

DIN EN 62769-2

ICS 35.240.50

Einsprüche bis 2013-03-07

Entwurf**Feldgeräteintegration (FDI) –
Teil 2: FDI-Client
(IEC 65E/260/CD:2012), Text in Englisch**

Field Device Integration (FDI) –
Part 2: FDI Client
(IEC 65E/260/CD:2012), Text in English

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2013-01-07 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 120 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die englische Originalfassung des internationalen Dokuments IEC 65E/260/CD:2012 „Field Device Integration (FDI) – Part 2: FDI Client“ (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und den nationalen Komitees zur Stellungnahme vorgelegt.

Die 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. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Da der Abstimmungszeitraum für einen FDIS bzw. Schluss-Entwurf prEN nur 2 Monate beträgt, und dann keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der CD als DIN-Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit frühzeitig berücksichtigen zu können.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ 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 IEC 62769-2 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Normentwurfs hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Dieser Normentwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Nationaler Anhang NA (informativ)

3 Begriffe, Abkürzungen und Konventionen

3.1 Begriffe

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

3.1.1

Zugriffssteuerungsdienste

(en: **Access Control Services**)

Untermenge der Gerätezugriffsdienste, über die der Zugriff auf ein Gerät gesteuert wird

3.1.2

Gerätemodelldienste

(en: **Device Model Services**)

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

3.1.3

Direktzugriffsdienste

(en: **Direct Access Services**)

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

3.2 Begriffe für Gerätezugriffsdienste

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe. Einige von ihnen ersetzen die in IEC 62769-1 angegebenen Definitionen.

3.2.1

Attribut

(en: **attribute**)

Informationselement eines Knotens; einige Attribute sind für alle NodeClasses (Knotenklassen) vorhanden, andere sind für eine bestimmte NodeClass spezifisch

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

3.2.2

Gerätemodell

(en: **device model**)

Hierarchie von Knoten, die ein vorhandenes Gerät darstellen

3.2.3

Knoten

(en: **node**)

Element im Gerätemodell, das über die Gerätezugriffsdienste adressiert werden kann

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

3.2.4

NodeClass

(en: **NodeClass**)

entweder ein Objekt oder eine Variable

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

3.2.5

Objekt

(en: **object**)

Instanz des NodeClass-Objektes

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

3.2.6

Variable

(en: **variable**)

Instanz der NodeClass-Variable

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

CONTENTS

FOREWORD.....	10
1 Scope.....	12
2 Normative references	12
3 Terms, definitions, abbreviated terms, acronyms and conventions.....	13
3.1 Terms and definitions	13
3.2 Terms used for Device Access Services	13
3.3 Abbreviated terms and acronyms	14
3.4 Conventions	14
4 Overview	14
5 FDI Client.....	16
5.1 Device Access Services	16
5.1.1 General	16
5.1.2 Device Model.....	16
5.1.3 Node model	18
5.1.4 Services	22
5.1.5 Base Property Services	26
5.1.6 Device Model Services	27
5.1.7 Access Control Services.....	37
5.1.8 Direct Access Services	42
5.1.9 Data types	44
5.2 Hosting Services	48
5.2.1 General	48
5.2.2 Services	48
5.2.3 Parameter Type Definitions	54
6 UIP.....	55
6.1 UIP Services	55
6.1.1 Services	55
6.1.2 Parameter Type Definition TraceLevel	57
6.2 UIP State Machine	57
6.2.1 States.....	57
6.2.2 State Transitions	58
6.3 UIP permissions	58
6.4 UIP downloads from FDI Server	59
7 Actions	60
7.1 General.....	60
7.2 Sequence diagram	61
7.3 FDI Action schema definition	64
8 User Interface Description (UID).....	64
8.1 Overview	64
8.2 UID execution.....	66
9 FDI Client Profiles	68
Annex A (normative) XML schema	70
A.1 AbortRequest	70
A.2 AbortResponse.....	70
A.3 AcknowledgementRequest.....	70

A.4 AcknowledgementResponse	71
A.5 AcknowledgementSeverity	71
A.6 Action	71
A.7 ActionArgument	72
A.8 ActionRequest	72
A.9 ActionResponse	73
A.10 ActionState	74
A.11 Axis	74
A.12 BitEnumeratedType	75
A.13 BitEnumeration	75
A.14 Break	76
A.15 Chart	76
A.16 ChartType	77
A.17 Color	77
A.18 ColorName	77
A.19 ColorValue	78
A.20 DateTimeDataType	78
A.21 DateTimeType	79
A.22 DelayMessageRequest	79
A.23 DelayMessageResponse	80
A.24 Dialog	80
A.25 EditDisplay	80
A.26 EnumeratedType	81
A.27 Enumeration	81
A.28 Graph	81
A.29 Grid	82
A.30 Group	82
A.31 Handling	83
A.32 Image	83
A.33 ImageFormat	84
A.34 InfoRequest	84
A.35 InfoResponse	85
A.36 InputRequest	85
A.37 InputResponse	85
A.38 InputValue	85
A.39 InputValueType	86
A.40 Layout	86
A.41 LimitLine	87
A.42 LineType	88
A.43 ListOfActions	89
A.44 ListOfActionArguments	89
A.45 ListOfBitEnumerations	89
A.46 ListOfButtons	90

A.47 ListOfEnumerations	90
A.48 ListOfIndices	90
A.49 ListOfLimitLines	91
A.50 ListOfOptions	91
A.51 ListOfParameters	91
A.52 ListOfPoints	92
A.53 ListOfPresetValues	92
A.54 ListOfRanges	92
A.55 ListOfSources	93
A.56 ListOfVariants	93
A.57 ListOfVectors	93
A.58 ListOfWaveforms	94
A.59 Menu	94
A.60 NumericDataType	94
A.61 NumericType	95
A.62 Orientation	95
A.63 Page	96
A.64 Parameter	96
A.65 Plugin	97
A.66 Point	97
A.67 PresetValue	98
A.68 Range	98
A.69 Scaling	99
A.70 SelectableItem	99
A.71 SelectionRequest	100
A.72 SelectionResponse	100
A.73 Size	100
A.74 Source	101
A.75 StringDataType	102
A.76 StringType	102
A.77 Table	103
A.78 UIDRequest	103
A.79 UIDResponse	103
A.80 Variant	104
A.81 Vector	104
A.82 Waveform	105
A.83 Window	106
Annex B (informative) Action example	107
Annex C (informative) Typical FDI Client Use Cases	115
C.1 General	115
C.2 Bulk operations	115
C.3 Progress bar support	115
Bibliography	117

