

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Field device integration (FDI) –
Part 100: Profiles – Generic protocols**

**Intégration des appareils de terrain (FDI) –
Partie 100: Profils – Protocoles génériques**

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2020 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Field device integration (FDI) –
Part 100: Profiles – Generic protocols**

**Intégration des appareils de terrain (FDI) –
Partie 100: Profils – Protocoles génériques**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-8858-0

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD	4
1 Scope	6
2 Normative references	6
3 Terms, definitions, abbreviated terms and conventions	6
3.1 Abbreviated terms	7
3.2 Conventions	7
3.2.1 EDDL syntax	7
3.2.2 XML syntax	7
3.2.3 Capitalizations	7
4 Profile for Generic Protocols	8
4.1 General	8
4.2 Catalog profile	8
4.2.1 Protocol support file	8
4.2.2 CommunicationProfile definition	8
4.2.3 Profile device	8
4.2.4 Protocol version information	9
4.3 Associating a Package with a device	9
4.3.1 Device type identification mapping	9
4.3.2 Device type revision mapping	10
4.4 Information Model mapping	10
4.4.1 ProtocolType definition	10
4.4.2 DeviceType mapping	11
4.4.3 FunctionalGroup identification definition	11
4.5 Topology elements	12
4.5.1 ConnectionPoint definition	12
4.5.2 Communication Device definition	13
4.5.3 Communication service provider definition	14
4.5.4 Network definition	14
4.6 Methods	15
4.6.1 Methods for FDI Communication Servers	15
4.6.2 Methods for Gateways	19
4.6.3 Transfer service parameters	26
Annex A (normative) Topology Scan result schema	27
A.1 General	27
A.2 Network	27
A.3 GenericNetworkT	27
A.4 GenericConnectionPointT	27
A.5 GenericIdentificationT	28
A.6 GenericAddressT	29
Annex B (normative) Transfer service parameters	30
B.1 General	30
B.2 sendData	30
B.3 receiveData	30
B.4 TransferSendDataT	30
B.5 EddDataTypeInfoListT	31

B.6	EddDataTypeInfoT	31
B.7	EddDataTypeT	32
B.8	TransferResultDataT	33
Annex C (normative) Protocol-specific definitions		34
C.1	General	34
C.2	Header	34
C.3	ProtocolIdentifier	34
C.4	Address	34
C.5	Manufacturer	34
C.6	DeviceModel	35
C.7	DeviceRevision	35
C.8	SerialNumber	35
C.9	Tag	35
C.10	ProfileId	35
C.11	Version	36
C.12	ProtocolSupportFile	36
Bibliography		37
Table 1 – ProtocolSupportFile for FDI Device Packages		8
Table 2 – Catalog values for profile devices		9
Table 3 – Device identification information mapping		10
Table 4 – Device revision information mapping		10
Table 5 – Protocol type GenericProtocol		11
Table 6 – Inherited DeviceType property mapping		11
Table 7 – Generic Protocol Device Types identification attributes		12
Table 8 – ConnectionPoint type for Generic Protocols		12
Table 9 – Method Connect arguments		15
Table 10 – Method Disconnect arguments		16
Table 11 – Method Transfer arguments		17
Table 12 – EddDataTypeInfo DataType Structure		17
Table 13 – EddDataTypeEnum Values		18
Table 14 – Method SetAddress arguments		19
Table 15 – Connect service arguments		20
Table 16 – Method Transfer arguments		22
Table 17 – Method SetAddress arguments		25
Table A.1 – Elements of GenericNetworkT		27
Table A.2 – Attributes of GenericConnectionPointT		28
Table A.3 – Elements of GenericConnectionPointT		28
Table A.4 – Attributes of GenericIdentificationT		29
Table B.1 – Attributes of TransferSendDataT		31
Table B.2 – Elements of TransferSendDataT		31
Table B.3 – Elements of EddDataTypeInfoListT		31
Table B.4 – Attributes of EddDataTypeInfoT		32
Table B.5 – Enumerations of EddDataTypeT		33
Table B.6 – Attributes of TransferResultDataT		33

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE INTEGRATION (FDI) –**Part 100: Profiles – Generic protocols****FOREWORD**

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

International Standard IEC 62769-100 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

FCG TS62769-100 Edition 1.1, *Field Device Integration Part 100: Profiles – Generic Protocols*, a specification of the FieldComm Group, PROFIBUS Nutzerorganisation e. V., OPC Foundation and FDT Group, serves as a basis for the elaboration of this standard.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
65E/739/FDIS	65E/743/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 document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62769 series, published under the general title *Field Device Integration (FDI)*, can be found on the IEC website.

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

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

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

FIELD DEVICE INTEGRATION (FDI) –

Part 100: Profiles – Generic protocols

1 Scope

This part of IEC 62769 specifies an FDI profile of IEC 62769 for generic protocols. That means that all interfaces are defined, and a host can add support for more protocols without changing its implementation. Nevertheless, there are some protocol-specific definitions (PSD) that need to be specified per protocol using this profile. Annex C specifies what PSDs need to be defined per protocol so that FDI Device Packages, FDI Communication Packages for Gateways and FDI Communication Servers, FDI Communication Servers, Gateways and Devices supporting such a protocol can work together in a host not aware about this specific protocol.

NOTE A host not using an FDI Communication Server but a proprietary mechanism for communication defines its own means to deal with this profile to support several protocols without changing its implementation. This is specific to the proprietary way how the communication driver is bound to the host.

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 61804 (all parts), *Function blocks (FB) for process control and Electronic Device Description Language (EDDL)*

IEC 61804-3, *Function blocks (FB) for process control and Electronic Device Description Language (EDDL) – Part 3: EDDL syntax and semantics*

IEC 62541-100:2015, *OPC Unified Architecture – Part 100: OPC UA for Devices*

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

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

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

IEC 62769-7, *Field Device Integration (FDI) – Part 7: FDI Communication Devices*

3 Terms, definitions, abbreviated terms and conventions

For the purposes of this document, the terms and definitions given in IEC 61804 series, IEC 62541-100, IEC 62769-2, IEC 62769-4, IEC 62769-5 and IEC 62769-7 apply.

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

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

3.1 Abbreviated terms

EDD electronic device description

EDDL Electronic Device Description Language (see IEC 61804)

FDI™ Field Device Integration™¹

FCG FieldComm Group

XML Extensible Markup Language (see FCG TS62769-100, Edition 1.1, Field Device Integration Part 100: Profiles – Generic Protocols, available at <https://fieldcommgroup.org>)

3.2 Conventions

3.2.1 EDDL syntax

This document specifies content for the EDD component that is part of FDI Communication Packages. The specification content using EDDL syntax uses the font Courier New. The EDDL syntax is used for method signature, variable, data structure and component declarations.

3.2.2 XML syntax

XML syntax examples use the font Courier New. The XML syntax is used to describe XML document schema.

Example: `<xs:simpleType name="ExampleType">`

3.2.3 Capitalizations

The IEC 62769 series uses capitalized terms to emphasize that these terms have an FDI-specific meaning.

Some of these terms use an initialism as a prefix, for example:

- FDI Client, or
- FDI Server.

