to entry: Some Attributes exist for all NodeClasses and some are specific to a given NodeClass.

Note 2 to entry: Supersedes the definition given in IEC 62769-1.

3.1.2.2

Device Model

hierarchy of Nodes that represents an existing Device

3.1.2.3

Node

element in the Device Model that can be addressed via the Device Access Services

Note 1 to entry: Supersedes the definition given in IEC 62769-1.

3.1.2.4

NodeClass

either an Object or a Variable

Note 1 to entry: Supersedes the definition given in IEC 62769-1.

3.1.2.5

Object

instance of the Object NodeClass

Note 1 to entry: Supersedes the definition given in IEC 62769-1.

3.1.2.6

Variable

instance of the Variable NodeClass

Note 1 to entry: Supersedes the definition given in IEC 62769-1.

3.2 Abbreviated terms

For the purposes of this document, the abbreviated terms and initialisms given in IEC 62769-1 and the following apply.

UTC	Coordinated Universal Time
XML	Extended mark-up language

3.3 Conventions

Conventions for service definitions are identical to those in IEC 62541-4.

Basic data types used are defined in IEC 62541-3.

"Parameter" is always an Information Model Parameter. "parameter" is the general use of the word. If ambiguous, additional context is given.

4 Overview

An FDI Package provides the necessary information for a Device Type to allow managing the Device within the system. It is provided by a device vendor and deployed in an FDI Server. It may contain two types of user interface components that are available to the FDI Client for display to a user. An FDI Package may contain only one type or both types. The two types are referred to as User Interface Plug-ins and User Interface Descriptions.

A User Interface Plug-in (UIP) is an executable element. A UIP is provided by an FDI Package and transferred to the FDI Client by the FDI Server. A UIP provides a set of UIP Services that the FDI Client uses to initialize and interact with the UIP.

NOTE 1 IEC 62769-6 defines application programming interfaces for the services described in this document.

User Interface Descriptions (UIDs) are defined using EDDL. A UID is provided to the FDI Client by the FDI Server. The FDI Client uses the UID Interpreter to interpret and execute the UID. A UID may make use of other UID and UIP components as subcomponents in order to provide a modular approach and make the best use of both descriptive and executable user interface elements.

NOTE 2 UIPs can make use of other UIPs but not of other UIDs.

The FDI Server makes UIDs and UIPs available to the FDI Client via the Information Model. The Information Model organizes the UIDs and UIPs by Device Type.

The FDI Client provides the execution environment for UIPs. The FDI Client loads the UIP from the FDI Server.

The FDI Client's UIP execution environment consists of the following sets of services that are made available to the UIP:

- Device Access Services
- Hosting Services
- User Interface Services
- Printing Services (if available in the hosting environment)

NOTE 3 It is implementation-specific whether different UIPs get the same or different interface instances to access these services. The only requirement is that there is no side-effect if two UIPs use the same instance of an interface.

Similar to the FDI Client, each UIP shall also provide a set of services (UIP Services) by which the FDI Client activates, controls and shuts down UIPs (see 6.1).

The Device Access Services enable interaction between the UIP and the FDI Server-maintained Information Model. The FDI Client takes care of the interaction with the FDI Server freeing the UIP to focus on the application level only.

UIP access to the Information Model (IM) via the Device Access Services is restricted to the Device and its sub-devices.

The Hosting Services are provided by the FDI Client for use by the UIP. The Hosting Services include services related to the FDI Client allowing the UIP to acquire information about the environment.

The User Interface Services provide the means by which the UIP accesses the user interface services of the underlying operating system. These services provide access to the screen, keyboard, mouse, and other operating system resources. The User Interface Services are defined by the chosen implementation technology and therefore no additional definitions are included in this document (see IEC 62769-6). UIPs shall use the Hosting Services to display message boxes or progress bars and shall never use comparable services provided by the underlying operating system.

There are no printing services provided by the Client. If a UIP needs to generate a printout, it accesses the printing services of the underlying operating system. No additional definitions are included in this document.

The FDI Client uses the culture setting of the currently signed-in user for the execution environment of the UIP. It uses the culture when creating OPC UA Sessions and it sets the culture for each thread it creates. UIPs shall take care for culture setting in all threads that they create.

The FDI Client provides a UID Interpreter that is used to interpret and execute UIDs. The UID XML Schema is defined in this document (see Annex A).

Business Logic is executed in the FDI Server. Some Business Logic may be exposed to the FDI Client as Actions (see Clause 7) and can be triggered by FDI Clients.

Some typical FDI Client uses cases are described in Annex C.

5 FDI Client

5.1 Device Access Services

5.1.1 General

The Device Access Services provide access to both the online and offline information of a Device or its components as defined by the FDI Package, in particular for

- browsing the Device Model,
- reading/writing of data and subscribing to data changes,
- controlling access to the Device, and
- directly communicating with the Device.

The scope for the Device Access Services will be a Device, Block, or Communication Server to which the UIP is assigned. The FDI Client is expected to map the Device Access Services to OPC UA services provided by the FDI Server.

The main Services are the Device Model Services to view and access Parameters. The Locking Services are used to control simultaneous access to a Device. The Direct Access Services allow a UIP to communicate with the Device.

Whether and how the different services are mapped to real interfaces is defined in IEC 62769-6. IEC 62769-6 also specifies how interfaces are obtained.

5.1.2 Device Model

The Device Model defines the structure of all data that are available to a UIP. It is confined to a single device instance. The entities in the structure (Parameters, Images, and Documents) are built from FDI Package information. User interface elements, like Menus, Graphs, Waveforms, are not part of the UIP Device Model.

All Device elements are organized as a defined hierarchy. The root of the hierarchy can be either a Device or a Block subject to where (by which MENU) the UIP is referenced in the User Interface Description of the FDI Package (see IEC 62769-4). The Nodes in the hierarchy are either Objects or Variables, where the main difference is that Variables provide a Value. Figure 2 illustrates the overall structure of a Device.

Figure 3 shows the structure of a block in more detail. The elements that will really be available depend on the contents of the respective FDI Package. Regular rectangles represent Object Nodes while the ones with rounded corners represent Variable Nodes. These NodeClasses are defined in 5.1.3. The top left Node is the root Node. The single hashed lines define the parent-child relationship in the hierarchy. As an example, the children of Device1 in Figure 2 are ParameterSet, ImageSet, Documentation, Blocks and SubDevices. The children of /SubDevices/Device_1b/ImageSet are Image_1 and Image_2.

