DIN EN IEC 62453-71

ICS 25.040.40; 35.160; 35.240.50 Einsprüche bis 2022-04-25

Entwurf

Field Device Tool (FDT)-Schnittstellenspezifikation – Teil 71: OPC UA Informationsmodell für FDT (IEC 65E/766/CD:2020); Text Englisch

Field device tool (FDT)-interface specification – Part 71: OPC UA Information Model for FDT (IEC 65E/766/CD:2020); Text in English

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2022-02-25 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

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

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe 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@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe 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 in DIN und VDE, Merianstraße 28, 63069 Offenbach 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 60 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Nationales Vorwort

Das internationale Dokument IEC 65E/766/CD:2020-11 "Field device tool (FDT) interface specification – Part 71: OPC UA Information Model for FDT" (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden.

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

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. 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 FprEN nur 6 Wochen 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 DKE/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.

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 den vorliegenden Normentwurf erfüllt.

Da sich die Benutzer des vorliegenden Normentwurfs der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der IEC 65E/766e/CD veröffentlicht. Zu deren Einleitung, Abschnitt 1, der den Anwendungsbereich festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Norm-Entwurfs 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 Norm-Entwurf 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.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. IEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Nationaler Anhang NA
(informativ)

Auszugsweise Übersetzung

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

EINLEITUNG

0.1 Allgemeines

Die neue OPC Unified Architecture (OPC UA, Vereinheitlichte Architektur) vereinheitlicht die bestehenden Normen und bringt sie durch eine serviceorientierte Architektur (SOA, en: Service Oriented Architecture) auf den neuesten Stand der Technik. Plattformunabhängige Technologie ermöglicht den Einsatz von OPC UA über die derzeitigen OPC-Anwendungen hinaus, die nur auf Windows-basierten PC-Systemen laufen. Die OPC UA kann sowohl auf eingebetteten Systemen als auch auf Linux-/ UNIX-basierten Unternehmenssystemen laufen. Die bereitgestellten Informationen können generisch modelliert werden, so dass beliebige Informationsmodelle mit OPC UA bereitgestellt werden können.

FDT standardisiert die Kommunikations- und Konfigurationsschnittstelle zwischen allen Feldgeräten und Hostsystemen. FDT bietet eine gemeinsame Umgebung für den Zugriff auf die anspruchsvollsten Funktionen der Geräte. Jedes Gerät kann über die standardisierte Benutzeroberfläche konfiguriert, bedient und gewartet werden – unabhängig von Hersteller, Typ oder Kommunikationsprotokoll.

Diese internationale Norm spezifiziert eine Synergie beider Ansätze und ermöglicht so einen einfachen, standardisierten Zugriff über OPC UA-Schnittstellen auf die Geräte-Fachkenntnis, die auf der Grundlage von FDT bereitgestellt wird.

0.2 Einführung in FDT

FDT ist eine Technologie, die den Datenaustausch zwischen Feldgeräten und Automatisierungssystemen unterstützt. Die Technologie basiert auf einer als IEC 62453 genormten Schnittstellenspezifikation. Die Spezifikation definiert zwei Hauptkonzepte: Device Type Manager (DTM, Gerätetypmanager) und Frame Application (Rahmenapplikation). Ein DTM ist eine Softwarekomponente, die für einen Feldgerätetyp spezifisch ist. Eine Frame Application ist eine Softwareumgebung (Teil des Automatisierungssystems) für die Integration von DTMs. Innerhalb einer Frame Application stellt jeder DTM Daten und Dienste bereit, die für das jeweilige Feldgerät spezifisch sind. Da die Technologie auf einem standardisierten Satz von Schnittstellen basiert, kann jeder DTM in jede Frame Application integriert werden. Basierend auf der FDT ist es möglich, Kommunikationsgeräte, Kommunikationsinfrastrukturgeräte (z. B. Gateways) und Feldgeräte, abhängig von ihren Kommunikationsprotokollen, zu integrieren. Die Unterstützung verschiedener Kommunikationsprotokolle erfolgt durch ergänzende Kommunikationsprotokollspezifikationen (z. B. für PROFINET, PROFIBUS, Ethernet IP, TCP, HART und FF) oder durch herstellerspezifischer Protokollintegration.