Figure 1 – FDI architecture diagram	12
Figure 2 – Overall structure of a Device	17
Figure 3 – Structure of Blocks	17
Figure 4 – DeviceModel NodeClasses	18
Figure 5 – Example: Variable hierarchy representing a RECORD	21
Figure 6 – Variable hierarchy representing a VALUE_ARRAY of RECORD's	22
Figure 7 – UIP state machine	58
Figure 8 – FDI Action sequence diagram	62
Figure 9 – User Interface Descriptions	65
Figure 10 – User Interface Description Sequence Diagram	67
Figure B.1 – Action example (step 1)	109
Figure B.2 – Action example (step 2)	110
Figure B.3 – Action example (step 3)	111
Figure B.4 – Action example (step 4)	112
Figure B.5 – Action example (step 5)	113
Figure B.6 – Action example (step 6)	114
Figure C.1 – Progress bar support	116
Table 1 – BaseNodeClass Attributes	18
Table 2 – Object NodeClass Attributes	18
Table 3 – Variable NodeClass Attributes	19
Table 4 – Additional Attributes for analog Variables	19
Table 5 – Additional Attributes for enumerated Variables	20
Table 6 – Additional Attributes for bit-enumerated Variables	20
Table 7 – Service Definition Table	23
Table 8 – StatusCode Bit Assignments	24
Table 9 – DataValue InfoBits	24
Table 10 – Service Result Codes	25
Table 11 – Operation Level Result Codes	25
Table 12 – GetDeviceAccessInterfaceVersion Service Parameters	26
Table 13 – GetOnlineAccessAvailability Service Parameters	26
Table 14 – Browse Service Parameters	27
Table 15 – CancelBrowse Service Parameters	28
Table 16 – Read Service Parameters	28
Table 17 – Read service result codes	29
Table 18 – Read operation result codes	29
Table 19 – CancelRead Service Parameters	30
Table 20 – Write service parameters	30
Table 21 – Write operation result codes	31
Table 22 – CancelWrite Service Parameters	31
Table 23 – CreateSubscription service parameters	32
Table 24 – CreateSubscription service result codes	32

Table 25 – Subscribe service parameters	33
Table 26 – Subscribe operation result codes.....	34
Table 27 – Unsubscribe Service Parameters.....	35
Table 28 – Unsubscribe operation result codes.....	35
Table 29 – DeleteSubscription Service Parameters.....	35
Table 30 – DataChangeCallback Service Parameters	36
Table 31 – DataChangeCallback result codes	36
Table 32 – InvokeAction service parameters.....	36
Table 33 – InvokeAction service result codes.....	37
Table 34 – CancelInvokeAction Service Parameters	37
Table 35 – InitLock service parameters.....	38
Table 36 – InitLock service result codes	39
Table 37 – ExitLock service parameters.....	39
Table 38 – ExitLock service result codes	39
Table 39 – EnterEditMode service parameters.....	40
Table 40 – EnterEditMode service result codes.....	40
Table 41 – ExitEditMode service parameters	41
Table 42 – ExitEditMode service result codes	41
Table 43 – AbortEditMode service parameters.....	42
Table 44 – AbortEditMode service result codes.....	42
Table 45 – InitDirectAccess service parameters.....	43
Table 46 – InitDirectAccess service result codes	43
Table 47 – ExitDirectAccess service parameters.....	43
Table 48 – ExitDirectAccess service result codes	43
Table 49 – Transfer service parameters.....	44
Table 50 – Transfer service result codes	44
Table 51 – Base data types	44
Table 52 – Identifiers Assigned to Attributes	45
Table 53 – DataValue	46
Table 54 – LocalizedText Definition	46
Table 55 – LocaleId Examples	47
Table 56 – Range Data Type Structure	47
Table 57 – EUInformation Data Type Structure	48
Table 58 – EnumValueType Definition	48
Table 59 – GetClientTechnologyVersion Service Parameters.....	49
Table 60 – GetLocaleSetting Service Parameters	49
Table 61 – OpenUserInterface service parameters.....	50
Table 62 – CloseUserInterface service parameters	50
Table 63 – OpenModalUserInterface service parameters	51
Table 64 – LogAuditTrailMessage service parameters	51
Table 65 – SaveUserSettings service parameters	52
Table 66 – LoadUserSettings service parameters	52
Table 67 – Trace service parameters	52

Table 68 – ShowMessageBox service parameters	53
Table 69 – ShowProgressBar service parameters	53
Table 70 – UpdateShowProgressBar service parameters	54
Table 71 – EndShowProgressBar service parameters	54
Table 72 – DefaultResult definition	55
Table 73 – ButtonSet definition	55
Table 74 – AcknStyle definition	55
Table 75 – Activate service parameters	56
Table 76 – Deactivate service parameters	56
Table 77 – SetSystemLabel service parameters	57
Table 78 – SetTraceLevel service parameters	57
Table 79 – TraceLevel definition	57
Table 80 – UIP States	58
Table 81 - UIP State Transitions	58
Table 82 – FDI Client Profile Conformance Units	69
Table A.1 – Elements of AbortRequest	70
Table A.2 – Elements of AcknowledgementRequest	71
Table A.3 – Enumerations of AcknowledgementSeverity	71
Table A.4 – Elements of Action	72
Table A.5 – Elements of ActionArgument	72
Table A.6 – Elements of ActionRequest	73
Table A.7 – Elements of ActionResponse	74
Table A.8 – Enumerations of ActionState	74
Table A.9 – Elements of Axis	75
Table A.10 – Elements of BitEnumeratedType	75
Table A.11 – Elements of BitEnumeration	76
Table A.12 – Elements of Chart	76
Table A.13 – Enumerations of ChartType	77
Table A.14 – Enumerations of ColorName	78
Table A.15 – Enumerations of DateTimeDataType	79
Table A.16 – Elements of DateTimeType	79
Table A.17 – Elements of DelayMessageRequest	80
Table A.18 – Elements of EditDisplay	80
Table A.19 – Elements of EnumeratedType	81
Table A.20 – Elements of Enumeration	81
Table A.21 – Elements of Graph	82
Table A.22 – Elements of Grid	82
Table A.23 – Enumerations of Handling	83
Table A.24 – Elements of Image	84
Table A.25 – Enumerations of ImageFormat	84
Table A.26 – Elements of InfoRequest	84
Table A.27 – Elements of InputRequest	85
Table A.28 – Elements of InputResponse	85