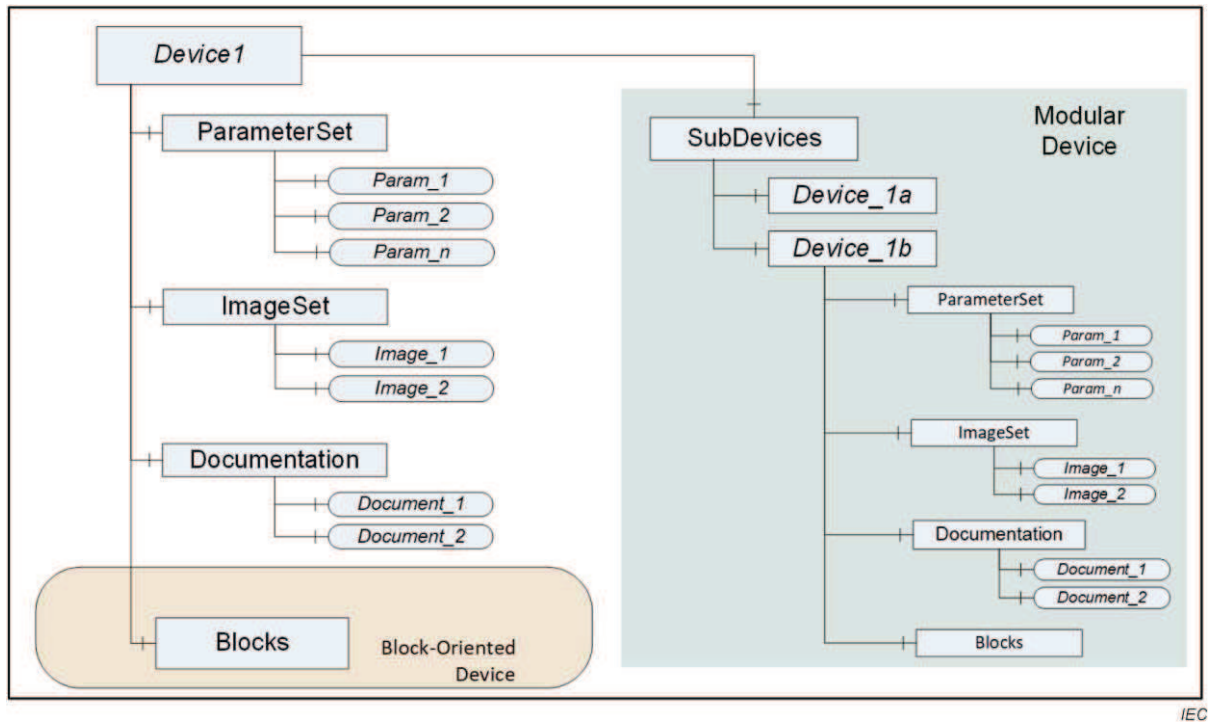


Figure 2 – Overall structure of a Device

Names that are in normal font are defined by the Device Access Services; names in italics are just placeholders for the real names as defined in the FDI Package.

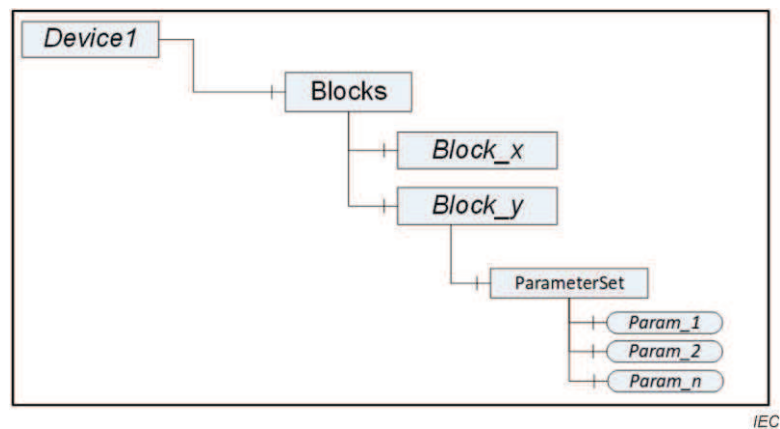


Figure 3 – Structure of Blocks

Each Node in the hierarchy is uniquely qualified with its pathname. This pathname is a concatenation of the individual Node Name Attributes. The separator is "/".

EXAMPLE The following examples illustrate qualified pathnames:

- / -- the root Node (here "Device1")
- / ParameterSet/Param_2 -- a Variable Node
- / SubDevices/Device_1b/ImageSet/Image_1 -- a picture

Certain Variables in the FDI Package may be tagged as "private", meaning they are non-browsable. Though they are non-browsable, they exist in the Device Model and can be addressed with their pathname.

5.1.3 Node model

5.1.3.1 General

The information of a Device is organized as a hierarchy of Nodes. Each Node is either an Object or a Variable. Both Object and Variable are derived from the BaseNodeClass, as illustrated in Figure 4.

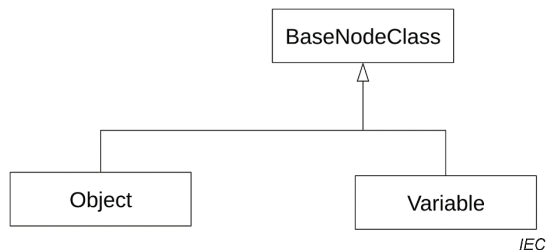


Figure 4 – Device Model NodeClasses

5.1.3.2 BaseNodeClass

This is the abstract parent NodeClass for Object and Variable Nodes. The BaseNodeClass Attributes are shown in Table 1. The Attributes of this NodeClass are available in both Object and Variable NodeClasses.

Table 1 – BaseNodeClass Attributes

Attribute	Datatype	Description
NodePath	String	Qualified path of the Node in the Device Model. This Attribute is returned by the Browse Service and cannot be read or written.
Name	String	Name of Node according to device-specific documentation.
Label	LocalizedText	Human-readable label of the Node.
Description (optional)	LocalizedText	Human-readable help string describing the Node.

Additional Attributes will be available for derived NodeClasses. For example, a Variable will have Attributes that define the DataType and the AccessRights.

5.1.3.3 Object NodeClass

Table 2 contains the list of Attributes for Objects beyond those inherited from the BaseNodeClass.

Table 2 – Object NodeClass Attributes

Attribute	Datatype	Description
LockedStatus	Boolean	This Attribute when "true" indicates that this Object is currently locked. Bad_AttributeInvalid defines that locking is not supported for this Object at all.

5.1.3.4 Variable NodeClass

5.1.3.4.1 General

Variables are used to represent Parameters, images and documents. When reading Variables that represent images or documents, the system will provide them as a ByteString. For documents, the Name Attribute will consist of the filename including the extension that can be used to identify the document type. FDI supports ".pdf" and ".txt". For the representation of images, see 5.1.3.4.2.

Table 3 contains the list of Attributes for Variables beyond those inherited from the BaseNodeClass.