0.3 Einführung in die OPC Unified Architecture

Der Hauptanwendungsfall für OPC-Normen ist der Online-Datenaustausch zwischen Geräten und HMI- oder SCADA-Systemen unter Verwendung der Data Access (Datenzugriffs-) Funktionalität. In diesem Anwendungsfall werden die Gerätedaten von einem OPC-Server bereitgestellt und von einem in das HMI- oder SCADA-System integrierten OPC-Client abgefragt. OPC DA bietet Funktionen zum Durchsuchen eines hierarchischen Namensraums, der Datenelemente enthält, und zum Lesen, Schreiben und Überwachen dieser Elemente auf Datenänderungen. Die OPC Classic-Spezifikationen basieren auf der Microsoft COM/DCOM-Technologie für die Kommunikation zwischen Softwarekomponenten verschiedener Zulieferer. Daher sind OPC Classic Server und Clients auf Automatisierungssysteme beschränkt, die auf Windows OS basieren.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Die OPC UA beinhaltet alle Funktionen der OPC Class-Spezifikationen wie OPC DA, A&E und HDA, definiert jedoch plattformunabhängige Kommunikationsmechanismen und generische, erweiterbare und objektorientierte Modellierungsmöglichkeiten für die Informationen, die ein System offenlegen möchte.

Der OPC UA-Netzwerkkommunikationsteil definiert verschiedene Mechanismen, die für unterschiedliche Anwendungsfälle optimiert sind. Die erste Version der OPC UA definiert ein optimiertes binäres TCP-Protokoll für die Hochleistungs-Intranet-Kommunikation sowie ein Mapping auf akzeptierte Internet-Standards wie Webdienste. Das abstrakte Kommunikationsmodell ist nicht von einem bestimmten Protokollmapping abhängig und ermöglicht das Hinzufügen neuer Protokolle in der Zukunft. Merkmale wie Sicherheit, Zugangskontrolle und Zuverlässigkeit sind direkt in die Transportmechanismen integriert. Auf der Grundlage der Plattformunabhängigkeit der Protokolle können OPC UA-Server und Clients direkt in Geräte und Steuerungen integriert werden.

Das OPC UA-Informationsmodell bietet einen Standardweg für Server, um Objekte für Clients verfügbar zu machen. Objekte im Sinne der OPC UA setzen sich aus anderen Objekten, Variablen und Methoden zusammen. Die OPC UA erlaubt auch die Darstellung von Beziehungen zu anderen Objekten.

Der Satz von Objekten und zugehörigen Informationen, die ein OPC UA- Server den Clients zur Verfügung stellt, wird als dessen AddressSpace (Adressraum) bezeichnet. Die Elemente des OPC UA-Objektmodells werden im AddressSpace als eine Menge von Nodes (Knoten) dargestellt, die durch Attribute beschrieben und durch Referenzen miteinander verbunden sind. Die OPC UA definiert acht Klassen von Nodes zur Darstellung von AddressSpace-Komponenten. Die Klassen sind Object, Variable, Method, ObjectType, DataType, ReferenceType und View (Objekt, Variable, Methode, Objekttyp, Datentype, Referenztyp und Ansicht). Jede NodeClass (Knotenklasse) hat einen definierten Satz von Attributen.

Diese Spezifikation nutzt zwei wesentliche OPC UA NodeClasses: Objekte und Variablen.

Objekte werden verwendet, um Komponenten eines Systems darzustellen. Einem Objekt ist ein entsprechender ObjectType zugeordnet, der Definitionen für dieses Objekt bereitstellt.

Variablen werden zur Darstellung von Werten verwendet. Es werden zwei Kategorien von Variablen definiert: Properties (Eigenschaften) und DataVariables (Datenvariablen).

Properties sind vom Server definierte Merkmale von Objekten, DataVariables und anderen Nodes. Für Eigenschaften dürfen keine Eigenschaften definiert sein. Ein Beispiel für Eigenschaften von Objekten ist die Eigenschaft „Revision Property of a DeviceType“ (Änderungseigenschaft eines Gerätetyps).

DataVariables stellen den Inhalt eines Objekts dar. DataVariables können DataVariables-Komponenten enthalten. Dies wird typischerweise von Servern verwendet, um einzelne Elemente von Reihen (arrays) und Strukturen freizulegen. Diese Spezifikation verwendet DataVariables zur Darstellung von Daten wie den Prozessvariablen, die von einem Gerät bereitgestellt werden.

0.4 Einführung in die OPC UA-Geräteintegration

Die Spezifikation „OPC UA Device Integration“ (OPC UA-Geräteintegration) ist eine Erweiterung der gesamten OPC Unified Architecture-Spezifikationsreihe und definiert das mit Devices im Zusammenhang stehende Informationsmodell. Das Modell soll eine einheitliche Sicht auf Devices bieten, unabhängig von den zugrunde liegenden Device-Protokollen. FDT befasst sich mit physikalischen oder logischen Devices und das Informationsmodell von OPC UA für Geräte wird daher als Grundlage für das FDT-Informationsmodell verwendet.