Some of these terms are compound terms, such as:

- Communication Servers, or
- Profile Package.

Parameter names or attributes are concatenated into a single term, where the original terms start in this term with a capital letter, such as:

- ProtocolSupportFile or
- ProtocolType.

Parameter names or attributes can also be constructed by using an underscore character to concatenate two or more terms, such as:

- DEVICE_REV or
- DEVICE_MODEL

¹ Field Device Integration (FDI) is the trademark of a product supplied by FieldComm Group. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

4 Profile for Generic Protocols

4.1 General

This profile document to the FDI specification in IEC 62769 specifies the protocol specifics needed for FDI Packages describing Communication Servers, Gateways and Devices.

For Communication Servers, this document defines protocol specifics as these need to be considered in the Information Model hosted by the Communication Servers.

4.2 Catalog profile

4.2.1 Protocol support file

4.2.1.1 FDI Device Package

Protocol-specific attachments are mentioned in the Package Catalog as defined in IEC 62769-5. As this document defines a profile generically suitable for many protocols, it does not define requirements for any protocol-specific attachments. However, it does also not prevent the usage of protocol-specific attachments. The PSDs (see Annex C) define the requirements on the need of ProtocolSupportFiles for a specific protocol. However, the configuration of a device using an FDI Device Package shall not require the usage of a protocol-specific attachment. Table 1 specifies the parameters of the ProtocolSupportFile in the FDI Device Package in case one or many are provided.

Table 1 – ProtocolSupportFile for FDI Device Packages

Parameter	Description
Content Type	text/plain
Root Namespace	empty
Source Relationship	http://fdi-cooperation.com/2010/relationship/attachment-protocol
Filename	Not defined

4.2.1.2 FDI Communication Packages

The same rules as for FDI Device Packages apply.

4.2.2 CommunicationProfile definition

IEC 62769-4 defines a CommunicationProfileT string for the Catalog XML schema. The string is protocol-specific and defined as ProfileIdentifier in the PSD (see Annex C).

4.2.3 Profile device

A Profile Package shall provide the catalog values for profile devices, enabling the FDI Server to leverage a generic device description, if a specific one is not available. The definitions in Table 2 focus on catalog content that is vendor-independent.

Table 2 – Catalog values for profile devices

Element	Attribute	Content
PackageType	—	Profile
Manufacturer	—	Empty
DeviceModel	—	The format of the DeviceModel is protocol specific and details on the format are defined in the PSD (see Annex C). In order to assign a scan result with a profile package the ProfileId of the scan result shall be mapped to the DeviceModel of the profile package.

4.2.4 Protocol version information

IEC 62769-4 defines an element type named InterfaceT for the Catalog XML schema. The element type InterfaceT contains an element named Version, which is supposed to provide version information about the applied communication protocol profile. The value shall follow the IEC 62769-4 defined version information schema defined in the element type VersionT. The PSDs (see Annex C) define the mapping of versions of a specific protocol to this field.

4.3 Associating a Package with a device

4.3.1 Device type identification mapping

The purpose of device-type identification mapping is to enable FDI host systems to compare the scan result against the topology representation in the Information Model. FDI host systems shall also be enabled to determine the FDI Device Package that fits for a device entry contained in the scan result. This will enable the user of an FDI host system to synchronize the Information Model with the actual installation.

The Communication Server implemented scan service (defined in 4.6.1.7) provides the scan result through an XML document (the schema is defined in Clause A.5).

The Gateway implemented scan service (defined in 4.6.2.7) provides the scan result by means of the Information Model that contains data structures created from EDD content as specified in 4.6.2.7.

For both ways of presenting the scan result, it is common that scan results contain device type identification and device instance identification.

FDI host systems comparing the actual network topology configuration against the topology representation in the Information Model shall be enabled to handle the following situations:

- The physical Device instance identified at a specific device address is not logically present in the Information Model (as Instance): enable the FDI Host system to find the appropriate FDI Device Package according to the device catalog information.
- The physical Device instance identified by the device address is logically present in the Information Model (as Instance): enable the FDI Host system to compare device type information presented in scan result (see the identification in Clause A.5) and the device type specific information of the Instance present in the Information Model.

The FDI Device Package contains device type identification information that can be compared to the scan results based on the Catalog Schema in IEC 62769-4 defining the XML (simple) element types “DeviceModel” and “Manufacturer”. Both types are used in the (complex) element types “Protocol” and “RegDeviceType”.

As a result of the FDI Package deployment, the FDI Package information is then present in the Information Model as the specified FunctionalGroup Identification containing SerialNumber and Tag (see 4.4.3).

The mapping between different device identification data sources is described in Table 3. Since scan results provided by the Communication Server or Gateway can convey data that is produced by the device (firmware), the device type identification mapping shall be supported by providing corresponding data in the Catalog and Information Model contained in the FDI Device Package.

Table 3 – Device identification information mapping

FDI Device Package	Information Model	Communication Server provided scan result	Gateway provided scan result
Catalog specified type Manufacturer	FunctionalGroup: Identification Browse Name: Manufacturer	Element (path): ConnectionPoint/Identification Attribute: Manufacturer	COLLECTION ConnectionPoint. Identification: Manufacturer
Catalog specified type DeviceModel	FunctionalGroup: Identification Browse Name: DeviceModel	Element (path): ConnectionPoint/Identification Attribute: DeviceModel	COLLECTION ConnectionPoint. Identification: DeviceModel

Since not all protocols that are intended to be used with this profile for generic protocols might support a mandatory discovery mechanism allowing to identify the type of device (Manufacturer and DeviceModel), the scan results provide the capability to exclude the identification of the device and only provide the address. In that case, some host-specific mechanisms can be used to assign the desired FDI package to the device, for example by user interaction.

Since some protocols do not even have the mandatory capabilities to identify if there is a device at all for a specific protocol, address hosts should provide the capability for some users to add devices by manually specifying address information.

4.3.2 Device type revision mapping

IEC 62769-4 envisions a concept that allows to determine the compatibility between an FDI Device Package and a Device. IEC 62769-4 specifies a lifecycle management process bearing on a single version information provided for the entire device. This is captured in the DeviceRevision (see Table 4). Mapping of version information is protocol-specific and needs to be defined in the PSDs (see Annex C).

Table 4 – Device revision information mapping

FDI Device Package	Information Model	Communication Server provided scan result	Gateway provided scan result
Catalog specified type ListOfSupportedDeviceRevisions	FunctionalGroup: Identification Browse Name: DeviceRevision	Element (path): ConnectionPoint/Identification Attribute: DeviceRevision	COLLECTION ConnectionPoint. Identification: DeviceRevision

4.4 Information Model mapping

4.4.1 ProtocolType definition

In Table 5, a subtype of ProtocolType is defined to identify network communication using this profile.

Table 5 – Protocol type GenericProtocol

Attribute	Value				
BrowseName	GenericProtocol				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the ProtocolType defined in IEC 62541-100.					
HasProperty	Variable	ProtocolIdentifier	String	PropertyType	Mandatory

The mandatory Variable ProtocolIdentifier defines which concrete protocol is represented using the GenericProtocol type. It shall match the ProtocolIdentifier defined for the CommunicationProfile in 4.2.2. The string is protocol-specific and defined as ProfileIdentifier in the PSDs (see Annex C).