Table A.29 – Elements of InputValue	86
Table A.30 – Elements of InputValueType	86
Table A.31 – Attributes of Layout	87
Table A.32 – Elements of Layout	87
Table A.33 – Elements of LimitLine	88
Table A.34 – Enumerations of LineType	88
Table A.35 – Elements of ListOfActions	89
Table A.36 – Elements of ListOfActionArguments	89
Table A.37 – Elements of ListOfBitEnumerations	90
Table A.38 – Elements of ListOfButtons	90
Table A.39 – Elements of ListOfEnumerations	90
Table A.40 – Elements of ListOfIndices	91
Table A.41 – Elements of ListOfLimitLines	91
Table A.42 – Elements of ListOfOptions	91
Table A.43 – Elements of ListOfParameters	92
Table A.44 – Elements of ListOfPoints	92
Table A.45 – Elements of ListOfPresetValues	92
Table A.46 – Elements of ListOfRanges	93
Table A.47 – Elements of ListOfSources	93
Table A.48 – Elements of ListOfVariants	93
Table A.49 – Elements of ListOfVectors	94
Table A.50 – Elements of ListOfWaveforms	94
Table A.51 – Enumerations of NumericDataType	95
Table A.52 – Elements of NumericType	95
Table A.53 – Enumerations of Orientation	96
Table A.54 – Attributes of Parameter	96
Table A.55 – Elements of Parameter	97
Table A.56 – Elements of Plugin	97
Table A.57 – Elements of Point	98
Table A.58 – Elements of PresetValue	98
Table A.59 – Elements of Range	99
Table A.60 – Enumerations of Scaling	99
Table A.61 – Elements of SelectableItem	99
Table A.62 – Elements of SelectionRequest	100
Table A.63 – Elements of SelectionResponse	100
Table A.64 – Enumerations of Size	101
Table A.65 – Elements of Source	102
Table A.66 – Enumerations of StringDataType	102
Table A.67 – Elements of StringType	103
Table A.68 – Elements of UIDRequest	103
Table A.69 – Elements of UIDResponse	104
Table A.70 – Elements of Variant	104
Table A.71 – Elements of Vector	105

Table A.72 – Elements of Waveform 106

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**FIELD DEVICE INTEGRATION (FDI) –
Part 2: FDI Client**

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.

International Standard IEC 62769-2 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/xxx/FDIS	65E/xxx/RVD

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

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

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

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

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

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

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.

FIELD DEVICE INTEGRATION (FDI) – Part 2: FDI Client

1 Scope

This International Standard 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 illustration.