Das Informationsmodell für Devices spezifiziert verschiedene ObjectTypes und Prozeduren, die verwendet werden, um Geräte und zugehörige Komponenten wie die Kommunikationsinfrastruktur in einem OPC UA-Adressraum darzustellen. Die Hauptanwendungsfälle sind Gerätekonfiguration und -diagnose, aber es ermöglicht einen allgemeinen und standardisierten Weg für jede Art von Anwendung, auf gerätebezogene Informationen zuzugreifen. Die folgenden Beispiele verdeutlichen die in dieser Spezifikation verwendeten Konzepte. Zur vollständigen Definition des Informationsmodells für Devices siehe UA Devices.

Figure zeigt ein Beispiel für einen Temperaturregler, der als Device Object (Geräteobjekt) dargestellt wird. Die Komponente ParameterSet (Parametereinstellung) enthält alle Variablen, die das Device beschreiben. Die

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Komponente MethodSet (Methodeneinstellung) enthält alle vom Device bereitgestellten Methoden. Beide Komponenten werden vom TopologyElementType (Topologieelementtyp) übernommen, dem Wurzelobjekttyp der Typenhierarchie der Device Objects. Objekte vom Typ FunctionalGroupType (Funktionsgruppentyp) werden verwendet, um die Parameter und Methoden des Geräts in logischen Gruppen zu gruppieren. Der FunctionalGroupType und das Gruppierungskonzept sind in UA Devices definiert, aber die Gruppen sind gerätetypspezifisch, d. h. die Gruppen ProcessData (Prozessdaten) und Configuration (Konfiguration) sind in diesem Beispiel durch den TemperatureControllerType (Temperaturreglertyp) definiert.

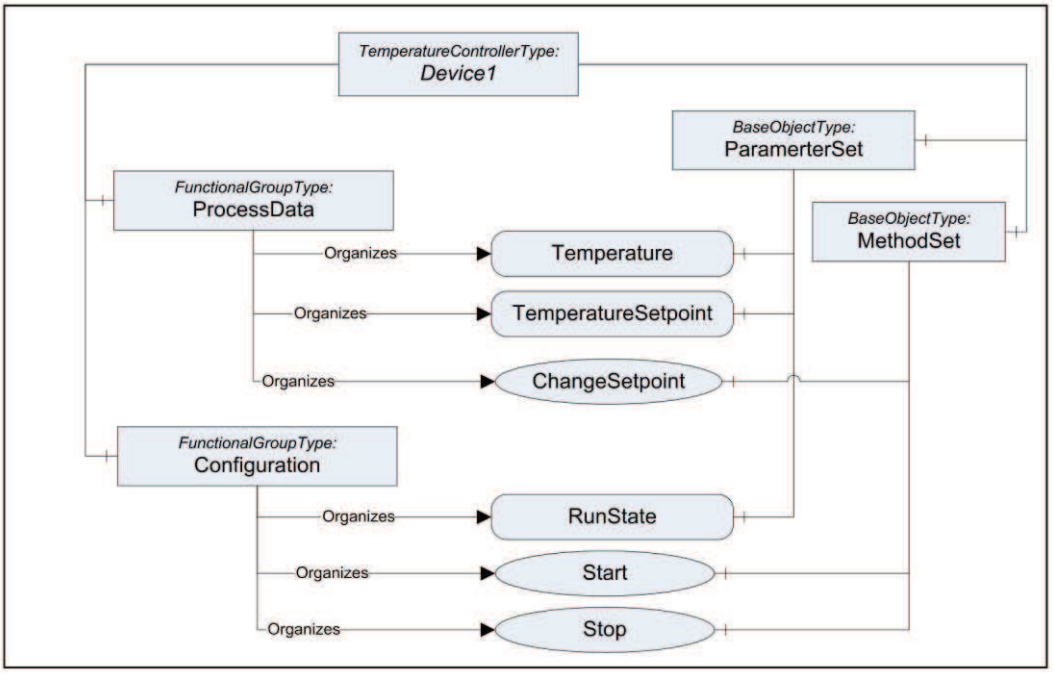


Bild 1 – Beispiel für OPC UA-Geräte

Die Anwendungsfälle in Anhang B stellen die Anwendung des Informationsmodells dar. Nicht alle erforderlichen Objekte müssen innerhalb eines konkreten OPC UA Servers realisiert werden.