Table 3 – Variable NodeClass Attributes

Attribute	Datatype	Description
Value	Variant	The value of the Variable as returned from the device (i.e. without applying the ScalingFactor). The Variant is specified in 5.1.9.4.
DataType	UInt32	The DataType Attribute specifies the data type of the Value Attribute. One of the data types specified in 5.1.9. IEC 62769-6 specifies the assignment of unique identifiers to each type.
ValueRank	Int32	Indicates whether the Value Attribute is an array. It may have the following values: >1 (MoreDimensions) – the value is an array with the specified number of dimensions. 1 (OneDimension) – the value is an array with one dimension. 0 (OneOrMoreDimensions) – the value is an array with one or more dimensions. -1 (Scalar) – the value is not an array. -2 (Any) – the value can be a scalar or an array with any number of dimensions. -3 (ScalarOrOneDimension) – the value can be a scalar or a one-dimensional array.
ArrayDimensions (optional)	UInt32[]	Specifies the length of each dimension for an array value. The Attribute is intended to describe the capability of the Variable, not the current size. The number of elements shall be equal to the value of the ValueRank Attribute. Shall be null if ValueRank <= 0. A value of 0 for an individual dimension indicates that the dimension has a Variable length. For example, if a Variable is defined by the following C array: <code>Int32 myArray[346];</code> then this Variable's DataType would point to an Int32, the Variable's ValueRank has the value 1 and the ArrayDimensions is an array with one entry having the value 346.
AccessRights	Byte	The access rights for the Value. An enumeration with one of the following values: NONE_0 The Variable value cannot be accessed READ_1 The value of the Variable may be read WRITE_2 The value of the Variable may be written READORWRITE_3 The value of the Variable may be read or written
UserAccessRights	Byte	This Attribute specifies the access rights to the Value for the currently authenticated user. They may be less than the potential access rights. The same enumeration as for AccessRights is used.
ScalingFactor (optional)	Double	This Attribute specifies a suggested scaling factor. Note that the Value Attribute contains the raw value returned from the device. It is assumed, that the (raw) value is multiplied by this factor before being displayed.
EngineeringUnits (optional)	EUInformation	EngineeringUnits specifies the units for the value (e.g. °C, hertz, seconds). See 5.1.9.3.8 for the definition of the EUInformation data type.

Attribute	Datatype	Description
Attributes for analog Variables (Variables that represent continuously variable physical quantities (e.g. pressure, temperature)).		
EURange (optional)	Range[]	<p>Defines one or more value ranges likely to be obtained in normal operation. They are intended for such use as automatically scaling a bar graph display.</p> <p>Sensor or instrument failure or deactivation can result in a returned item value that is actually outside this range. UIP software shall be prepared to deal with this.</p> <p>See 5.1.9.3.7 for the definition of the Range data type.</p> <p>Ranges may change during operation, for example by changing the operation mode of an instrument.</p> <p>Like the Value itself, Ranges are always unscaled (i.e. without applying the ScalingFactor).</p>
Attributes for discrete (enumerated) Variables (for data that may take on only a certain number of possible values (e.g. OPENING, OPEN, CLOSING, CLOSED)).		
CurrentLabel	String	Enumerated Variables expose the current numeric state in their Value Attribute. The CurrentLabel Attribute provides the name of the current enumeration value.
EnumValues	EnumValuesType[]	EnumValues is an array of {StateValue, Enumeration Name, and Help Information}. See 5.1.9.3.9 for the definition of this type. FDI Clients/UIPs may read this Attribute in advance and store it for lookup of name or help when they receive the numeric representation.
Attributes for bit-enumerated Variables (for data that represent a bit mask).		
OptionNames	String[]	<p>Bit-enumerated Variables transmit a bit mask encoded in an unsigned integer of a length that is sufficient to represent all bits.</p> <p>The OptionNames Attribute provides a human-readable representation for each valid bit of the bit mask.</p> <p>The order of the bits of the bit mask points to a position of the array of Strings in the OptionNames Attribute, i.e. the first bit points to the first entry in the array, and so on.</p> <p>The array contains an empty String for each bit that has no specific meaning.</p>

5.1.3.4.2 Representation of images

All images have the DataType and are transferred as a ByteString. FDI supports three image formats. To identify the format of the image provided in the ByteString, the initial bytes have to be parsed as outlined in Table 4.

Table 4 – Parsing of the initial bytes

Image type	Description																		
GIF	<p>Defines an image in GIF (Graphics Interchange Format). GIF is specified in http://www.w3.org/Graphics/GIF/spec-gif89a.txt . The first bytes of a GIF image are as follows:</p> <table><tr><td>Byte</td><td>1</td><td>2</td><td>3</td></tr><tr><td>Hex</td><td>47</td><td>49</td><td>46</td></tr></table>	Byte	1	2	3	Hex	47	49	46										
Byte	1	2	3																
Hex	47	49	46																
JPG	<p>Defines an image in JPG (Joint Photographic Experts Group File Interchange Format). JPG is defined in ISO/IEC 10918-1. The first bytes of a JPG image are as follows:</p> <table><tr><td>Byte</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Hex</td><td>FF</td><td>D8</td><td>FF</td><td>E0</td></tr></table>	Byte	1	2	3	4	Hex	FF	D8	FF	E0								
Byte	1	2	3	4															
Hex	FF	D8	FF	E0															
PNG	<p>Defines an image in PNG (Portable Network Graphics format). PNG is defined in IETF RFC 2083 and ISO/IEC 15948. The first bytes of a PNG image are as follows:</p> <table><tr><td>Byte</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>Hex</td><td>89</td><td>50</td><td>4E</td><td>47</td><td>0D</td><td>0A</td><td>1A</td><td>0A</td></tr></table>	Byte	1	2	3	4	5	6	7	8	Hex	89	50	4E	47	0D	0A	1A	0A
Byte	1	2	3	4	5	6	7	8											
Hex	89	50	4E	47	0D	0A	1A	0A											

5.1.3.4.3 Representation of records

A Variable hierarchy is used to represent EDDL RECORD Parameters. The root Variable represents the record itself. It will have component Variables that represent the EDDL RECORD MEMBERS (the MEMBERS of an EDDL RECORD are defined in EDDL by means of a reference to an EDDL VARIABLE.).

An example of how a record is represented in the Device Model is shown in Figure 5.