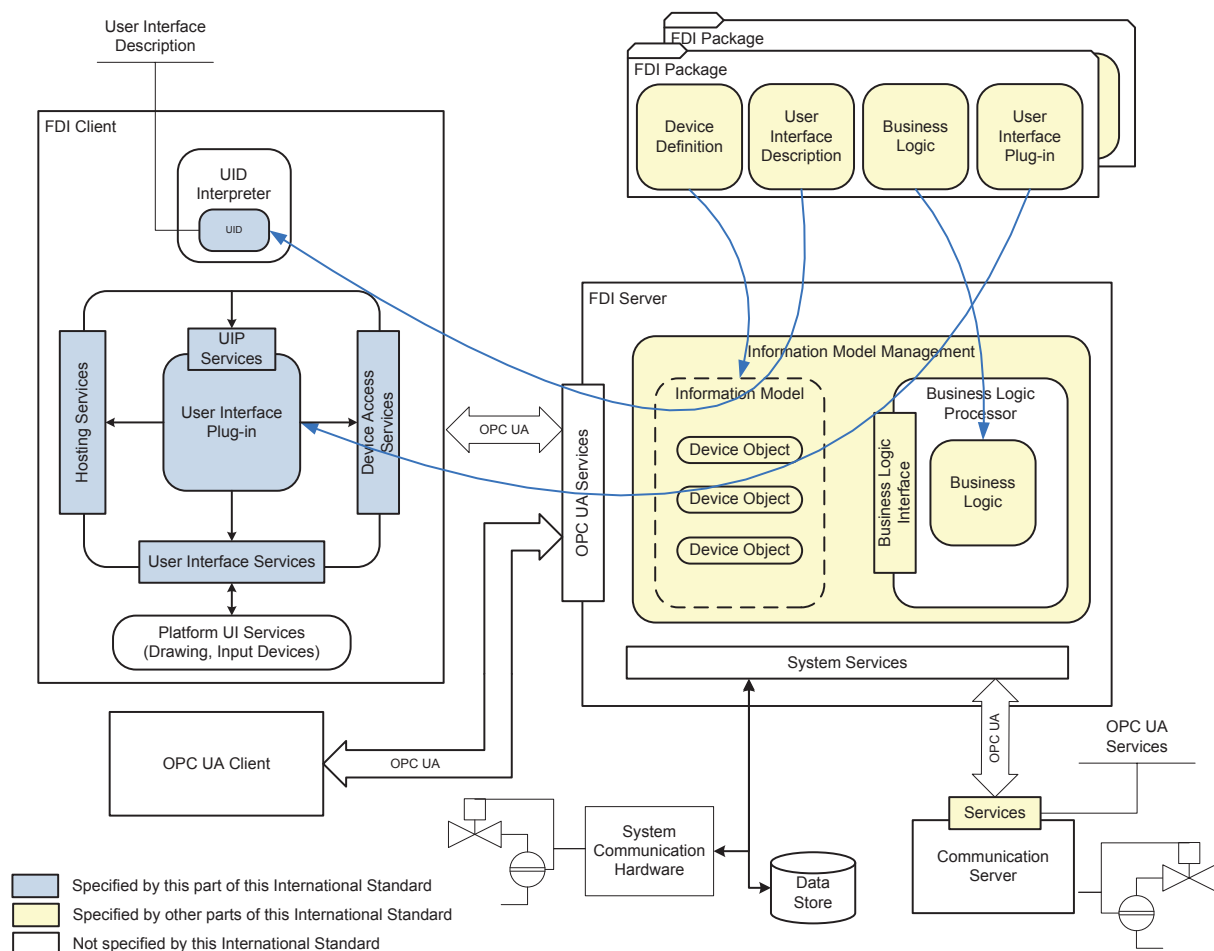


Figure 1 – FDI architecture diagram

2 Normative references

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 62769-1, *Field device integration (FDI) – Part 1: Overview*

NOTE IEC 62769-1 is technical identical to FDI-2021

IEC 62769-3, *Field device integration (FDI) – Part 2: FDI Server*

- 20 NOTE IEC 62769-3 is technical identical to FDI-2023
- 21 IEC 62769-4, *Field device integration (FDI) – Part 2: FDI Packages*
- 22 NOTE IEC 62769-4 is technical identical to FDI-2024
- 23 IEC 62769-5, *Field device integration (FDI) – Part 5: FDI Information Model*
- 24 NOTE IEC 62769-5 is technical identical to FDI-2025
- 25 IEC 62769-6, *Field device integration (FDI) – Part 6: FDI Technology Mapping*
- 26 NOTE IEC 62769-6 is technical identical to FDI-2026
- 27 IEC 62769-7, *Field device integration (FDI) – Part 7: FDI Communication Devices*
- 28 NOTE IEC 62769-7 is technical identical to FDI-2027
- 29 IEC 62541-3:2011, *OPC Unified Architecture - Part 3: Address Space Model*
- 30 IEC 62541-4:2011, *OPC Unified Architecture - Part 4: Services*
- 31 IEC 62541-7:2011, *OPC Unified Architecture - Part 7: Profiles*
- 32 OPC UA Part DI, *OPC Unified Architecture for Devices (DI) Companion Specification*
- 33 XMLSchema-1, *XML Schema: Structures (available at <http://www.w3.org/TR/xmlschema-1/>)*
- 34 XMLSchema-2, *XML Schema: Datatypes (available at <http://www.w3.org/TR/xmlschema-2/>)*
- 35 **3 Terms, definitions, abbreviated terms, acronyms and conventions**
- 36 **3.1 Terms and definitions**
- 37 For the purposes of this document, the terms and definitions given in IEC 62769-1 and the
38 following apply.
- 39 **3.1.1**
- 40 **Access Control Services**
- 41 sub-set of the Device Access Services through which access to a Device is controlled
- 42 **3.1.2**
- 43 **Device Model Services**
- 44 sub-set of the Device Access Services through which a UIP can access the information of a
45 Device
- 46 **3.1.3**
- 47 **Direct Access Services**
- 48 sub-set of the Device Access Services through which a UIP can directly access a Device
- 49 **3.2 Terms used for Device Access Services**
- 50 For the purposes of this document, the following terms and definitions apply. Some of them
51 supersede the definition given in IEC 62769-1.

3.2.1

attribute

information element of a Node; some Attributes exist for all NodeClasses and some are specific to a given NodeClass

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

3.2.2

device model

hierarchy of Nodes that represent an existing Device

3.2.3

node

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

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

3.2.4

NodeClass

either an object or a variable

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

3.2.5

object

instance of the Object NodeClass

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

3.2.6

variable

instance of the Variable NodeClass

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

3.3 Abbreviated terms and acronyms

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

UTC Coordinated Universal Time

XML Extended Mark-up language

3.4 Conventions

Conventions for service definitions are identical to IEC 62541-4:2011, 3.6. Service definitions precede the parameter type definitions.

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

"Parameter" is always an IM 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 of the types. The two types are referred to as User Interface Plug-ins and User Interface Descriptions.

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

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

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

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

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

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

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

- 107 • Device Access Services
- 108 • Hosting Services
- 109 • User Interface Services
- 110 • Printing Services

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

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

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

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

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

124 The User Interface Services provide the means by which the UIP accesses the user interface
125 services of the underlying operating system. These services provide access to the screen,
126 keyboard, mouse, and other operating system resources. The User Interface Services are
127 defined by the chosen implementation technology and therefore no additional definitions are
128 included in this document (see IEC 62769-6:2013, 4.1.2 and 5.1). UIPs shall use the Hosting
129 Services to display message boxes or progress bars and shall never use comparable services
130 provided by the underlying operating system.

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

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

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

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

142 **5 FDI Client**

143 **5.1 Device Access Services**

144 **5.1.1 General**

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

- 147 • browsing the Device Model,
- 148 • reading / writing of data and subscribing to data changes,
- 149 • invoking Actions (EDD methods),
- 150 • controlling access to the Device, and
- 151 • directly communicating with the Device.

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

155 The main services are the Device Model Services to view and access data and execute
156 Actions. The Access Control Services provide for locking and Edit Mode. The Direct Access
157 Services allow a UIP to communicate with the Device.