1 Anwendungsbereich

Dieser Teil von IEC 62453 legt ein OPC UA-Informationsmodell zur Repräsentation der Geräte-Informationen auf Grundlage der FDT-definierten Geräte-Integration fest.

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die in IEC 62453-1, IEC 62453-2 sowie in IEC 62451-100 aufgeführten Begriffe und Definitionen.

AVA Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Inhalt

Seite

0.1 General ... 11
0.2 Introduction to FDT ... 11
0.3 Introduction to OPC Unified Architecture ... 11
0.4 Introduction to OPC UA Device Integration ... 12
1 Scope ... 14
2 Normative references ... 14
3 Terms, definitions, abbreviations, and conventions ... 14
3.1 Terms and definitions ... 14
3.2 Abbreviations and symbols ... 14
3.3 Conventions used in this document ... 15
3.3.1 Document conventions ... 15
3.3.2 Conventions for FDT methods ... 15
3.3.3 Conventions for Node descriptions ... 15
3.3.4 NodeIds and BrowseNames ... 16
3.3.5 Common Attributes ... 17
3.3.6 Graphical Notation ... 18
4 Concept ... 20
4.1 System Architecture ... 20
4.2 Handling of OPC UA namespaces ... 20
5 FDT specific OPC UA datatypes ... 21
5.1 General ... 21
5.2 Object Types ... 21
5.2.1 FdtDeviceType ... 21
5.2.2 FdtDeviceClassificationType ... 22
5.2.3 Documentation ... 23
5.2.4 FdtFunctionalGroupType ... 24
5.2.5 FdtProtocolType ... 24
5.2.6 FdtTransferServiceType ... 25
5.2.7 FdtIoSignalInfoType ... 26
5.3 Variable types ... 26
5.3.1 FdtParameter ... 26
5.3.2 DataRefType ... 27
5.3.3 SemanticInfoType ... 27
5.4 Enumeration datatypes ... 28
5.4.1 AlarmType ... 28
5.4.2 ApplicationIdEnumeration ... 28
5.4.3 ClassificationDomainId ... 29
5.4.4 ClassificationId ... 29
5.4.5 DocumentClassification ... 31

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

5.4.6 RangeType 31
5.4.7 SubstitutionType 32
5.4.8 SupportedTransfer 32
5.4.9 FunctionExecutionResultState 32
5.4.10 FdtIecDataType 33
5.4.11 SignalTypeEnumeration 33
5.5 Reference types 34
5.5.1 HasIOSignalRef 34
5.6 Event types related to audit trail 34
5.6.1 Overview 34
5.6.2 FdtAuditEventType 34
5.6.3 FdtStartMethodEventType 35
5.6.4 FdtEndMethodEventType 35
5.6.5 FdtAuditWriteUpdateEventType 35
6 Mapping of DataTypes 36
6.1 Primitive data types 36
6.1.1 DeviceHealth 36
6.2 Mapping to OPC DI Types 36
6.2.1 General 36
6.2.2 Device Type 36
6.2.3 TopologyElementType 41
6.2.4 FunctionalGroupType 42
6.2.5 Identification FunctionalGroup 43
6.2.6 Device Data and Device Methods 43
6.2.7 Methods 45
6.2.8 Variable 47
6.2.9 Device Support Information 50
6.2.10 Mapping to FdtIoSignalInfoType 51
6.2.11 ProtocolType 52
Anhang A (normative) FDT Namespace and Mappings 53
A.1 Namespace and identifiers for FDT Information Model 53
Anhang B (informative) Use cases 54
B.1 General 54
B.2 Use case: List topology 54
B.3 Use case: Identify device 55
B.4 Get list of available device parameters 55
B.4.1 Use case: Browse device parameters 55
B.4.2 Use case: Get attributes of a device parameter 56
B.5 Use case: Get Device Status 57
B.6 Use case: Get Device Diagnostics 58

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

B.7 Read parameters 58
B.7.1 Use case: Read offline data 58
B.7.2 B.7.2Use case: Read online data 59
B.8 Use case: Write device parameters 59
B.9 Use case: Audit trail 60

Bilder

Bild 1 – Beispiel für OPC UA-Geräte 5
Figure 1 – OPC UA Devices Example 13
Figure 2 – OPC UA Graphical Notation for NodeClasses 18
Figure 3 – OPC UA Graphical Notation for References 19
Figure 4 – OPC UA Graphical Notation Example 19
Figure 5 – System architecture according to IEC 62453-42 20
Figure 6 – FdtDeviceType overview 22
Figure 7 – FdtProtocolType overview 25
Figure 8 – FdtTransferServiceType overview 25
Figure 9 – FdtIoSignalInfoType overview 26
Figure 10 – Audit event type overview 34
Figure 11 – Example for sources of DeviceType information 37
Figure 12 – Example for sources of TopologyType information 41
Figure 13 – Example for mapping of data and function information 44
Figure 14 – Example for source of function information 45
Figure 15 – Example for source of static function information 46