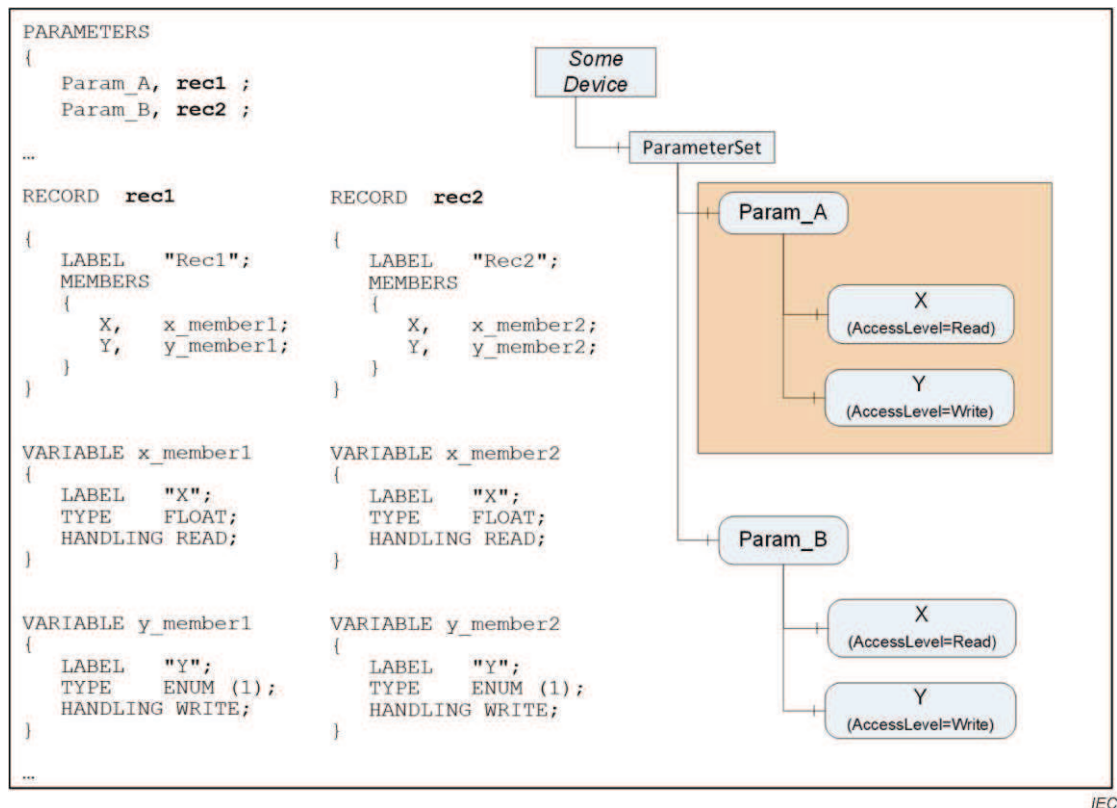


Figure 5 – Example: Variable hierarchy representing a RECORD

The Name and Label Attributes of the root Variable are set to the name of the RECORD and the LABEL Attribute, respectively. The DataType Attribute of the "root" Variable is Variant. The ValueRank Attribute is used to specify that the Value contains an array. The Value Attribute represents the values of all members in the order as defined for the RECORD. According to the example in Figure 5, the first variant will be a FLOAT and the second will be an ENUM.

For each component Variable that represents an EDDL RECORD MEMBER:

- the Name Attribute is set to the identifier of the corresponding EDDL VARIABLE,
- the Label is the LABEL Attribute of the corresponding EDDL VARIABLE,
- the Description is the HELP Attribute of the corresponding EDDL VARIABLE, and
- the AccessRights Attribute is derived from the EDDL HANDLING Attribute.

Each member of the record can be accessed with its pathname as specified above. Browsing can also be used.

EXAMPLE Example of a pathname: /ParameterSet/Param_A/X.

5.1.3.4.4 Representation of arrays, and lists of members with simple data types

A single Variable will represent an EDDL VALUE_ARRAY or LIST item when the data type of the referenced array element has a simple data type.

The DataType Attribute is set to one of the base data types (see 5.1.9.2).

The ValueRank Attribute is used to specify that the Value contains an array. In case of an EDDL VALUE_ARRAY the number of elements is exposed via the ArrayDimensions Attribute. In the case of an EDDL LIST, the number of elements is unspecified since the size can change dynamically.

5.1.3.4.5 Representation of arrays, and lists of RECORDs

Value arrays or lists of non-simple EDDL Parameters will be represented as an array of Variable hierarchies. Figure 6 shows the EDDL sample code of a VALUE_ARRAY of RECORDs and the corresponding Variable hierarchy.

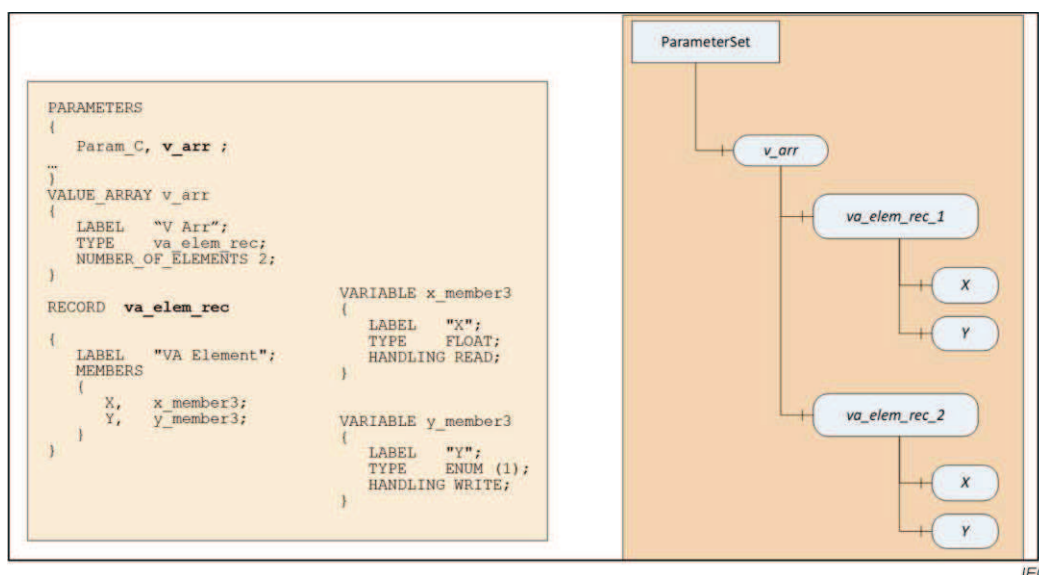


Figure 6 – Variable hierarchy representing a VALUE_ARRAY of RECORDs

The Name and Label Attributes of the root Variable are set to the name of the ARRAY and the LABEL Attribute, respectively. The DataType Attribute of the "root" DataVariable is Variant. The ValueRank Attribute is used to specify that the Value contains an array. The Value Attribute represents all VALUE_ARRAY entries. The first Variant corresponds to the first array entry and so on. Each Variant in turn contains an array. This may either be an array of simple types or an array of Variants. A RECORD is always represented as an array of Variant.

The VALUE_ARRAY element, which is in fact a RECORD, is represented as a component Variable hierarchy. The Name and Label Attributes of each root Variable representing a RECORD are set to the name of the RECORD and the LABEL Attribute, respectively. The array index (_1, _2) is appended to allow unique identification. Note that the index always begins with '1'.

The RECORD MEMBERS are also represented as component Variables as specified in 5.1.3.4.3.

Members of each record can be accessed with a pathname. Browsing can also be used.

EXAMPLE Example of a pathname: /ParameterSet/v_arr/va_elem_rec_2/Y.

5.1.4 Services

5.1.4.1 General

All Services specified in this part of IEC 62769 rely on the conventions defined in 5.1.4.2. They are specified in an abstract manner with request and response parameters in a single table.

Programmatic access to the services may be synchronous or asynchronous (see IEC 62769-6). However, asynchronicity exists to maintain responsiveness of the User Interface. Service execution shall always be assumed to be sequential.

5.1.4.2 Conventions for service definitions

The service specifications use tables to describe service parameters, as shown in Table 5. Parameters are organised in this table into request parameters and response parameters.

Table 5 – Service Definition Table

Name	Type	Description
Request		Defines the request parameters of the service
simple Parameter Name		Description of this parameter
constructed Parameter Name	structure	Description of the constructed parameter
component Parameter Name		Description of the component parameter
Response		Defines the response parameters of the service

The Name, Type and Description columns contain the name, data type and description of each parameter. All parameters are mandatory, although some may be unused under certain circumstances. The description column specifies the value to be supplied when a parameter is unused. Parameter names always begin with a lower-case character. This allows differentiating if name and type are the same, for example, name = "nodeld", type = "Nodeld".

Two types of parameters are defined in these tables, simple and constructed. Simple parameters have a simple data type, such as Boolean or String.

Constructed parameters are composed of two or more component parameters, which can be simple or constructed. Component parameter names are indented below the constructed parameter name.

The data types used in these tables may be base types or service-specific types. Base data types are listed in 5.1.9. Data types that are service-specific are defined in the parameter table of the service.