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

160 **5.1.2 Device Model**

161 The information of a Device corresponds to the structure of its FDI Package information.

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

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

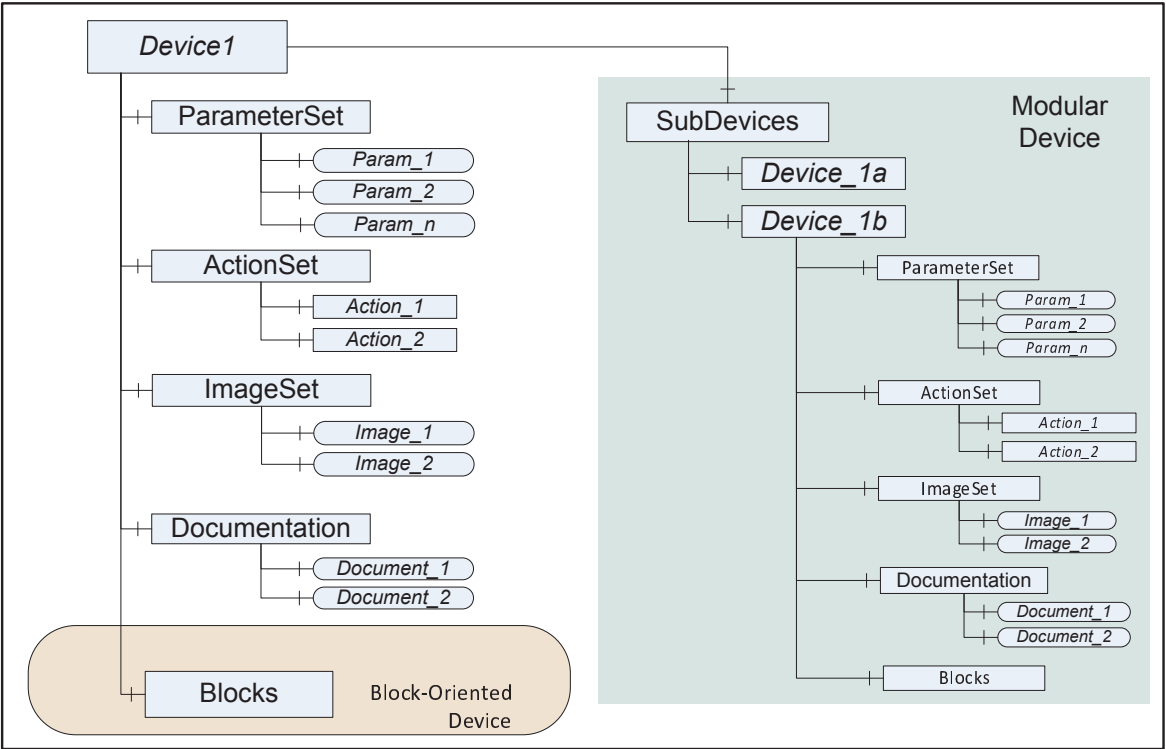


Figure 2 – Overall structure of a Device

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

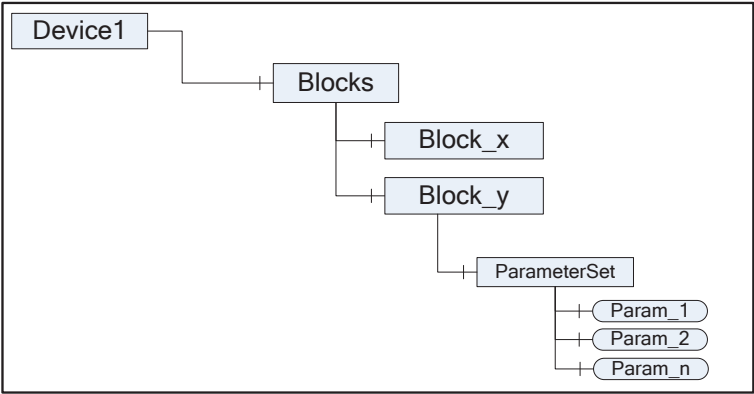


Figure 3 – Structure of Blocks

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/ActionSet/Action_2 -- an Action 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 organised 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.

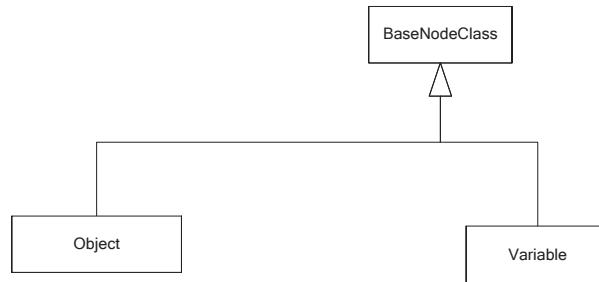


Figure 4 – DeviceModel NodeClasses

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
Path	String	Qualified path of the Node in the Device Model.
NodeID	ByteString	System-specific unique identifier of the Node. This Id can be retrieved either through browsing or by reading the respective Attribute. Use of the NodeID instead of the Path is recommended for repeated access to certain Nodes.
Name	String	Name of Node according to device specific documentation.
Label	String	Human readable label of the Node.
Description	LocalizedText	Optional - Human readable help string describing the Node.

Additional Attributes will be available for derived NodeClasses as well as for certain categories. For example, an analog Variable will have Attributes that define the EURange, the InstrumentRange and the EngineeringUnits.

5.1.3.3 Object NodeClass

Table 2 contains the list of Attributes for Objects beyond those inherited from the Base NodeClass:

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.

211 **5.1.3.4 Variable NodeClass**212 **5.1.3.4.1 General**

213 Variables are used to represent Parameters, images and documents. When reading Variables
214 that represent images or documents the system will provide them as a ByteString.

215 Table 3 contains the list of Attributes for Variables beyond those inherited from the
216 BaseNodeClass:

217 **Table 3 – Variable NodeClass Attributes**

Attribute	Datatype	Description
Value	Variant	The value of the Variable. The Variant is specified in 5.1.9.4.
DataType	UInt32	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: n>1: the Value is an array with the specified number of dimensions. OneDimension (1): The value is an array with one dimension. OneOrMoreDimensions (0): The value is an array with one or more dimensions. Scalar (-1): The value is not an array. Any (-2): The value can be a scalar or an array with any number of dimensions. ScalarOrOneDimension (-3): The value can be a scalar or a one dimensional array.
ArrayDimensions	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: Int32 myArray[346]; 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.

218

219 **5.1.3.4.2 Analog Variable**

220 Table 4 contains the list of Attributes for analog Variable Nodes beyond those inherited from
221 the Variable NodeClass.