Tabellen

Table 1 – Type Definition Table 15
Table 2 – Examples of DataTypes 16
Table 3 – Common Node Attributes 17
Table 4 – Common Object Attributes 17
Table 5 – Common Variable Attributes 18
Table 6 – Common VariableType Attributes 18
Table 7 – Namespaces used in an FDT OPC UA Server 21
Table 8 – Namespaces used in this specification 21
Table 9 – FdtDeviceType Definition 22
Table 10 – FdtDeviceClassificationType Definition 22
Table 11 – Documentation Definition 23
Table 12 – FdtDocumentType Definition 23
Table 13 – FdtDocumentFile Definition 24
Table 14 – FdtDocumentUrl Definition 24
Table 15 – FdtFunctionalGroupType Definition 24
Table 16 – FdtProtocolType Definition 25
Table 17 – FdtTransferServiceType Definition 25

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Table 18 – FdtIoSignalInfoType Definition 26
Table 19 – FdtParameter Definition 27
Table 20 – DataRefType Definition 27
Table 21 – SemanticInfoType Definition 28
Table 22 – AlarmType Values 28
Table 23 – AlarmType Definition 28
Table 24 – ApplicationIdEnumeration Values 29
Table 25 – ApplicationIdEnumeration Definition 29
Table 26 – ClassificationDomainId Values 29
Table 27 – ClassificationDomainId Definition 29
Table 28 – ClassificationId Values 30
Table 29 – ClassificationId Definition 31
Table 30 – DocumentClassification Values 31
Table 31 – DocumentClassification Definition 31
Table 32 – RangeType Values 32
Table 33 – RangeType Definition 32
Table 34 – SubstitutionType Values 32
Table 35 – SubstitutionType Definition 32
Table 36 – SupportedTransfer Values 32
Table 37 – SupportedTransfer Definition 32
Table 38 – FunctionExecutionResultState Values 33
Table 39 – FunctionExecutionResultState Definition 33
Table 40 – FdtIecDataType Values 33
Table 41 – FdtIecDataType Definition 33
Table 42 – SignalTypeEnumeration Values 34
Table 43 – FdtIecDataType Definition 34
Table 44 – FdtAuditEventType Definition 35
Table 45 – FdtStartMethodEventType Definition 35
Table 46 – FdtEndMethodEventType Definition 35
Table 47 – FdtAuditWriteUpdateEventType Definition 35
Table 48 – Mapping for DeviceHealth 36
Table 49 – DeviceType mapping 37
Table 50 – Device information mapping 39
Table 51 – Offline device parameter mapping 40
Table 52 – Online device parameter mapping 40
Table 53 – TopologyElementType mapping 42
Table 54 – FunctionalGroupType mapping 43
Table 55 – Identification FunctionalGroup mapping 43
Table 56 – Method node information mapping 46
Table 57 – Method node information mapping for Static Function 47

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Table 58 – TransferService mapping 47
Table 59 – Mapping of FDT data items 48
Table 60 – FdtParameter mapping 48
Table 61 – Mapping of simple data types 49
Table 62 – EUinformation mapping 50
Table 63 – Device Type Image mapping 50
Table 64 – ProtocolSupport mapping 51
Table 65 – ProtocolSupport Folder 51
Table 66 – FdtIoSignalInfoType node information mapping 51
Table 67 – ProtocolType node information mapping 52

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

INTRODUCTION

0.1 General

The new OPC Unified Architecture (OPC UA) unifies the existing standards and brings them to state-of-the-art technology using service-oriented architecture (SOA). Platform-independent technology allows the deployment of OPC UA beyond current OPC applications only running on Windows-based PC systems. OPC UA can also run on embedded systems as well as Linux / UNIX based enterprise systems. The provided information can be generically modelled and therefore arbitrary information models can be provided using OPC UA.

FDT standardizes the communication and configuration interface between all field devices and host systems. FDT provides a common environment for accessing the devices' most sophisticated features. Any device can be configured, operated, and maintained through the standardized user interface – regardless of supplier, type or communication protocol.

This international standard specifies a synergy of both approaches, thus allowing easy, standardized access via OPC UA interfaces to device know-how provided on base of FDT.