4.4.2 DeviceType mapping

Each device type inherits the properties of DeviceType. The mapping of the inherited properties from DeviceType is defined in Table 6. Note that only the attributes defined in Annex C, and therefore expected by each generic protocol, are used. Specific protocols can provide, for example, a SoftwareRevision, but since this is not accessible for the host, this profile does not make use of it.

Table 6 – Inherited DeviceType property mapping

Property	Generic Protocol Mapping
SerialNumber	SerialNumber (see Annex C)
RevisionCounter	-1 (not defined)
Manufacturer	String taken from FDI package catalog (ManufacturerName from PackageT)
Model	String taken from FDI package catalog (Name of DeviceTypeT, which is a localized name)
DeviceManual	empty text string (not supported) ^a
DeviceRevision	DeviceRevision (see Annex C)
SoftwareRevision	empty string (not defined)
HardwareRevision	empty string (not defined)
^a Device manuals are exposed as attachments of the FDI Device Package.	

4.4.3 FunctionalGroup identification definition

As defined in IEC 62541-100:2015, 5.3, each device representation in the FDI Server hosted Information Model shall contain a protocol-specific FunctionalGroup named Identification. The Parameters of this FunctionalGroup are defined for generic protocol device types as follows:

Table 7 – Generic Protocol Device Types identification attributes

BrowseName	DataType	Mandatory/Optional
Manufacturer	String	Mandatory
DeviceModel	String	Mandatory
SerialNumber	String	Optional
Tag	String	Optional
DeviceRevision	UInt16	Optional
ProfileId	String	Optional

The BaseDataVariable instances shall be created from VARIABLE declarations with identifiers that correspond to the browse names listed in Table 7.

4.5 Topology elements

4.5.1 ConnectionPoint definition

The ConnectionPoint type GenericConnectionPoint shall be used to parameterize network access points using the generic protocols. The ConnectionPoint type GenericConnectionPoint is a sub type of the abstract type ConnectionPointType defined in IEC 62541-100. Table 8 specifies the representation of the GenericConnectionPoint in the AddressSpace.

Table 8 – ConnectionPoint type for Generic Protocols

Attribute	Value				
BrowseName	GenericConnectionPoint				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Sub type of the ConnectionPointType defined in IEC 62541-100.					
HasProperty	Variable	Address	String	PropertyType	Mandatory
HasProperty	Variable	ProtocolIdentifier	String	PropertyType	Mandatory

The ConnectionPoint type GenericConnectionPoint shall be described by an EDD element contained in a Communication Device related FDI Package that can drive a generic protocol network. Actual ConnectionPoint properties are declared by VARIABLE constructs grouped together in a COLLECTION named ConnectionPoint. For this profile, it shall only contain the CONNECTION_POINT_ADDRESS, mapped to the OPC UA Property Address. In addition, the PROTOCOL specified by the COMPONENT shall be mapped to the ProtocolIdentifier Property. The following EDDL source code is an example describing a Connection Point for an ExampleProtocol. The ProtocolIdentifier defined by the PSDs (see Annex C) shall be used as the PROTOCOL name in the EDD.

```

COMPONENT ConnectionPoint_Generic
{
    LABEL "Generic Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL ExampleProtocol;
    CONNECTION_POINT ConnectionPoint;
}

VARIABLE Address
{

```

```

    LABEL "Address";
    HELP "Address of the device";
    TYPE EUC(<protocol-specific>);
    CLASS LOCAL;
}

COLLECTION ConnectionPoint
{
    LABEL "Connection Point";
    MEMBERS
    {
        CONNECTION_POINT_ADDRESS, Address;
    }
}

```

4.5.2 Communication Device definition

In accordance with IEC 62769-7, each FDI Communication Package shall contain an EDD element describing the communication device. The following EDDL source code in is an example describing a Communication Server.

```

COMPONENT Generic_Communication_Server
{
    LABEL "Generic communication server";
    PRODUCT_URI "urn:Company Name:Product Name";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS
    {
        Generic_Communication_Device_Setup
    }
}

COMPONENT_RELATION Generic_Communication_Device_Setup
{
    LABEL "Relation between Device and communication device";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        Generic_Communication_Device{AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 4;
}

```

In accordance with IEC 62769-7, each FDI Communication Package shall contain at least one EDD element describing at least one communication device component. The following EDDL source code in is an example for a generic protocol communication device:

```

COMPONENT Generic_Communication_Device
{
    LABEL "Generic communication device";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { Generic_Service_Provider_Relation }
}

COMPONENT_RELATION Generic_Service_Provider_Relation
{
    LABEL "Relation to communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        Generic_Service_Provider{AUTO_CREATE 1;}
    }
}

```

```

    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

In an actual communication device the ConnectionPoint_Generic needs to be adapted according to the supported protocol and the related connection point definitions given in 4.5. The attribute BYTE_ORDER shall not be used for this profile as the byte order handling shall be done in the gateway business logic.

4.5.3 Communication service provider definition

In accordance with IEC 62769-7, each FDI Communication Package shall contain at least one EDD element describing at least one communication service provider component. The following EDDL source code below is an example for a generic protocol communication service provider component.

The component reference (ConnectionPoint_Generic) corresponds to the related connection point definition in 4.5. The attribute BYTE_ORDER shall not be used for this profile as the byte order handling shall be done in the gateway business logic.

```

COMPONENT Generic_Service_Provider
{
    LABEL "Generic Protocol communication service provider";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
    COMPONENT_RELATIONS
    {
        Generic_Service_Provider_Connection_Point_Relation
    }
}

COMPONENT_RELATION Generic_Service_Provider_Connection_Point_Relation
{
    LABEL "Relation between communication service provider and Connection Point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {Address}
    COMPONENTS
    {
        ConnectionPoint_Generic{ AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

4.5.4 Network definition

In accordance with to IEC 62769-7, each FDI Communication Package shall contain at least one EDD element describing network configuration constraints using the component construct.

```

COMPONENT Network_Generic
{
    LABEL "Generic Network";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK;
    COMPONENT_RELATIONS
    {
        Generic_Network_Connection_Point_Relation
    }
}

COMPONENT_RELATION Generic_Network_Connection_Point_Relation
{
    LABEL "Relation between network and Connection Point";
    RELATION_TYPE CHILD_COMPONENT;
}

```

```

ADDRESSING {Address}
COMPONENTS
{
    ConnectionPoint_Generic
}
MINIMUM_NUMBER 1;
MAXIMUM_NUMBER 32;
}

```

4.6 Methods

4.6.1 Methods for FDI Communication Servers

4.6.1.1 General

The Communication Server contained Information Model shall implement services according to method signatures described in 4.6.1.

4.6.1.2 Connect

Signature:

```

Connect (
    [in]   ByteString   CommunicationRelationId,
    [in]   String       Address,
    [out]  Int32         ServiceError);

```

Table 9 provides the description of the arguments.