5.1.4.3 Auditing

Auditing is a requirement in many systems. It provides a means for tracking activities that occur as part of normal operation of the system. It also provides a means for tracking abnormal behavior. It is also a requirement from a security standpoint.

When an audit trail is maintained by the system, audit trail records for the Write Service invoked by the UIP will be implicitly created by the system.

In addition, UIPs have means for providing additional audit context information for things that the system cannot know:

- They can call the LogAuditTrailMessage service (see 5.2.2.5). This shall be executed in particular when the DirectAccess Services are used. The UIP shall include sufficient information in the message for precise description of the activity performed.
- They can provide a context text when using the Locking Services (see 5.1.7) or the Direct Access Services (see 5.1.8).

5.1.4.4 Services and operations

Several of the Device Model Services (e.g. Read and Write) allow specifying an array of elements that shall be processed. The processing of each individual element is called an "operation". This is an important differentiation for error handling as a service execution is considered successful even if individual operations fail. The following list explains the differences between the service and operation result codes.

- Service result code
If a service succeeds, the result code is "Good" and the response parameters are valid. If it fails, the service result code is any of the "Bad_..." service failure codes defined in 5.1.4.6. In such a case, an InnerErrorInfo (see 5.1.9.3.4) may be provided as well. Programmatic access to service failure codes is specific to the technology and may be based on exceptions (see IEC 62769-6).
- Operation result code
The result code for each operation is returned as part of the service-specific response parameters (see 5.1.4.7 for the list of available operation result codes).

Operations can return an InnerErrorInfo with each result code other than "Good" if returnInnerErrorInfo="true" in the service request.

5.1.4.5 StatusCode

The StatusCode used for service results or operational results reports the outcome of an operation. It is a 32-bit unsigned integer. The top 16 bits represent the numeric value of the code that shall be used for detecting specific errors or conditions. The bottom 16 bits are bit flags that contain additional information but do not affect the meaning of the StatusCode.

UIPs shall always check the StatusCode associated with a result before using it. Results that have an uncertain/warning status associated with them shall be used with care since these results might not be valid in all situations. Results with a bad/failed status shall never be used.

The exact bit assignments are shown in Table 6. IEC 62769-6 provides functions to help in evaluation of the StatusCode.

Table 6 – StatusCode Bit Assignments

Field	Bit Range	Description															
Severity	30 .. 31	Indicates whether the StatusCode represents a good, bad or uncertain condition. These bits have the following meanings:															
		<table><tr><th>Severity</th><th>Bits</th><th>Description</th></tr><tr><td>Good Success</td><td>00</td><td>The operation was successful; results may be used.</td></tr><tr><td>Uncertain Warning</td><td>01</td><td>The operation was partially successful; results might not be suitable for some purposes.</td></tr><tr><td>Bad Failure</td><td>10</td><td>The operation failed and any associated results cannot be used.</td></tr><tr><td>Reserved</td><td>11</td><td>Reserved for future use. Should also be treated as "Bad".</td></tr></table>	Severity	Bits	Description	Good Success	00	The operation was successful; results may be used.	Uncertain Warning	01	The operation was partially successful; results might not be suitable for some purposes.	Bad Failure	10	The operation failed and any associated results cannot be used.	Reserved	11	Reserved for future use. Should also be treated as "Bad".
		Severity	Bits	Description													
		Good Success	00	The operation was successful; results may be used.													
		Uncertain Warning	01	The operation was partially successful; results might not be suitable for some purposes.													
		Bad Failure	10	The operation failed and any associated results cannot be used.													
Reserved	11	Reserved for future use. Should also be treated as "Bad".															
Reserved	29 .. 28	Reserved for future use. Shall always be zero.															
SubCode	16 .. 27	The code is a numeric value assigned to represent different conditions. Each code has a symbolic name and a numeric value. All descriptions in this specification refer to the symbolic name. IEC 62769-6 maps the symbolic names onto a numeric value.															
Reserved	12 .. 15	Reserved for future use. Shall always be zero.															
InfoType	10 .. 11	The type of information contained in the info bits. These bits have the following meanings:															
		<table><tr><th>InfoType</th><th>Bits</th><th>Description</th></tr><tr><td>NotUsed</td><td>00</td><td>The info bits are not used and shall be set to zero.</td></tr><tr><td>DataValue</td><td>01</td><td>The StatusCode and its info bits are associated with a data value returned from the FDI Server.</td></tr><tr><td>Reserved</td><td>1X</td><td>Reserved for future use. The info bits shall be ignored.</td></tr></table>	InfoType	Bits	Description	NotUsed	00	The info bits are not used and shall be set to zero.	DataValue	01	The StatusCode and its info bits are associated with a data value returned from the FDI Server.	Reserved	1X	Reserved for future use. The info bits shall be ignored.			
		InfoType	Bits	Description													
		NotUsed	00	The info bits are not used and shall be set to zero.													
DataValue	01	The StatusCode and its info bits are associated with a data value returned from the FDI Server.															
Reserved	1X	Reserved for future use. The info bits shall be ignored.															
InfoBits	0 .. 9	Additional information bits that depend on the Info Type field.															

Table 7 describes the structure of the InfoBits when the Info Type is set to DataValue (01).

Table 7 – DataValue InfoBits

Info Type	Bit Range	Description		
LimitBits	8 .. 9	The limit bits associated with the data value. The limits bits have the following meanings:		
		Limit	Bits	Description
		None	00	The value is free to change.
		Low	01	The value is at the lower limit for the data source.
		High	10	The value is at the higher limit for the data source.
		Constant	11	The value is constant and cannot change.
Overflow	7	If this bit is set, not every detected change has been returned since the FDI Server's queue buffer for the subscribed Variable reached its limit and had to purge out data.		
Reserved	0 .. 6	Reserved for future use. Shall always be zero.		

5.1.4.6 Service failure codes

Table 8 defines result codes for the services. The column Service in Table 8 lists which code may be returned by which service. Result codes that are specific to an individual service are also specified with the service.

Table 8 – Service result codes

Exception name code	Description	Service
Bad_AlreadyLocked	The Node is already locked by another FDI Client.	InitLock
Bad_LockRequired	The passed Node is not yet locked.	Write, DirectAccess
Bad_MaxAgeInvalid	The max age parameter is invalid.	Read
Bad_NodeInvalid	The identifier does not refer to a valid Node in the Device Model. This result code is used both as service- and as operation-level result code.	Browse, InitLock, ExitLock
Bad_NothingToDo	There was nothing to do because the caller passed an empty list of operations.	Read, Write, Subscribe, Unsubscribe
Bad_NotSupported	The Service is not supported for the specified Node or for the Device/Block in general.	Locking, DirectAccess
Bad_RequestCancelled	The request was cancelled by Client / UIP.	All
Bad_SubscriptionIdInvalid	The subscription id is not valid.	Subscribe, Unsubscribe, DeleteSubscription
Bad_Timeout	The operation timed out.	All
Bad_TooManySubscriptions	The FDI Server has reached its maximum number of subscriptions.	CreateSubscription
Bad_InvalidState	The specified Node is in a state that does not permit this operation.	Services for Locking
Bad_TooManyOperations	The request could not be processed because it specified too many operations.	Read, Write, Subscribe
Bad_UnexpectedError	An unexpected error occurred.	All

5.1.4.7 Operational status codes

Table 9 defines the status codes for all operation level results (for services that have individual operations such as Read, Write or Subscribe). Each service defines the applicable subset and instead of a description will contain references to this table.