0.2 Introduction to FDT

FDT is a technology supporting the data exchange between field devices and automation systems. The technology is based on an interface specification standardized as IEC 62453. The specification defines two main concepts: Device Type Manager (DTM) and Frame Application. A DTM is a software component specific to a field device type. A Frame Application is a software environment (part of the automation system) for integration of DTMs. Within a Frame Application every DTM provides data and services specific to the respective field device. Since the technology is based on a standardized set of interfaces, every DTM may be integrated in every Frame Application. Based on FDT it is possible to integrate communication devices, communication infrastructure devices (e.g. gateways) and field devices, depending on their communication protocols. Support for different communication protocols is provided by means of supplemental communication protocol specifications (e.g. for PROFINET, PROFIBUS, Ethernet IP, TCP, HART and FF) or by means of manufacturer-specific protocol integration.

0.3 Introduction to OPC Unified Architecture

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

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

The OPC UA network communication part defines different mechanisms optimized for different use cases. The first version of OPC UA is defining an optimized binary TCP protocol for high performance intranet communication as well as a mapping to accepted internet standards like Web Services. The abstract communication model does not depend on a specific protocol mapping and allows adding new protocols in the future. Features like security, access control and reliability are directly built into the transport mechanisms. Based on the platform independence of the protocols, OPC UA servers and clients can be directly integrated into devices and controllers.

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

The set of Objects and related information that an OPC UA Server makes available to Clients is referred to as its AddressSpace. The elements of the OPC UA Object Model are represented in the AddressSpace as a set of Nodes described by Attributes and interconnected by References. OPC UA defines eight classes of Nodes

AVA Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

to represent AddressSpace components. The classes are Object, Variable, Method, ObjectType, DataType, ReferenceType and View. Each NodeClass has a defined set of Attributes.

This specification makes use of two essential OPC UA NodeClasses: Objects and Variables.

Objects are used to represent components of a system. An Object is associated with a corresponding ObjectType that provides definitions for that Object.

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

Properties are Server-defined characteristics of Objects, DataVariables and other Nodes. Properties are not allowed to have Properties defined for them. An example for Properties of Objects is the Revision Property of a DeviceType.

DataVariables represent the contents of an Object. DataVariables may have component DataVariables. This is typically used by Servers to expose individual elements of arrays and structures. This specification uses DataVariables to represent data like the process variables provided by a device.

0.4 Introduction to OPC UA Device Integration

The specification “OPC UA Device Integration” is an extension of the overall OPC Unified Architecture specification series and defines the information model associated with Devices. The model is intended to provide a unified view of Devices irrespective of the underlying Device protocols. FDT deals with physical or logical Devices and the information model of OPC UA for Devices therefore is used as base for the FDT information model.

The Devices information model specifies different ObjectTypes and procedures used to represent devices and related components like the communication infrastructure in an OPC UA Address Space. The main use cases are device configuration and diagnostic, but it allows a general and standardized way for any kind of application to access device related information. The following examples illustrate the concepts used in this specification. See UA Devices for the complete definition of the Devices information model.

Figure shows an example for a temperature controller represented as Device Object. The component ParameterSet contains all Variables describing the Device. The component MethodSet contains all Methods provided by the Device. Both components are inherited from the TopologyElementType which is the root Object type of the Device Object type hierarchy. Objects of the type FunctionalGroupType are used to group the Parameters and Methods of the Device into logical groups. The FunctionalGroupType and the grouping concept are defined in UA Devices but the groups are device type specific i.e. the groups ProcessData and Configuration are defined by the TemperatureControllerType in this example.