Table 9 – Method Connect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network that is directly connected to the Communication Server hardware. The nodeId allows finding the direct parent-child relation.
Address	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 4.5. The argument value holds the protocol-specific device's network address that is unique within the network segment.
ServiceError	0: OK / execution finished, connection established successfully -1: Connect Failed / canceled by caller -3: Connect Failed / device not found -4: Connect Failed / invalid device address -5: Connect Failed / invalid device identification

4.6.1.3 Disconnect

Signature:

```

Disconnect (
    [in]   ByteString   CommunicationRelationId,
    [out]  Int32         ServiceError);

```

Table 10 provides the description of the arguments.

Table 10 – Method Disconnect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the Communication Server hardware. The nodeId allows finding the direct parent-child relation.
ServiceError	0: OK / disconnect finished successfully -1: Disconnect Failed / no existing communication relation -2: Disconnect Failed / invalid communication relation identifier

4.6.1.4 Transfer

Signature

```

Transfer (
    [in]   ByteString      CommunicationRelationId,
    [in]   String          Header,
    [in]   ByteString      RequestData,
    [in]   EddDataTypeInfo[] RequestDataTypes,
    [in]   EddDataTypeInfo[] ResponseDataTypes,
    [out]  ByteString      ResponseData,
    [out]  ByteString      RESPONSE_CODES,
    [out]  Int32           ServiceError);

```

Table 11 provides the description of the arguments.

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

Table 11 – Method Transfer arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network within the Information Model.
Header	The protocol-specific information on the data to be transferred. The Header is used in the COMMANDs of the EDD, and the PSDs (see Annex C) define the format of this string per protocol. The first communication service treating this string shall validate the string according to its format defined in the PSDs.
RequestData	The argument name shall match with the corresponding COMMAND sub-element name REQUEST. The byte stream submitted through the argument is created from definitions provided by the REQUEST element of the corresponding COMMAND that shall be processed.
RequestDataTypes	This array contains information on what data types are used to create the RequestData ByteString. The byte order used in the RequestData is EDD-specific and providing the information allows the gateway or communication server to change it to a protocol-specific byte order of the data type or applying any other protocol-specific characteristics (e.g. bit order). The overall length of all provided data types in the RequestDataTypes shall match the length of the RequestData.
ResponseDataTypes	This array contains information on what data types are expected when receiving the ResponseData ByteString. The knowledge about the expected data types is in the EDD of the device and therefore the gateway or communication server needs to get this information in order to apply protocol-specific characteristics to the response data and knows how to separate the response data into an array of ByteString. The overall length of all provided data types in the ResponseDataTypes shall match the length of the ResponseData.
ResponseData	The argument name shall match with the corresponding COMMAND sub-element name REPLY. The byte stream returned by this argument applies to definitions provided by the REPLY element of the corresponding COMMAND that shall be processed.
RESPONSE_CODES	The argument name shall match with the COMMAND sub-element name RESPONSE_CODES. The argument value conveys the specific communication service response bytes.
ServiceError	0: OK / execution finished -1: Transfer Failed / canceled by caller -3: Transfer Failed / no existing communication relation. -4: Transfer Failed / invalid communication relation identifier -5: Transfer Failed / invalid sendData content -6: Transfer Failed / invalid receiveData format

The EddDataTypeInfo DataType defines the data type information of an EDD data type used in a COMMAND. Its elements are defined in Table 12.

Table 12 – EddDataTypeInfo DataType Structure

Name	Type	Description
EddDataTypeInfo	structure	This structure specifies information about a data type used in an EDD COMMAND.
eddDataType	EddDataTypeEnum	The EddDataType used.
size	UInt32	The size of the eddDataType. It shall always be provided, even if not explicitly specified by the EDD. In that case, the default value for that data type shall be provided. For data types where no size can be defined (e.g. BOOLEAN) the value shall be set to 0.

The `EddDataTypeEnum` `DataType` is an enumeration that defines the possible EDD data types. Its values are defined in Table 13.

Table 13 – EddDataTypeEnum Values

Value	Description
BOOLEAN	Data type as defined by IEC 61804-3
DOUBLE	Data type as defined by IEC 61804-3
FLOAT	Data type as defined by IEC 61804-3
INTEGER	Data type as defined by IEC 61804-3
UNSIGNED_INTEGER	Data type as defined by IEC 61804-3
DATE	Data type as defined by IEC 61804-3
DATE_AND_TIME	Data type as defined by IEC 61804-3
DURATION	Data type as defined by IEC 61804-3
TIME	Data type as defined by IEC 61804-3
TIME_VALUE	Data type as defined by IEC 61804-3
BIT_ENUMERATED	Data type as defined by IEC 61804-3
ENUMERATED	Data type as defined by IEC 61804-3
ASCII	Data type as defined by IEC 61804-3
BITSTRING	Data type as defined by IEC 61804-3
EUC	Data type as defined by IEC 61804-3
OCTET	Data type as defined by IEC 61804-3
PACKED_ASCII	Data type as defined by IEC 61804-3
PASSWORD	Data type as defined by IEC 61804-3
VISIBLE	Data type as defined by IEC 61804-3

4.6.1.5 GetPublishedData

This method is not supported by the generic protocols profile.

4.6.1.6 SetAddress

Signature

```
SetAddress(
    [in] String      OldAddress,
    [in] String      NewAddress,
    [out] Int32       ServiceError);
```

Table 14 provides the description of the arguments.

Table 14 – Method SetAddress arguments

Argument	Description
OldAddress	The argument value holds the current address of a device. The same semantic applies to this field as the Address parameter in the Connect method.
NewAddress	The argument value holds the new address for a device. The same semantic applies to this field as the Address parameter in the Connect method.
ServiceError	0: OK / execution finished successfully -1: SetAddress Failed / canceled by caller -3: SetAddress Failed / not initialized -4: SetAddress Failed / not connected to a network -5: SetAddress Failed / no device found responding to oldAddress -6: SetAddress Failed / duplicate address error -7: SetAddress Failed / device did not accept new address -8: SetAddress Failed / invalid oldAddress (in terms of syntax, data type, data format, and so on) -9: SetAddress Failed / invalid newAddress (in terms of syntax, data type, data format, and so on) -10: SetAddress Failed / not possible in status connected

4.6.1.7 Scan

The method signature specified in IEC 62769-7 applies. The corresponding topologyScanResult schema is specified in Clause A.2.

4.6.1.8 ResetScan

The method signature specified in IEC 62769-7 applies.

4.6.2 Methods for Gateways

4.6.2.1 General

The methods signatures defined in 4.6.2 apply. The methods shall be implemented in the EDD element (IEC 62769-4) contained in a Gateway-related FDI Package containing the communication device definitions.

4.6.2.2 Connect

This subclause describes the generic protocol-specific implementation of the service Connect specified in IEC 62769-7.

```

METHOD BeginConnect(
    DD_STRING          CommunicationRelationId,
    DD_STRING          Address,
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

```

METHOD EndConnect (
    DD_STRING          CommunicationRelationId,
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

METHOD CancelConnect (
    DD_STRING          CommunicationRelationId,
    unsigned long      ServiceId,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

Table 15 provides the description of the arguments.