NOTE The value in Table 9 is referring to the Parameter value, not to the communication quality.

Table 9 – Operation level result codes

Operation Level Result Code	Description
Good	The operation completed successfully.
Good_LocalOverride	The value has been overridden.
Good_PostActionFailed	The value of a Variable was successfully read or written but one of the post actions failed.
Good_Edited	The Value has been modified in the EditContext and is not yet in the Device.
Good_DependentValueChanged	The Value of a dependent Variable was changed but not yet applied.
Uncertain	The operation completed however its outputs may not be usable.
Uncertain_NoCommunicationLastValue	Communication to the data source has failed. The Variable value is the last value that had a good quality.
Uncertain_LastUsableValue	Whatever was updating this value will no longer be doing so.

Operation Level Result Code	Description
Uncertain_SubstituteValue	The value is an operational value that was manually overwritten.
Uncertain_InitialValue	The value is an initial value for a Variable that normally receives its value from another Variable.
Uncertain_SensorNotAccurate	The value is at one of the sensor limits.
Uncertain_EngineeringUnitsExceeded	The value is outside of the range of values defined for this parameter.
Uncertain_SubNormal	The value is derived from multiple sources and has less than the required number of good sources.
Uncertain_DominantValueChanged	A change of a dominant value – e.g. an engineering unit – made a dependent value invalid.
Bad	The operation failed.
Bad_AttributeInvalid	The attributeId is not supported for the specified Node.
Bad_ConfigurationError	There is a problem with the configuration that affects the usefulness of the value.
Bad_DeviceFailure	There has been a failure in the device/data source that generates the value.
Bad_IndexRangeInvalid	The indexRange parameter has an invalid syntax.
Bad_NodeInvalid	The identifier does not refer to a valid Node in the Device Model. This result code is used both as service- and as operation-level result code.
Bad_NotConnected	The Variable should receive its value from another Variable, but has never been configured to do so.
Bad_NotReadable	The access level does not allow reading or subscribing to the Node.
Bad_NotWritable	The access level does not allow writing to the Node.
Bad_OutOfRange	The value was out of range.
Bad_OutOfService	The source of the data is not operational.
Bad_SensorFailure	There has been a failure in the sensor from which the value is derived by the device/data source.
Bad_TypeMismatch	The value supplied is not of the same type as the Variable's value.
Bad_UIPHandleInvalid	The handle does not refer to a subscribed Node Attribute.
Bad_UserAccessDenied	User does not have permission to perform the requested operation.
Bad_WaitingForInitialData	Waiting for the FDI Server to obtain values from the underlying data source. After subscribing to Variables, it may take some time before values are delivered. In such cases an initial update may be sent with this status prior to the Notification with the first valid value.

5.1.5 Base Property Services

5.1.5.1 Overview

The Base Property Services provide access to basic Device Access properties.

5.1.5.2 Get DeviceAccess interface version

5.1.5.2.1 Description

This service returns the version of the interface that the UIP is using.

5.1.5.2.2 Parameters

Table 10 defines the parameters for the service.

Table 10 – GetDeviceAccessInterfaceVersion Service parameters

Name	Type	Description
Request		
Response		
version	String	FDI Technology Version of the DeviceAccess interface. The format of the value is xx.yy.zz as defined in IEC 62769-4.

5.1.5.2.3 Service results

There are no service results other than the common codes specified in 5.1.4.6.

5.1.5.3 GetOnlineAccessAvailability

5.1.5.3.1 Description

This service returns a hint as to whether online access is available in principle. This is general information. It does not clarify whether online access to a specific device is possible.

5.1.5.3.2 Parameters

Table 11 defines the parameters for the service.

Table 11 – GetOnlineAccessAvailability Service parameters

Name	Type	Description
Request		
Response		
onlineAccess	Boolean	This value when "false" specifies that online access is not available. "true" indicates basic availability.

5.1.5.3.3 Service results

There are no service results other than the common codes specified in 5.1.4.6.

5.1.6 Device Model Services

5.1.6.1 Overview

The Device Model Services include:

- Browse
- Read
- Write

- Subscription
 - CreateSubscription
 - Subscribe
 - Unsubscribe
 - DeleteSubscription

The services provide access to the offline representation and the online representation of the Device. The online Nodes for a Device are always present. However, access to data that require communication may be rejected with proper status codes such as Bad_NotConnected.

NOTE The online model does not contain SubDevices. If the UIP uses a path that includes SubDevices to access an online Node, this path will be evaluated in the offline hierarchy.

Cancel services are available for Browse, Read, and Write. The exact definition and mechanics depend on the technology used for the implementation of these services.

5.1.6.2 Browse

5.1.6.2.1 Description

Browses a single level in the hierarchy in the Device Model and returns Attributes for all children of the specified Node.

5.1.6.2.2 Parameters

Table 12 defines the parameters for the service.

Table 12 – Browse Service parameters

Name	Type	Description
Request		
nodeToBrowse	NodeSpecifier	The identifier of the Node to be browsed. See 5.1.9.3.2 for the definition of the NodeSpecifier type.
Response		
browseResult[]	structure	List of Nodes that are on the next level down in the hierarchy. For each Node the following Attributes will be returned.
nodePath	String	See BaseNodeClass in 5.1.3.2.
name	String	See BaseNodeClass in 5.1.3.2.
label	LocalizedText	See BaseNodeClass in 5.1.3.2.

5.1.6.2.3 Service results

No specific Service result codes are defined for Browse. Common StatusCodes are defined in Table 8.

5.1.6.3 CancelBrowse

5.1.6.3.1 Description

Calling CancelBrowse indicates that the UIP has no further interest in the results of this service. Execution will be stopped when possible.

Cancel is a suggestion to the system. Owing to asynchronous execution, the service may already be fully or partially completed.

5.1.6.3.2 Parameters

Table 13 defines the parameters for the service.

Table 13 – CancelBrowse Service parameters

Name	Type	Description
Request		
serviceld	<technology dependent>	The identifier of the service to cancel.
Response		

5.1.6.3.3 Service results

No specific Service result codes are defined for CancelBrowse. Successfully cancelled Browse requests shall respond with Bad_RequestCancelled. Common StatusCodes are defined in Table 8.

5.1.6.4 Read

5.1.6.4.1 Description

This service is used to read Attributes of Object or Variable Nodes. UIs that need to monitor Variable Attributes for changes shall use the Subscription services instead of Read.

5.1.6.4.2 Parameters

Table 14 defines the parameters for the service.