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

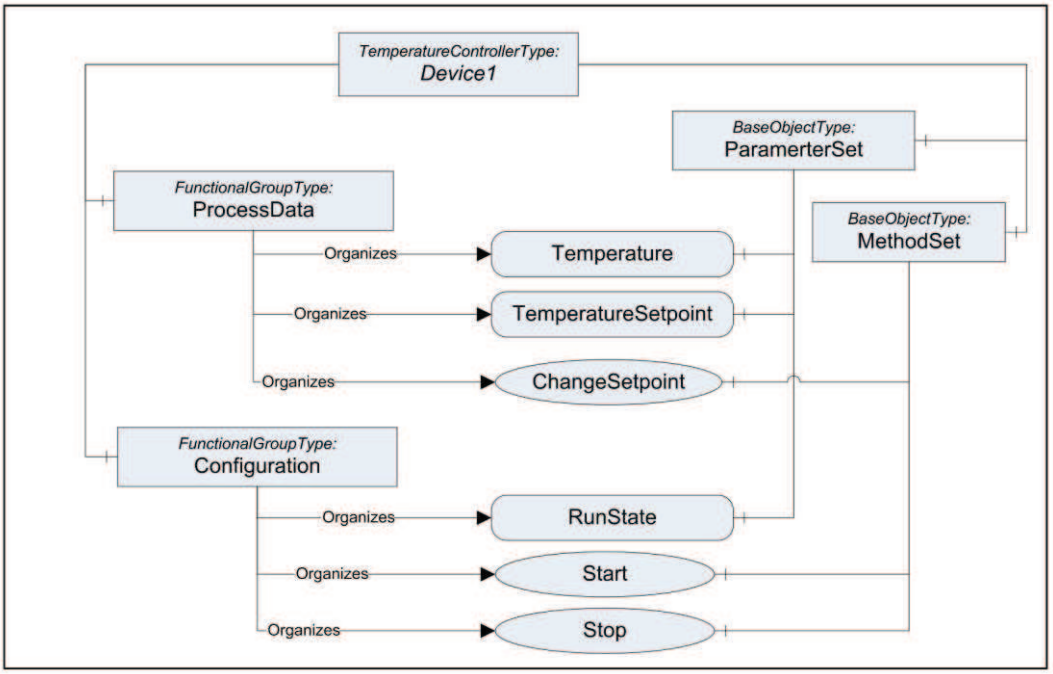


Figure 1 – OPC UA Devices Example

The use cases in Annex B illustrate the usage of the information model. Not all necessary Objects must be realized within a concrete OPC UA Server.

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 71: OPC UA Information Model for FDT

1 Scope

This part of IEC 62453 specifies an OPC UA Information Model to represent the device information based on FDT-defined device integration.

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 62453-1:—[1], *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:—[1], *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

IEC 62541-100:2015, *OPC UA Specification – Part 100: Device Interface*

3 Terms, definitions, abbreviations, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62453-1, IEC 62453-2 as well as in IEC 62451-100 apply.

3.2 Abbreviations and symbols

For the purposes of this document, the abbreviations given in IEC 62453-1, IEC 62453-2 as well as the following apply.

A&E	Alarms & Events
DA	Data Access
DTM	Device Type Manager
FDT	Field Device Technology
HDA	Historical Data Access
HMI	Human-Machine Interface
IEC	International Electrotechnical Commission
MES	Manufacturing Execution System
UA	Unified Architecture
XML	Extensible Markup Language

[1] To be published concurrently with this international standard.

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

3.3 Conventions used in this document

3.3.1 Document conventions

Throughout this document certain document conventions are used.

Italics are used to denote a defined term or definition that appears in the “Terms and definition” clause in this document or in one of the referenced OPC UA documents.

Italics are also used to denote the name of a service input or output parameter or the name of a structure or element of a structure that are usually defined in tables.

3.3.2 Conventions for FDT methods

FDT defines synchronous methods and asynchronous methods. Asynchronous methods are implemented with a set of methods. For example an asynchronous method <MethodName> is implemented with Begin<MethodName>(), Cancel<MethodName>(), and End<MethodName>().

In this document asynchronous methods are indicated by ‘<>’ in front of the name (e.g. <>MethodName).

3.3.3 Conventions for Node descriptions

Node definitions are specified using tables (See Table 1)

Table 1 – Type Definition Table

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

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

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

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

AVA Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

"{ScalarOrOneDimension}" and the ValueRank is set to the corresponding value (see UA Part 3) and the ArrayDimensions is set to null or is omitted. In Table 2 examples are given.

Table 2 – Examples of DataTypes

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

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

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

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

Components of Nodes can be complex, i.e. containing components by themselves. The TypeDefinition, NodeClass, DataType and ModellingRule can be derived from the type definitions, and the symbolic name can be created as defined in 3.3.4.1. Therefore, those containing components are not explicitly specified; they are implicitly specified by the type definitions.

If components of a Node are defined in other OPC UA namespaces, the namespaces are referenced by their default index. For more explanation see 5.2.

3.3.4 NodeIds and BrowseNames

3.3.4.1 NodeIds

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

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

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

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

Note: This specification does not only define concrete Nodes, but also requires that some Nodes have to be generated, for example one for each device type available in the frame application. The NodeIds of those Nodes are server-specific, including the Namespace. But the NamespaceIndex of those Nodes cannot be the NamespaceIndex used for the Nodes defined by this specification, because they are not defined by FDT but generated by the Server.

3.3.4.2 BrowseNames

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

3.3.5 Common Attributes

3.3.5.1 General

For all Nodes specified in this part, the Attributes named in Table 3 shall be set as specified in the table.