Table 15 – Connect service arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeld of the ConnectionPoint representing the connection between a device and a physical network, which is directly connected to the Communication Server hardware. The nodeld allows finding the direct parent-child relation.
Address	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 4.5. The argument value holds the protocol-specific device's network address that is unique within the network segment.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndConnect invocation cycle that shall not be shorter than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndConnect 0: OK / execution finished, connection established successfully -1: Connect Failed / canceled by caller -2: Call Failed / unknown service ID -3: Connect Failed / device not found -4: Connect Failed / invalid device address -5: Connect Failed / invalid device identification

4.6.2.3 Disconnect

This subclause describes the generic protocol-specific implementation of the service Disconnect specified in IEC 62769-7.

```

METHOD Disconnect(
    DD_STRING CommunicationRelationId,
    long      &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

All the arguments of the Disconnect method are described in Table 10.

4.6.2.4 Transfer

This subclause describes the generic protocol specific implementation of the service Transfer specified in IEC 62769-7. Owing to limitations in the capabilities of EDD methods (it is not possible to transfer any collections of a dynamic size), this profile requires some additional handling when using the transfer via Gateways. In addition to the calls of the EDD method, a host first needs to provide the data type information of the request and the response data. The following order needs to be executed by the host:

- 1) Add the request and response data type information into a list defined by the gateway EDD (called DATA_TYPE_INFO_PER_SERVICE_ID). The entry shall include the ServiceId also used in the method calls.
- 2) Call BeginTransfer using the same ServiceId and follow the asynchronous call pattern as defined in IEC 62769-7.
- 3) When the host has received the response, it is the responsibility of the host to remove the entry from the list. Thus, only the host is manipulating the list and the gateway business logic is only accessing, but not changing, the list in order to get the information on the data types.

```

METHOD BeginTransfer(
    DD_STRING      CommunicationRelationId,
    DD_STRING      Header,
    DD_STRING      RequestData,
    unsigned long   ServiceId,
    DD_STRING      &ResponseData,
    DD_STRING      &RESPONSE_CODES,
    long           &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

```

METHOD EndTransfer(
    DD_STRING      CommunicationRelationId,
    unsigned long   ServiceId,
    DD_STRING      &ResponseData,
    DD_STRING      &RESPONSE_CODES,
    unsigned long   &DelayForNextCall,
    long           &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

```

METHOD CancelTransfer (
    DD_STRING          CommunicationRelationId,
    DD_STRING          &ResponseData,
    DD_STRING          &RESPONSE_CODES,
    unsigned long      ServiceId,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

Table 16 provides the description of the arguments.

Table 16 – Method Transfer arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeld of the ConnectionPoint representing the connection between a device and a physical network within the Information Model.
Header	The protocol-specific information on the data to be transferred. The Header is used in the COMMANDs of the EDD and the PSDs (see Annex C) define the format of this string per protocol. The first communication service treating this string shall validate the string according to its format defined in the PSDs.
RequestData	The argument name shall match with the corresponding COMMAND sub-element name REQUEST. The byte stream submitted through the argument is created from definitions provided by the REQUEST element of the corresponding COMMAND that shall be processed. The byte stream is separated in several byte streams for the different requested data types. This allows a protocol-specific byte order of specific data types in the gateway or communication server to be considered.
ResponseData	The argument name shall match with the corresponding COMMAND sub-element name REPLY. The byte stream returned by this argument applies to definitions provided by the REPLY element of the corresponding COMMAND that shall be processed. The data are separated into several byte strings based on the ResponseDataTypes field. This allows the gateway or communication server to consider protocol-specific byte order of data types.
RESPONSE_CODES	The argument name shall match with the COMMAND sub-element name RESPONSE_CODES. The argument value conveys the specific communication service response bytes.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndTransfer invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndTransfer 0: OK / execution finished -1: Transfer Failed / canceled by caller -2: Call Failed / unknown service ID -3: Transfer Failed / no existing communication relation -4: Transfer Failed / invalid communication relation identifier -5: Transfer Failed / invalid sendData content -6: Transfer Failed / invalid receiveData format

The List that is used to add the data type information so the business logic of the gateway can do the needed data type transfers shall be called DATA_TYPE_INFO_LIST. It is defined in the following:

VARIABLE EddDataTypeEnum

```
{
  LABEL "Edd Data Type";
  CLASS LOCAL;
  TYPE ENUMERATED
  {
    { 0x00, "BOOLEAN" }
    { 0x01, "DOUBLE" }
    { 0x02, "FLOAT" }
    { 0x03, "INTEGER" }
    { 0x04, "UNSIGNED_INTEGER" }
    { 0x05, "DATE" }
    { 0x06, "DATE_AND_TIME" }
    { 0x07, "DURATION" }
    { 0x08, "TIME" }
    { 0x09, "TIME_VALUE" }
    { 0x0a, "BIT_ENUMERATED" }
    { 0x0b, "ENUMERATED" }
    { 0x0c, "ASCII" }
    { 0x0d, "BITSTRING" }
    { 0x0e, "EUC" }
    { 0x10, "OCTET" }
    { 0x11, "PACKED_ASCII" }
    { 0x12, "PASSWORD" }
    { 0x13, "VISIBLE" }
  }
}
```

VARIABLE SizeV

```
{
  LABEL "Size of data type";
  TYPE UNSIGNED_INTEGER(4);
  CLASS LOCAL;
}
```

COLLECTION DataTypeInfo

```
{
  MEMBERS
  {
    Size, SizeV;
    EddDataType, EddDataTypeEnum;
  }
}
```

LIST DataTypeInfoList

```
{
  TYPE DataTypeInfo;
}
```

VARIABLE ServiceIdentification

```
{
  LABEL "Identification of service call";
  TYPE UNSIGNED_INTEGER(4);
  CLASS LOCAL;
}
```

COLLECTION DataTypeInfoPerServiceId

```
{
  MEMBERS
  {
    ServiceId, ServiceIdentification;
    RequestDataTypes, DataTypeInfoList;
    ResponseDataTypes, DataTypeInfoList;
  }
}
```

```

    }
}
LIST DATA_TYPE_INFO_PER_SERVICE_ID
{
    TYPE DataTypeInfoPerServiceId;
}

```

4.6.2.5 GetPublishedData

This method is not supported in the generic protocol profile.

4.6.2.6 SetAddress

This subclause describes the generic protocol specific implementation of the service SetAddress specified in IEC 62769-7.

```

BeginSetAddress (
    DD_STRING          OldAddress,
    DD_STRING          NewAddress,
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

EndSetAddress (
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

Table 17 provides the description of the arguments.

Table 17 – Method SetAddress arguments

Argument	Description
OldAddress	The argument value holds the current address of a device. The same semantic applies to this field as the Address parameter in the Connect method.
NewAddress	The argument value holds the new address for a device. The same semantic applies to this field as the Address parameter in the Connect method.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndTransfer invocation cycle that shall not be shorter than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndSetAddress 0: OK / execution finished successfully -1: SetAddress Failed / canceled by caller -2: Call Failed / unknown service ID -3: SetAddress Failed / not initialized -4: SetAddress Failed / not connected to a network -5: SetAddress Failed / no device found responding to oldAddress -6: SetAddress Failed / duplicate address error -7: SetAddress Failed / device did not accept new address -8: SetAddress Failed / invalid oldAddress (in terms of syntax, data type, data format, and so on) -9: SetAddress Failed / invalid newAddress (in terms of syntax, data type, data format, and so on) -10: SetAddress Failed / not possible in status connected

4.6.2.7 Scan

The method signature specified in IEC 62769-7 applies. The generic protocol gateway business logic shall create the scan result in accordance with IEC 62769-7. The following definitions shall be present in the COMPONENT declaration that holds the definitions for a communication device. The data structure corresponds to the data structure defined in the XML schema in Annex A. The SCAN_LIST attribute inside the COMPONENT declaration shall refer to LIST TopologyScanResult.