222 **Table 4 – Additional Attributes for analog Variables**

Attribute	Datatype	Description
InstrumentRange	Range	Defines the value range that can be returned by the instrument. Example: InstrumentRange ::= {-9999.9, 9999.9} See 5.1.9.3.6 for the definition of the Range data type.
EURange	Range	Defines the value range likely to be obtained in normal operation. It is intended for such use as automatically scaling a bar graph display. 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. See 5.1.9.3.6 for the definition of the Range data type.
EngineeringUnits	EUInformation	EngineeringUnits specifies the units for the value (e.g., °C, hertz, seconds). See 5.1.9.3.7 for the definition of the EUInformation data type.

223

5.1.3.4.3 Enumerated Variable

Table 5 contains the list of Attributes for “enumerated” Variable Nodes beyond those inherited from the Variable NodeClass.

Table 5 – Additional Attributes for enumerated Variables

Attribute	Datatype	Description
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 5.1.9.3.8 for the definition of this type. FDI Clients / UIs may read this Attribute in advance and store it for lookup of name or help when they receive the numeric representation.

5.1.3.4.4 Bit-Enumerated Variable

The value of bit-enumerated Variables represents a bit mask. 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.

The data type of bit-enumerated Variables will be an unsigned integer of a length that is sufficient to represent all bits.

Table 6 contains the list of Attributes for “BitEnumerated” Variable Nodes beyond those inherited from the Variable NodeClass.

Table 6 – Additional Attributes for bit-enumerated Variables

Attribute	Datatype	Description
OptionNames	String[]	A human-readable representation for each valid bit of the bit mask. The array contains an empty String for each bit that has no specific meaning.

5.1.3.4.5 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.

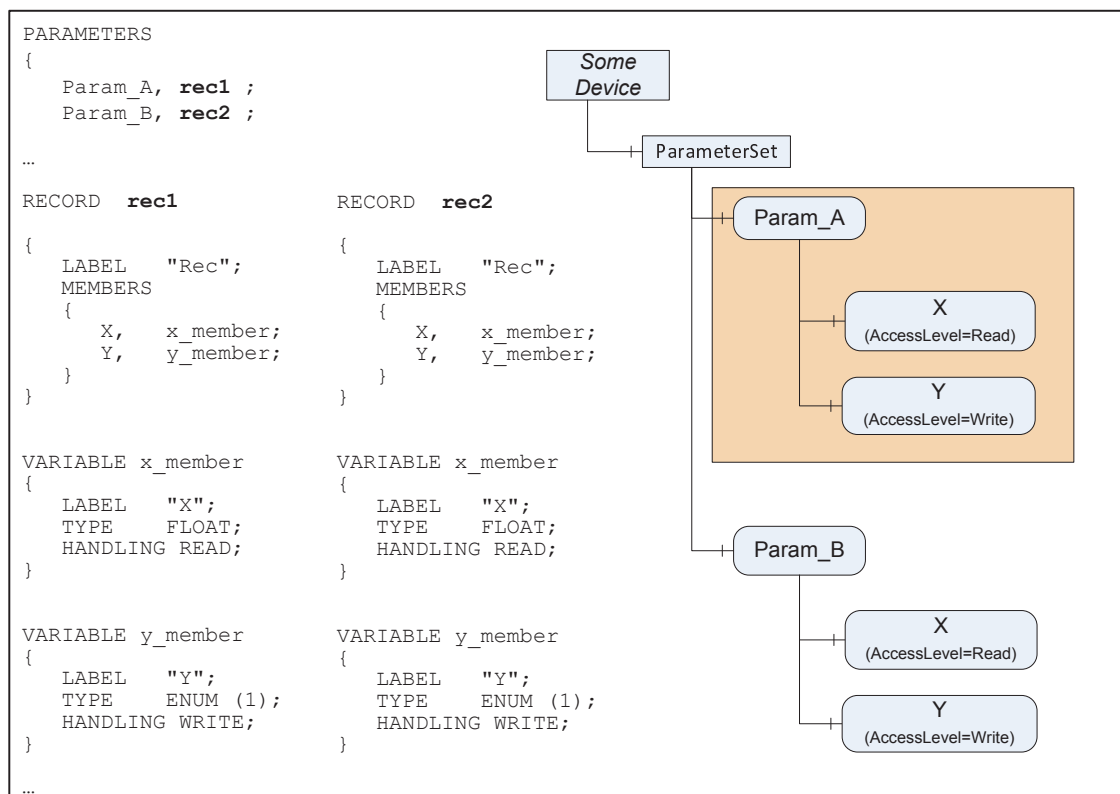


Figure 5 – Example: Variable hierarchy representing a RECORD

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.

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

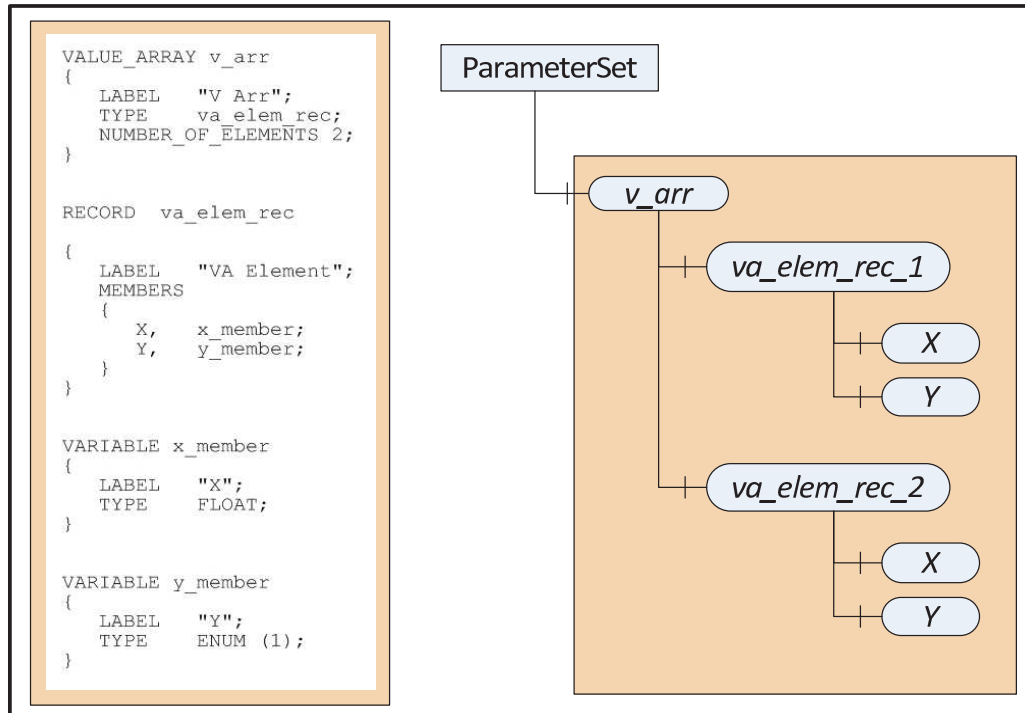
A single Variable will represent an EDDL VALUE_ARRAY or LIST item when the data type of the referenced array element has a simple data type.