Table 3 – Common Node Attributes

Attribute	Value
DisplayName	The DisplayName is a LocalizedText. Each server shall provide the DisplayName identical to the BrowseName of the Node for the LocaleId “en”. Whether the server provides translated names for other LocaleIds is vendor specific.
Description	Optionally a vendor specific description is provided
NodeClass	Shall reflect the NodeClass of the Node
NodeId	The NodeId is described by BrowseNames as defined in 3.3.4.1 and defined in Annex A.
WriteMask	Optionally the WriteMask Attribute can be provided. If the WriteMask Attribute is provided, it shall set all Attributes to not writeable that are not said to be vendor-specific. For example, the Description Attribute may be set to writeable since a Server may provide a server-specific description for the Node. The NodeId shall not be writeable, because it is defined for each Node in this specification.
UserWriteMask	Optionally the UserWriteMask Attribute can be provided. The same rules as for the WriteMask Attribute apply.

3.3.5.2 Objects

For all Objects specified in this part, the Attributes named in Table 4 shall be set as specified in the table.

Table 4 – Common Object Attributes

Attribute	Value
EventNotifier	Whether the Node can be used to subscribe to Events or not is vendor specific

3.3.5.3 Variables

For all Variables specified in this part, the Attributes named in Table 5 shall be set as specified in the table.

AVA Documents Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03

Table 5 – Common Variable Attributes

Attribute	Value
MinimumSamplingInterval	Optionally, a vendor-specific minimum sampling interval is provided
AccessLevel	The access level for Variables used for type definitions is vendor-specific, for all other Variables defined in this part, the access level shall allow a current read; other settings are vendor specific.
UserAccessLevel	The value for the UserAccessLevel Attribute is vendor-specific. It is assumed that all Variables can be accessed by at least one user.
Value	For Variables used as InstanceDeclarations, the value is vendor-specific; otherwise it shall represent the value described in the text.
ArrayDimensions	If the ValueRank does not identify an array of a specific dimension (i.e. ValueRank <= 0) the ArrayDimensions can either be set to null or the Attribute is missing. This behaviour is vendor-specific. If the ValueRank specifies an array of a specific dimension (i.e. ValueRank > 0) then the ArrayDimensions Attribute shall be specified in the table defining the Variable.

3.3.5.4 VariableTypes

For all VariableTypes specified in this part, the Attributes named in Table 6 shall be set as specified in the table.

Table 6 – Common VariableType Attributes

Attributes	Value
Value	Optionally a vendor-specific default value can be provided
ArrayDimensions	If the ValueRank does not identify an array of a specific dimension (i.e. ValueRank <= 0) the ArrayDimensions can either be set to null or the Attribute is missing. This behaviour is vendor-specific. If the ValueRank specifies an array of a specific dimension (i.e. ValueRank > 0) then the ArrayDimensions Attribute shall be specified in the table defining the VariableType.

3.3.6 Graphical Notation

OPC UA defines a graphical notation for an OPC UA AddressSpace. It defines graphical symbols for all NodeClasses and how different types of References between Nodes can be visualized. Figure shows the symbols for the six NodeClasses used in this specification. NodeClasses representing types always have a shadow.

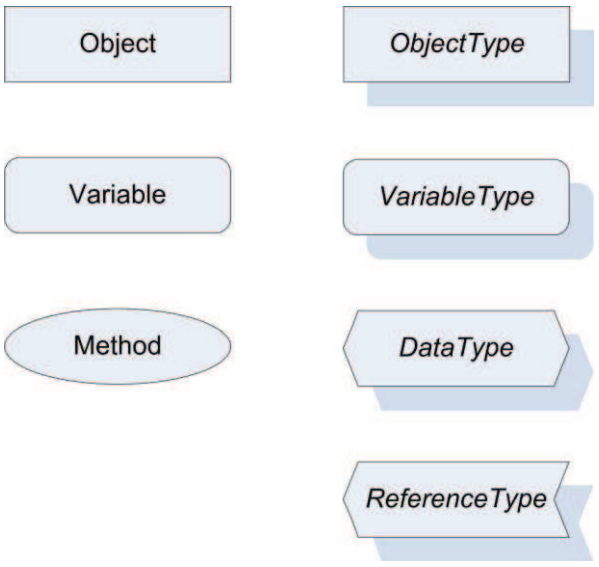


Figure 2 – OPC UA Graphical Notation for NodeClasses

Documents
Document Sharing Hub

Full document details: E DIN EN IEC 62453-71:2022-03