```
VARIABLE GenericAddress
{
  LABEL "Station address";
  TYPE EUC(<protocol-specific>);
  CLASS LOCAL;
}
```

```
VARIABLE Manufacturer
{
  LABEL "Manufacturer ID";
  TYPE EUC(<protocol-specific>);
  CLASS LOCAL;
}
```

```
VARIABLE DeviceModel
{
  LABEL "Device ID";
  TYPE EUC(<protocol-specific>);
  CLASS LOCAL;
}
```

```

VARIABLE DeviceRevision
{
    LABEL "Device revision";
    TYPE UNSIGNED_INTEGER(2);
    CLASS LOCAL;
}

VARIABLE SerialNumber
{
    LABEL "Serial number";
    TYPE EUC(<protocol-specific>);
    CLASS LOCAL;
}

VARIABLE Tag
{
    LABEL "Tag";
    TYPE EUC(<protocol-specific>);
    CLASS LOCAL;
}

VARIABLE ProfileId
{
    LABEL "ProfileId";
    TYPE EUC(<protocol-specific>);
    CLASS LOCAL;
}

COLLECTION ScanItemType
{
    MEMBERS
    {
        ADDRESS, GenericAddress;
        MANUFACTURER_ID, Manufacturer;
        DEVICE_MODEL, DeviceModel;
        DEVICE_REV, DeviceRevision;
        SERIAL_NUMBER, SerialNumber;
        TAG, Tag;
        PROFILEID, ProfileId;
    }
}

LIST TopologyScanResult
{
    TYPE ScanItemType;
}

```

4.6.2.8 ScanNext

The method signature specified in IEC 62769-7 applies. The generic gateway business logic shall create the scan result in accordance with IEC 62769-7. The method ScanNext stores the result into data structures described for the method Scan (4.6.2.7).

4.6.3 Transfer service parameters

Transfer service parameters are defined in Annex B.

Annex A (normative)

Topology Scan result schema

A.1 General

The topology scan result schema specified in Annex A describes the generic protocols specific format Method Scan argument `topologyScanResult`. The XML document content and structure shall correspond to the Information Model-designed concept to describe a topology in order to enable generic matching between physical devices connected to the network and the Information Model hosted by the FDI Server.

A.2 Network

The subsequent element is used to return the scan result corresponding to the Information Model described in IEC 62769-5.

The XML schema for a Network element is:

```
<xs:element name="Network" type="GP:GenericNetworkT"/>
```

A.3 GenericNetworkT

The element type describes the complete scan result for a single network because of the scan method that is provided per instance of a "Communication Device", which exists in a 1:1 relation to a network instance.

The XML schema for a GenericNetworkT type is:

```
<xs:complexType name="GenericNetworkT">
  <xs:sequence>
    <xs:element name="ConnectionPoint" type="GP:GenericConnectionPointT"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The elements of a GenericNetworkT type are described in Table A.1

Table A.1 – Elements of GenericNetworkT

Element	Description
ConnectionPoint	The ConnectionPoint element holds the address and identification of the network-connected device that has been found during bus scan operations.

A.4 GenericConnectionPointT

The XML schema for a GenericConnectionPointT type is as follows:

```
<xs:complexType name="GenericConnectionPointT">
  <xs:sequence>
    <xs:element name="Identification" type="GP:GenericIdentificationT"
      minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Address" type="GP:GenericAddressT" use="required"/>
</xs:complexType>
```

The attributes of a GenericConnectionPointT type are described in Table A.2.

Table A.2 – Attributes of GenericConnectionPointT

Attribute	Description
Address	The Attribute value holds the address of the network-connected device used to uniquely address the device in the network.

The elements of a GenericConnectionPointT type are described in Table A.3.

Table A.3 – Elements of GenericConnectionPointT

Element	Description
Identification	The element data holds the device type identification data. Compared to the Information Model (IEC 62541-100), the ConnectionPoint does not contain or refer to the device type identification data. But in order to support the FDI host system in finding the package that matches the connected device, this schema associates the device type identification with the ConnectionPoint. Note that this element is optional since it is possible that a protocol does not support identification of a device.

A.5 GenericIdentificationT

The element content corresponds to the "FunctionalGroup Identification".

The XML schema for a GenericIdentificationT type is:

```
<xs:complexType name="GenericIdentificationT">
  <xs:attribute name="Manufacturer" type="xs:string" use="required"/>
  <xs:attribute name="DeviceModel" type="xs:string" use="required"/>
  <xs:attribute name="SerialNumber" type="xs:string" use="optional"/>
  <xs:attribute name="Tag" type="xs:string" use="optional"/>
  <xs:attribute name="DeviceRevision" type="xs:unsignedShort"
    use="optional"/>
  <xs:attribute name="ProfileId" type="xs:string" use="optional"/>
</xs:complexType>
```

The attributes of a GenericIdentificationT type are described in Table A.4.

Table A.4 – Attributes of GenericIdentificationT

Attribute	Description
Manufacturer	See Table 7
DeviceModel	See Table 7
SerialNumber	See Table 7
Tag	See Table 7
DeviceRevision	See Table 7
ProfileId	See Table 7

A.6 GenericAddressT

The protocol-specific address used to uniquely identify a device in a network segment.

The XML schema for a GenericAddressT type is:

```
<xs:simpleType name="GenericAddressT">  
  <xs:restriction base="xs:string"/>  
</xs:simpleType>
```

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

Annex B (normative)

Transfer service parameters

B.1 General

Direct Access Services specified in IEC 62769-2 enable the User Interface Plug-in (UIP) to directly exchange data with the device. Direct data exchange means that data exchanged between a device and a UIP may not be reflected in the Information Model. The IEC 62769-5 defined interface IDirectAccess corresponds to the IEC 62769-2 specified Direct Access Services. Interface IDirectAccess defined functions BeginTransfer and EndTransfer need to convey protocol specific information. The protocol specifics shall be captured in an XML document.

B.2 sendData

The element described in the following contains data to be submitted through the IDirectAccess function BeginTransfer-defined argument sendData.

The XML schema for a sendData element is:

```
<xs:element name="sendData" type="GP:TransferSendDataT"/>
```

B.3 receiveData

The element contains data that is returned through the return value defined in EndTransfer in the IDirectAccess function.