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.

269 5.1.3.4.7 Representation of arrays, and lists of RECORDs

270 Value arrays or lists of non-simple EDDL Parameters will be represented as array of Variable
 271 hierarchies. Figure 6 shows the EDDL sample code of a VALUE_ARRAY of RECORD's and
 272 the corresponding Variable hierarchy.



273

274 **Figure 6 –Variable hierarchy representing a VALUE_ARRAY of RECORD's**

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

281 VALUE_ARRAY element, which is in fact a RECORD, is represented as component Variable
 282 hierarchy. The Name and Label Attributes of each root Variable representing a RECORD are
 283 set to the name of the RECORD and the LABEL Attribute, respectively. The array index (_1,
 284 _2) is appended to allow unique identification.

285 The RECORD MEMBERS are also represented as component Variables as specified in
 286 5.1.3.4.5.

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

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

289 5.1.4 Services

290 5.1.4.1 General

291 All services specified in subsequent subclauses rely on the conventions defined in this
 292 subclause. They are specified in an abstract manner with request and response parameters in
 293 a single table.

294 Programmatic access to the services may be synchronous or asynchronous (see
295 IEC 62769-6).

296 5.1.4.2 Conventions for service definitions

297 The service specifications use tables to describe service parameters, as shown in Table 7.
298 Parameters are organised in this table into request parameters and response parameters.

299 **Table 7 – 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

300
301 The Name, Type and Description columns contain the name, data type and description of
302 each parameter. All parameters are mandatory, although some may be unused under certain
303 circumstances. The description column specifies the value to be supplied when a parameter is
304 unused. Parameter names always begin with a lower case character. This allows
305 differentiating if name and type are the same, for example, name = “nodeld”, type = “Nodeld”.

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

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

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

314 5.1.4.3 Auditing

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

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

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

- 322 • They can call the LogAuditTrailMessage service (see 5.2.2.7). This shall be executed in
323 particular when the DirectAccess Services are used. The UIP shall include sufficient
324 information in the message for precise description of the activity performed.
- 325 • They can provide a context text when using the Access Control Services (see 5.1.6.10) or
326 the Direct Access Services (see 5.1.8).

327

328 5.1.4.4 StatusCode

329 The StatusCode used for service results or operational results reports the outcome of an
330 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 8. IEC 62769-6 provides functions to help in evaluation of the StatusCode.

Table 8 – 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: Good 00 The operation was successful; results may be used. Success Uncertain 01 The operation was partially successful; results might not be suitable for some purposes. Warning 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: 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 9 describes the structure of the InfoBits when the Info Type is set to DataValue (01).

Table 9 – 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.

5.1.4.5 Common Service Failure Codes

Table 10 defines common result codes for the services. The 3rd column in the table lists which code may be returned by which service. Result codes that are specific to an individual service are specified with the service. Programmatic access to service failures will be based on exceptions.

349

Table 10 – Service Result Codes

Exception name	Description	Service
Bad_NodeInvalid	The identifier does not refer to a valid Node in the Device Model.	Browse, InvokeAction, InitLock, ExitLock, EnterEditMode, ExitEditMode, AbortEditMode
Bad_NothingToDo	There was nothing to do because the caller passed an empty list of operations.	Read, Write, Subscribe, Unsubscribe
Bad_SubscriptionIdInvalid	The subscription id is not valid.	Subscribe, Unsubscribe, DeleteSubscription
Bad_Timeout	The operation timed out.	All
Bad_InvalidState	The specified node is in a state that does not permit this operation.	Services for Locking and EditMode
Bad_TooManyOperations	The request could not be processed because it specified too many operations.	Read, Write, Subscribe
Bad_UnexpectedError	An unexpected error occurred.	All

350

351 **5.1.4.6 Operational Status Codes**

352 Table 11 defines the status codes for all operation level results (for services that have
 353 individual operations (like Read, Write or Subscribe). Each service defines the applicable
 354 subset and instead of a description will contain references to this table.

355 NOTE Value in Table 11 is referring to Parameter value, not communication quality

356

Table 11 – Operation Level Result Codes

Symbolic ID	Description
Good	The operation completed successfully
Good_LocalOverride	The value has been overridden
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
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
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 that has affected 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.
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.
---------------------------	--

357

358 **5.1.5 Base Property Services**359 **5.1.5.1 Overview**

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

361 **5.1.5.2 Get DeviceAccess Interface Version**

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

363 **5.1.5.2.1 Parameters**

364 Table 12 defines the parameters for the service.

365 **Table 12 – 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 IEC 62769-4.

366

367 **5.1.5.2.2 Service results**

368 This service always succeeds.

369 **5.1.5.3 GetOnlineAccessAvailability**370 **5.1.5.3.1 Description**371 This service returns a hint as to whether online access is available in principle. This is general
372 information. It does not clarify whether online access to a specific device is possible.373 **5.1.5.3.2 Parameters**

374 Table 13 defines the parameters for the service.

375 **Table 13 – 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.