Table 14 – Read Service parameters

Name	Type	Description
Request		
returnInnerErrorInfo	Boolean	A value of "true" requests error information from calls to an underlying system to be returned when available. "false" defines that this information shall not be returned.
attributesToRead[]	Structure	The Attributes to read.
node	NodeSpecifier	The identifier of the Node that contains the Attribute to read. See 5.1.9.3.2 for the definition of the NodeSpecifier type.
attributeId	AttributeId	Numeric identifier of the Attribute to Read. See 5.1.9.3.3.
indexRange	NumericRange	This parameter is used to identify a single element of an array, or a single range of indexes for arrays. If a range of elements is specified, the values are returned as a composite. The first element is identified by index 0 (zero). This parameter is ignored if the specified Attribute is not an array or a structure. However, if the specified Attribute is an array or a structure, and this parameter is null (Null or empty String), then all elements are to be included in the range. See 5.1.9.3.6 for a detailed definition.
maxAge	UInt32	Maximum age of the value to be read in milliseconds. If the FDI Server has a cached value no older than maxAge, it will return the cached value rather than requesting a new value from the device. If maxAge is set to 0, the FDI Server shall read a new value from the data source. Values greater than $2^{31} - 1$ (0x7fff ffff) are invalid for maxAge.
Response		
readResult []	DataValue	The StatusCode, Value and timestamps for each Node Attribute that was read. The order of this list matches the order of the attributesToRead request parameter. The DataValue is defined in 5.1.9.3.3.
innerErrorInfos []	InnerErrorInfo	List of error information from calls to an underlying system. See 5.1.9.3.4. Matches the size and order of the attributesToRead request parameter. This list is empty if inner error information was not requested or if no information was encountered in the processing of the request.

5.1.6.4.3 Service results

Table 15 defines values for the Service result code. Other common StatusCodes are defined in Table 8.

Table 15 – Read Service result codes

Result code	Description
Bad_MaxAgeInvalid	The max age parameter is invalid.

5.1.6.4.4 Operation result codes

Table 16 defines values for the operation status code contained in the DataValue of each readResult element. All operational status codes with their description are in Table 9.

Table 16 – Read operation result codes

Result code
Good
Good_LocalOverride
Good_PostActionFailed
Good_DependentValueChanged
Uncertain
Uncertain_NoCommunicationLastValue
Uncertain_LastUsableValue
Uncertain_SubstituteValue
Uncertain_InitialValue
Uncertain_SensorNotAccurate
Uncertain_EngineeringUnitsExceeded
Uncertain_SubNormal
Uncertain_DominantValueChanged
Bad
Bad_UserAccessDenied
Bad_ConfigurationError
Bad_NotConnected
Bad_DeviceFailure
Bad_SensorFailure
Bad_OutOfRange
Bad_OutOfService
Bad_NodeInvalid
Bad_AttributeInvalid
Bad_IndexRangeInvalid
Bad_NotReadable

5.1.6.5 CancelRead

5.1.6.5.1 Description

Calling CancelRead indicates that the UIP has no further interest in the results of this service. Execution will be stopped when possible.

Cancel is a suggestion to the system. Owing to asynchronous execution, the service may already be fully or partially completed.

5.1.6.5.2 Parameters

Table 17 defines the parameters for the service.

Table 17 – CancelRead Service parameters

Name	Type	Description
Request		
serviceld	<technology dependent>	The identifier of the service to cancel.
Response		

5.1.6.5.3 Service results

No specific Service result codes are defined for CancelRead. Successfully cancelled Read requests shall respond with Bad_RequestCancelled. Common StatusCodes are defined in Table 8.

5.1.6.6 Write

5.1.6.6.1 Description

This service is used to write values to one or more Variables. For array values, this service allows writing the entire array, writing individual elements or writing ranges of elements.

Explicit locking is required (see 5.1.7). If the Node is not locked, the request will be rejected with Bad_LockRequired.

The service response is not returned until the write operation has been executed by the system. Rollback is the responsibility of the FDI Client/UIP.

The values shall have the correct data type.

NOTE No automatic data type translation takes place (see Bad_TypeMismatch operation result code).

5.1.6.6.2 Parameters

Table 18 defines the parameters for the service.

Table 18 – Write Service parameters

Name	Type	Description
Request		
returnInnerErrorInfo	Boolean	A value of "true" requests error information from calls to an underlying system to be returned when available. "false" defines that this information shall not be returned.
variablesToWrite[]	structure	List of Variables to write to. No assumptions should be made about the order of processing this list. If the order matters, separate service requests shall be used.
node	NodeSpecifier	The identifier of the Node that contains the Attribute to write. See 5.1.9.3.2 for the definition of the NodeSpecifier type.
indexRange	NumericRange	See 5.1.9.3.6 for a detailed definition.
value	Variant	Value to write.
Response		
writeResult[]	UInt32	Status codes for operation results as defined in Table 19. The order of this list matches the order of the variablesToWrite request parameter.
innerErrorInfos []	InnerErrorInfo	List of error information from calls to an underlying system. See 5.1.9.3.4. Matches the size and order of the variablesToWrite request parameter. This list is empty if inner error information was not requested or if no information was encountered in processing of the request.

5.1.6.6.3 Service results

There are no service results other than the common codes specified in 5.1.4.6.

5.1.6.6.4 Operation result codes

Table 19 defines values for the operation status code contained in the writeResult elements. All operational status codes with their description are in Table 9.

Table 19 – Write operation result codes

Result code
Good
Good_PostActionFailed
Bad
Bad_UserAccessDenied
Bad_NodeInvalid
Bad_IndexRangeInvalid
Bad_TypeMismatch
Bad_OutOfRange
Bad_NotWritable

5.1.6.7 CancelWrite

5.1.6.7.1 Description

Calling CancelWrite indicates that the UIP has no further interest in the results of this service. Execution will be stopped when possible.

Cancel is a suggestion to the system. Owing to asynchronous execution, the service may already be fully or partially completed.

5.1.6.7.2 Parameters

Table 20 defines the parameters for the service.

Table 20 – CancelWrite Service parameters

Name	Type	Description
Request		
serviceld	<technology dependent>	The identifier of the service to cancel.
Response		

5.1.6.7.3 Service results

No specific Service result codes are defined for CancelWrite. Successfully cancelled Write requests shall respond with Bad_RequestCancelled. Common StatusCodes are defined in Table 8.

5.1.6.8 Subscriptions

5.1.6.8.1 Subscription mechanism

Subscriptions allow the UIP to receive unsolicited callbacks from the FDI Client when the subscribed Node Attributes change. Subscriptions are a more efficient way to get periodic updates of data than by issuing repeated calls to the Read service, i.e. polling. The UIP creates a subscription by calling the CreateSubscription service (see 5.1.6.8.2). When the CreateSubscription service is called the UIP shall supply a callback parameter with the UIP-specific DataChangeCallback service (see 5.1.6.8.6). After receiving the subscription identifier in the CreateSubscription response, the UIP shall add Node Attributes of interest to the Subscription by calling the Subscribe service.

Once the initial values have been reported, the DataChangeCallback service is only called when Node Attributes change and only the Node Attributes that have changed are reported to the callback. The UIP controls the maximum frequency at which the callback should be invoked by specifying a rate in milliseconds.

After Node Attributes have been subscribed, their values may not be immediately available and may become available at different times. As such, the initial callback may not include all of the subscribed Node Attributes. Furthermore, it is possible that the initial update for a subscribed Node Attribute will return a Bad or Uncertain StatusCode.

Additionally, the time span between when a change happens and when the callback is invoked depends on where the information to be changed is maintained. Some values are maintained by the system, which allows immediate notification; others are located in the physical device so that communication is required to access it.

5.1.6.8.2 CreateSubscription Service

5.1.6.8.2.1 Description

This service is used to create a subscription.

5.1.6.8.2.2 Parameters

Table 21 defines the parameters for the service.