The XML schema for a receiveData element is:

```
<xs:element name="receiveData" type="GP:TransferResultDataT"/>
```

B.4 TransferSendDataT

A complex type that defines the service parameter data format that shall be applied to the Transfer-defined argument sendData.

The XML schema for a TransferSendDataT type is:

```
<xs:complexType name="TransferSendDataT">
  <xs:sequence>
    <xs:element name="RequestDataTypes" type="GP:EddDataTypeInfoListT"/>
    <xs:element name="ResponseDataTypes" type="GP:EddDataTypeInfoListT"/>
  </xs:sequence>
  <xs:attribute name="Header" type="xs:string" use="required"/>
  <xs:attribute name="RequestData" type="xs:hexBinary" use="required"/>
</xs:complexType>
```

The attributes of a TransferSendDataT type are described in Table B.1.

Table B.1 – Attributes of TransferSendDataT

Attribute	Description
Header	The attribute corresponds to the Transfer method argument Header.
RequestData	The attribute corresponds to the Transfer method argument RequestData.

The elements of a TransferSendDataT type are described in Table B.2.

Table B.2 – Elements of TransferSendDataT

Element	Description
RequestDataTypes	The element corresponds to the Transfer method argument RequestDataTypes
ResponseDataTypes	The element corresponds to the Transfer method argument ResponseDataTypes

B.5 EddDataTypeInfoListT

List of EDD data type information.

The XML schema for a EddDataTypeInfoListT type is as follows:

```
<xs:complexType name="EddDataTypeInfoListT">
  <xs:sequence>
    <xs:element name="EddDataTypeInfo" type="GP:EddDataTypeInfoT"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The elements of a EddDataTypeInfoListT type are described in Table B.3.

Table B.3 – Elements of EddDataTypeInfoListT

Element	Description
EddDataTypeInfo	Describes an individual EDD data type and its size.

B.6 EddDataTypeInfoT

Describes an individual EDD data type and its size.

The XML schema for a EddDataTypeInfoT type is:

```
<xs:complexType name="EddDataTypeInfoT">
  <xs:attribute name="EddDataType" type="GP:EddDataTypeT" use="required"/>
  <xs:attribute name="Size" type="xs:nonNegativeInteger" use="required"/>
</xs:complexType>
```

The attributes of a EddDataTypeInfoT type are described in Table B.4.

Table B.4 – Attributes of EddDataTypeInfoT

Attribute	Description
EddDataType	The EddDataType used.
Size	The size of the eddDataType. It shall always be provided, even if not explicitly specified by the EDD. In that case, the default value for that data type shall be provided. For data types where no size can be defined (e.g. BOOLEAN), the value shall be set to 0.

B.7 EddDataTypeT

A simple type that defines the possible EDD data types.

The XML schema for a EddDataTypeT enumeration type is:

```

<xs:simpleType name="EddDataTypeT">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BOOLEAN"/>
    <xs:enumeration value="DOUBLE"/>
    <xs:enumeration value="FLOAT"/>
    <xs:enumeration value="INTEGER"/>
    <xs:enumeration value="UNSIGNED_INTEGER"/>
    <xs:enumeration value="DATE"/>
    <xs:enumeration value="DATE_AND_TIME"/>
    <xs:enumeration value="DURATION"/>
    <xs:enumeration value="TIME"/>
    <xs:enumeration value="TIME_VALUE"/>
    <xs:enumeration value="BIT_ENUMERATED"/>
    <xs:enumeration value="ENUMERATED"/>
    <xs:enumeration value="ASCII"/>
    <xs:enumeration value="BITSTRING"/>
    <xs:enumeration value="EUC"/>
    <xs:enumeration value="OCTET"/>
    <xs:enumeration value="PACKED_ASCII"/>
    <xs:enumeration value="PASSWORD"/>
    <xs:enumeration value="VISIBLE"/>
  </xs:restriction>
</xs:simpleType>

```

The enumeration values of an EddDataTypeT enumeration type are described in Table B.5.

Table B.5 – Enumerations of EddDataTypeT

Enumeration	Description
BOOLEAN	Data type as defined in IEC 61804-3.
DOUBLE	Data type as defined in IEC 61804-3.
FLOAT	Data type as defined in IEC 61804-3.
INTEGER	Data type as defined in IEC 61804-3.
UNSIGNED_INTEGER	Data type as defined in IEC 61804-3.
DATE	Data type as defined in IEC 61804-3.
DATE_AND_TIME	Data type as defined in IEC 61804-3.
DURATION	Data type as defined in IEC 61804-3.
TIME	Data type as defined in IEC 61804-3.
TIME_VALUE	Data type as defined in IEC 61804-3.
BIT_ENUMERATED	Data type as defined in IEC 61804-3.
ENUMERATED	Data type as defined in IEC 61804-3.
ASCII	Data type as defined in IEC 61804-3.
BITSTRING	Data type as defined in IEC 61804-3.
EUC	Data type as defined in IEC 61804-3.
OCTET	Data type as defined in IEC 61804-3.
PACKED_ASCII	Data type as defined in IEC 61804-3.
PASSWORD	Data type as defined in IEC 61804-3.
VISIBLE	Data type as defined in IEC 61804-3.

B.8 TransferResultDataT

A complex type that defines the service parameter data format that shall be applied to the Transfer-defined receivedData return value.