376

377 **5.1.5.3.3 Service results**

378 This service always succeeds.

379 5.1.6 Device Model Services

380 5.1.6.1 Overview

381 The Device Model Services include:

- 382 • Browse
- 383 • Read
- 384 • Write
- 385 • Subscription
 - 386 — CreateSubscription
 - 387 — Subscribe
 - 388 — Unsubscribe
 - 389 — DeleteSubscription
- 390 • InvokeAction

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

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

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

398 5.1.6.2 Browse

399 5.1.6.2.1 Description

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

402 5.1.6.2.2 Parameters

403 Table 14 defines the parameters for the service.

404 **Table 14 – Browse Service Parameters**

Name	Type	Description
Request		
nodeToBrowse	NodeSpecifier	The identifier of the Node to be browsed. See 5.1.9.3.1 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.
nodeId	ByteString	See BaseNodeClass in 5.1.3.
name	String	See BaseNodeClass in 5.1.3.
label	String	See BaseNodeClass in 5.1.3.

405

406 5.1.6.2.3 Service results

407 No specific service result codes are defined for Browse. Common StatusCodes are defined in
 408 Table 10.

409 **5.1.6.3 CancelBrowse**410 **5.1.6.3.1 Description**

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

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

415 **5.1.6.3.2 Parameters**

416 Table 15 defines the parameters for the service.

417 **Table 15 – CancelBrowse Service Parameters**

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

418

419 **5.1.6.3.3 Service results**

420 No specific service result codes are defined for CancelBrowse. Common StatusCodes are
 421 defined in Table 10.

422 **5.1.6.4 Read**423 **5.1.6.4.1 Description**

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

426 **5.1.6.4.2 Parameters**

427 Table 16 defines the parameters for the service.

428 **Table 16 – Read Service Parameters**

Name	Type	Description
Request		
attributesToRead[]	Structure	The Attributes to read
node	NodeSpecifier	The identifier of the Node that contains the Attribute to read. See 5.1.9.3.1 for the definition of the NodeSpecifier type.
attributeld	Attributeld	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, then all elements are to be included in the range. See 5.1.9.3.5 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.

429

430 **5.1.6.4.3 Service results**

431 Table 17 defines values for the service result code. Other common StatusCodes are defined
432 in Table 10.

433 **Table 17 – Read service result codes**

Symbolic ID	Description
Bad_MaxAgeInvalid	The max age parameter is invalid.

434

435 **5.1.6.4.4 Operation result codes**

436 Table 18 defines values for the operation status code contained in the DataValue of each
437 readResult element. All operational status codes with their description are in Table 11.

438 **Table 18 – Read operation result codes**

Symbolic ID
Good
Good_LocalOverride
Uncertain
Uncertain_NoCommunicationLastValue
Uncertain_LastUsableValue
Uncertain_SubstituteValue
Uncertain_InitialValue
Uncertain_SensorNotAccurate
Uncertain_EngineeringUnitsExceeded
Uncertain_SubNormal
Bad
Bad_UserAccessDenied
Bad_ConfigurationError
Bad_NotConnected
Bad_DeviceFailure
Bad_SensorFailure
Bad_OutOfService
Bad_NodeInvalid
Bad_AttributeInvalid
Bad_IndexRangeInvalid
Bad_NotReadable

439

440 **5.1.6.5 CancelRead**441 **5.1.6.5.1 Description**

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

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

5.1.6.5.2 Parameters

Table 19 defines the parameters for the service.

Table 19 – CancelRead Service Parameters

Name	Type	Description
Request		
serviceId	<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. Common StatusCodes are defined in Table 10.

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, to write individual elements or to write ranges of elements.

The service does not return until it writes the values or determines that the value cannot be written. Rollback is the responsibility of the FDI Client / UIP.

NOTE The values have to have the correct data type. No automatic data type translation takes place (see Bad_TypeMismatch operation result code).

5.1.6.6.2 Parameters

Table 20 defines the parameters for the service.

Table 20 – Write service parameters

Name	Type	Description
Request		
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 5.1.9.3.1 for the definition of the NodeSpecifier type.
indexRange	NumericRange	See 5.1.9.3.5 for a detailed definition.
value	Variant	Value to write.
Response		
writeResult[]	UInt32	Status codes for operation results as defined in Table 21. The order of this list matches the order of the variablesToWrite request parameter.

5.1.6.6.3 Service results

There are no service results other than the common codes specified in 5.1.4.5.

5.1.6.6.4 Operation result codes

Table 21 defines values for the operation status code contained in the writeResult elements.

All operational status codes with their description are in Table 11.

Table 21 – Write operation result codes

Symbolic ID
Good
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. Due to asynchronous execution, the service may already be fully or partially completed.

5.1.6.7.2 Parameters

Table 22 defines the parameters for the service.

Table 22 – CancelWrite Service Parameters

Name	Type	Description
Request		
serviceId	<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. Common StatusCodes are defined in Table 10.

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.

Furthermore, 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 23 defines the parameters for the service.

Table 23 – 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 requested rate, a call back 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.

515

5.1.6.8.2.3 Service results

Table 24 defines values for the service result. Common results are defined in Table 10.

Table 24 – CreateSubscription service result codes

Symbolic ID	Description
Bad_TooManySubscriptions	The FDI Server has reached its maximum number of subscriptions.

519

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.

524 **5.1.6.8.3.2 Parameters**

525 Table 25 defines the parameters for the service.

526 **Table 25 – Subscribe service parameters**

Name	Type	Description
Request		
subscriptionId	SubscriptionId	The identifier for an existing subscription that was returned by the CreateSubscription service.
monitoredItemsToAdd[]	structure	The Attributes to add to the subscription.
node	NodeSpecifier	The identifier of the Node that contains the Attribute to subscribe. See 5.1.9.3.1 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 5.1.9.3.5 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 paired with the data in the DataChangeCallback service so that the UIP can easily associate each changed value with the subscribed Node Attribute. <i>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).</i>
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 26.
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 capabilities of the underlying system.

527

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

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

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