Table 21 – CreateSubscription Service parameters

Name	Type	Description
Request		
requestedUpdateRate	UInt32	The fastest rate, in milliseconds, at which the UIP requests to be called back with data changes, specified by the minimum milliseconds to elapse between updates. Regardless of the requested rate, a callback only occurs if data has changed. A rate of "0" indicates that the caller wants to be notified of changes as soon as possible. The service will return the fastest possible rate as revisedUpdateRate.
dataChangeCallback	DataChangeCallback	Callback for sending data change updates to the UIP. See 5.1.6.8.6. The DataChangeCallback is a service implemented and provided by the UIP.
Response		
revisedUpdateRate	UInt32	The actual rate that the FDI Server will use, expressed as the minimum milliseconds to elapse between updates (if data has changed since the previous update).
subscriptionId	SubscriptionId	The callee-assigned identifier for the subscription. The SubscriptionId type is technology dependent.

5.1.6.8.2.3 Service results

Table 22 defines values for the service result. Common results are defined in Table 8.

Table 22 – CreateSubscription Service result codes

Result code	Description
Bad_TooManySubscriptions	The FDI Server has reached its maximum number of subscriptions.

5.1.6.8.3 Subscribe Service

5.1.6.8.3.1 Description

This service is used to add one or more Node Attributes to an existing subscription.

Subscribing is permitted for all Node Attributes.

5.1.6.8.3.2 Parameters

Table 23 defines the parameters for the service.

Table 23 – Subscribe Service parameters

Name	Type	Description
Request		
subscriptionId	SubscriptionId	The identifier for an existing subscription that was returned by the CreateSubscription service.
returnInnerErrorInfo	Boolean	A value of "true" requests error information from calls to an underlying system to be passed in DataChangeCallbacks, when available. "false" defines that this information shall not be returned.
monitoredItemsToAdd[]	structure	The Attributes to add to the subscription.
node	NodeSpecifier	The identifier of the Node that contains the Attribute to subscribe. See 5.1.9.3.2 for the definition of the NodeSpecifier type.
attributeId	AttributeId	Numeric identifier of the Attribute to subscribe. See 5.1.9.3.3.
indexRange	NumericRange	This parameter is used to identify a single element of an array, or a single range of indexes for arrays. If a range of elements is specified, the values are returned as a composite. The first element is identified by index 0 (zero). This parameter is ignored if the specified Attribute is not an array or a structure. However, if the specified Attribute is an array or a structure, and this parameter is null, then all elements are to be included in the range. See 5.1.9.3.6 for a detailed definition.
samplingInterval	Int32	The interval that defines the fastest rate at which the Attribute should be accessed and evaluated. This interval is defined in milliseconds. The value 0 indicates that the FDI Server should use the fastest practical rate. The value -1 indicates that the default sampling interval defined by the UpdateRate of the Subscription is used. See 5.1.6.8.3.3 for further details on the sampling interval.
uiPHandle	UInt32	A handle (an identifier) provided by the UIP for the subscribed Node Attribute. This handle will be passed together with the data in the DataChangeCallback service so that the UIP can easily associate each changed value with the subscribed Node Attribute. The uiPHandle is likely an index into a table somewhere. It does not have to be unique (there could be multiple subscribed items pointing to the same table entry).
Response		
subscribeResult []	structure	List of results for the subscribed Attributes. The size and order of the list matches the size and order of the attributesToSubscribe request parameter.
statusCode	UInt32	Status code for the respective Attribute to subscribe as defined in Table 24.
monitoredItemId	UInt32	FDI Server-assigned id for the subscribed Attribute. This id is unique within the Subscription and shall be used when calling Unsubscribe. This parameter is present only if the statusCode indicates that the Attribute was successfully subscribed.
revisedSamplingInterval	Int32	The actual sampling interval that will be used. This value is based on a number of factors, including the capabilities of the underlying system.

Subscribing will succeed even if the current user is not authorized to access the Node Attribute. If this is the case, the initial callback will return the operation result "Bad_UserAccessDenied". Once this denial is cleared away (for instance, after a more powerful user identity is provided) the UIP will receive data changes for this Node Attribute.

It is possible to subscribe to any Attribute – not just the Value. While it may not make sense for all Attributes, monitoring of some Attributes provides additional possibilities. Some examples include:

- monitoring the LockedStatus to determine when the lock by some other client is removed;
- monitoring the CurrentLabel of Enumerations to receive a displayable name rather than a numeric value.

5.1.6.8.3.3 Sampling interval

Each subscribed item is assigned a sampling interval that is either inherited from the updateRate of the Subscription or that is defined specifically to override that rate. The sampling interval indicates the fastest rate at which the value should be sampled in the device for data changes.

The assigned sampling interval defines a "best effort" cyclic rate that is used to sample the item from its source. "Best effort" in this context means that the system does its best to sample at this rate. Sampling at rates faster than this rate is acceptable, but not necessary to meet the needs of the UIP. How the system deals with the sampling rate and how often it actually polls its data source internally is a system implementation detail. However, the time between values returned to the UIP shall be greater than or equal to the sampling interval.

The FDI Client may also specify 0 for the sampling interval, which indicates that the system should use the fastest practical rate. It is expected that systems will support only a limited set of sampling intervals to optimize their operation. If the exact interval requested by the UIP is not supported, then the most appropriate interval as determined by the system will be assigned and returned to the UIP.

Data may be collected based on a sampling model or generated based on an exception-based model. The fastest supported sampling interval may be equal to 0, which indicates that the data item is exception-based rather than being sampled at some period. When it is exception-based, the underlying system does not require sampling.

In many cases, the system has no knowledge of the data update logic. In this case, even though the system samples at the negotiated rate, the data might be updated by the underlying system at a much slower rate. In this case, changes can only be detected at this slower rate.

UIPs should also be aware that the sampling by the system and the update cycle of the device are usually not synchronized, which can lead to additional delays.

5.1.6.8.3.4 Service results

There are no service results other than the common codes specified in 5.1.4.6.

5.1.6.8.3.5 Operation result codes

Table 24 defines values for the operation statusCode contained in the results. All operational status codes with their description are in Table 9.

Table 24 – Subscribe operation result codes

Result code
Good
Bad
Bad_NodeInvalid
Bad_AttributeInvalid
Bad_NotReadable
Bad_UserAccessDenied

5.1.6.8.4 Unsubscribe Service

5.1.6.8.4.1 Description

This service is used to unsubscribe to one or more Node Attributes.

5.1.6.8.4.2 Parameters

Table 25 defines the parameters for the service.

Table 25 – Unsubscribe Service Parameters

Name	Type	Description
Request		
subscriptionId	SubscriptionId	The identifier for an existing subscription that was returned by the CreateSubscription service.
monitoredItemIds[]	UInt32	Identifiers for subscribed Attributes that were returned by the Subscribe service.
Response		
unsubscribeResult[]	UInt32	Status codes for operation results as defined in Table 26. The order of this list matches the order of the monitoredItemIds request parameter.

5.1.6.8.4.3 Service results

There are no service results other than the common codes specified in 5.1.4.6.

5.1.6.8.4.4 Operation result codes

Table 26 defines values for the operation status code contained in unsubscribeResult. All operational status codes with their description are in Table 9.