```
<xs:complexType name="TransferResultDataT">
  <xs:attribute name="ResponseData" type="xs:hexBinary" use="required"/>
  <xs:attribute name="RESPONSE_CODES" type="xs:hexBinary" use="required"/>
</xs:complexType>
```

The attributes of a TransferResultDataT type are described in Table B.6.

Table B.6 – Attributes of TransferResultDataT

Attribute	Description
ResponseData	The attribute corresponds to the Transfer method argument ResponseData.
RESPONSE_CODES	The attribute corresponds to the Transfer method argument RESPONSE_CODES.

Annex C (normative)

Protocol-specific definitions

C.1 General

This annex defines what the PSDs need to define per protocol.

C.2 Header

The HEADER attribute is used in an EDD to define the addressing of a COMMAND. When using this annex, each EDD needs to comply with the profile for generic protocol extensions defined in IEC 61804-3 and define COMMANDs in the EDD using the HEADER as addressing in order to access the corresponding VARIABLES in the device. The HEADER consists of a string. Per protocol, the way in which the protocol-specific addressing is mapped to the HEADER attribute needs to be defined. Examples of protocol-specific addressing is the COMMAND number in HART or SLOT and INDEX in Profibus.

The first communication service treating this string shall validate the string according to its format defined in the PSD.

C.3 ProtocolIdentifier

ProtocolIdentifier is used in the Package Catalog defined in IEC 62769-4 (called CommunicationProfile) as well as in the EDD for Gateways defined in IEC 61804-3 (called PROTOCOL) to identify the protocol. It is represented as a string in the EDD and in the Package Catalog. In order to have a unique identifier per protocol, ProtocolIdentifier needs to be defined once per protocol and a URI syntax should be used in order to guarantee the uniqueness of the string.

C.4 Address

Address uniquely identifies a device in a network segment and is provided in the scan result, as well as in the Connect and SetAddress methods of an FDI Communication Server or a Gateway. This profile uses an OPC UA String, XML string or an EDD EUC or DD_STRING in EDD Methods as data structure in order to allow high flexibility on the actual used protocol-specific address as well as an easy way to edit an address by the user. For each protocol, the size of the EUC needs to be defined, as well as details on how to structure the data in the data type and what corresponding protocol-specific addressing is used and whether the SetAddress method can be supported. If SetAddress is not supported, the Gateways and Communication Server shall not provide this method at all.

C.5 Manufacturer

Manufacturer identifies the manufacturer of a device and is provided in the scan result, in the OPC UA information model and in the Package Catalog. This profile uses an XML string, an EDD EUC, or an OPC UA String data type as data structure. For each protocol, the mapping to a protocol-specific representation needs to be defined as well as how to get the information when scanning a network. Manufacturer is used to identify corresponding FDI Device Packages, which also contain the information relating to the manufacturer.

An EDD also contains manufacturer information, in MANUFACTURER. Unless defined differently in the PSD, MANUFACTURER shall have all bits in the integer value set to one, and the MANUFACTURER_EXT shall contain the manufacturer string as defined in the PSDs.

C.6 DeviceModel

DeviceModel identifies the type of device and is provided in the scan result, in the OPC UA information model and in the Package Catalog. This profile uses an XML string, an EDD EUC, or an OPC UA String data type as data structure. For each protocol, the mapping to a protocol-specific representation needs to be defined as well as how to get the information when scanning a network. The DeviceModel is used to identify corresponding FDI Device Packages that also contain the DeviceModel information.

An EDD also contains information about the type of device in DEVICE_TYPE. Unless defined differently in the PSD, the DEVICE_TYPE shall have all bits in the integer value set to one.

C.7 DeviceRevision

DeviceRevision allows to determine the compatibility between an FDI Device Package and a Device and is optionally provided in the scan result and in the OPC UA information model. This profile uses an XML unsignedShort, an EDD UNSIGNED_INTEGER (2), or an OPC UA UInt16 data type as data structure. For each protocol, it needs to be specified whether DeviceRevision is supported and, if it is supported, the mapping to a protocol-specific representation needs to be defined as well as how to get the information when scanning a network.

C.8 SerialNumber

SerialNumber uniquely identifies a device. It is optionally provided in the scan result and in the OPC UA information model. This profile uses an XML String, an EDD EUC or an OPC UA String data type as the data structure. For each protocol, it needs to be specified whether SerialNumber is supported and, if it is supported, the mapping to a protocol-specific representation needs to be defined as well as how to get the information when scanning a network and the size of the EUC. The SerialNumber can be used by Host applications to identify a device, even if it has been moved in the network or is accessible via different network paths.

C.9 Tag

Tag provides a unique name for a device and is optionally provided in the scan result and in the OPC UA information model. This profile uses an XML String, an EDD EUC, or an OPC UA String data type as data structure. For each protocol, it needs to be specified whether Tag is supported and, if it is supported, the mapping to a protocol-specific representation needs to be defined as well as how to get the information when scanning a network and the size of the EUC.

C.10 ProfileId

ProfileId identifies the supported profile of a device. It is optionally provided in the scan result and in the OPC UA information model. This profile uses an XML String, an EDD EUC, or an OPC UA String data type as the data structure. For each protocol it needs to be specified whether ProfileId is supported and, if it is supported, the mapping to a protocol-specific representation needs to be defined, as well as how to get the information when scanning a network and the size of the EUC.

C.11 Version

The protocol version is used in the FDI Package Catalog to define which version of a protocol is supported by the FDI Package. It contains a major number, minor number, and revision number. The mapping of protocol versions to this schema needs to be defined per protocol.

C.12 ProtocolSupportFile

ProtocolSupportFiles can be provided by an FDI Package. Whether such files are required needs to be defined per protocol, including what type of files need to be supported.

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

Bibliography

FCG TS62769-100, Edition 1.1, *Field Device Integration Part 100: Profiles – Generic Protocols*, available at <https://fieldcommgroup.org>

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

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

SOMMAIRE

AVANT-PROPOS	40
1 Domaine d'application	42
2 Références normatives	42
3 Termes, définitions, termes abrégés et conventions	42
3.1 Termes abrégés	43
3.2 Conventions	43
3.2.1 Syntaxe EDDL	43
3.2.2 Syntaxe XML	43
3.2.3 Capitalisations	43
4 Profils des protocoles génériques	44
4.1 Généralités	44
4.2 Profil du catalogue	44
4.2.1 Fichier de prise en charge du protocole	44
4.2.2 Définition de CommunicationProfile	44
4.2.3 Appareil de profil	44
4.2.4 Informations relatives à la version du protocole	45
4.3 Association d'un Paquetage à un appareil	45
4.3.1 Mapping d'identification du type d'appareil	45
4.3.2 Mapping de révisions de type d'appareil	46
4.4 Mapping du Modèle d'information	47
4.4.1 Définition de ProtocolType	47
4.4.2 Mapping de DeviceType	47
4.4.3 Définition du FunctionalGroup "Identification"	47
4.5 Éléments topologiques	48
4.5.1 Définition de ConnectionPoint	48
4.5.2 Définition de l'appareil de communication	49
4.5.3 Définition du fournisseur de services de communication	50
4.5.4 Définition du réseau	50
4.6 Méthodes	51
4.6.1 Méthodes pour les Serveurs de communication FDI	51
4.6.2 Méthodes pour les Passerelles	55
4.6.3 Paramètres du service de transfert	62
Annexe A (normative) Schéma du résultat d'analyse topologique	63
A.1 Généralités	63
A.2 Network	63
A.3 GenericNetworkT	63
A.4 GenericConnectionPointT	63
A.5 GenericIdentificationT	64
A.6 GenericAddressT	65
Annexe B (normative) Paramètres du service de transfert	66
B.1 Généralités	66
B.2 sendData	66
B.3 receiveData	66
B.4 TransferSendDataT	66
B.5 EddDataTypeInfoListT	67
B.6 EddDataTypeInfoT	67

B.7	EddDataTypeT	68
B.8	TransferResultDataT	69
Annexe C (normative) Définitions spécifiques au protocole		70
C.1	Généralités	70
C.2	Header	70
C.3	ProtocolIdentifier	70
C.4	Address	70
C.5	Manufacturer	70
C.6	DeviceModel	71
C.7	DeviceRevision	71
C.8	SerialNumber	71
C.9	Tag	71
C.10	ProfileId	71
C.11	Version	72
C.12	ProtocolSupportFile	72
Bibliographie		73
Tableau 1 – ProtocolSupportFile pour les Paquetages d'appareils FDI		44
Tableau 2 – Valeurs du catalogue pour les appareils de profil		45
Tableau 3 – Mapping des informations d'identification d'appareils		46
Tableau 4 – Mapping des informations de révisions d'appareils		46
Tableau 5 – Type de protocole GenericProtocol		47
Tableau 6 – Mapping de la propriété DeviceType héritée		47
Tableau 7 – Attributs d'identification de types d'appareils de protocole générique		48
Tableau 8 – Type ConnectionPoint pour les protocoles génériques		48
Tableau 9 – Arguments de la méthode Connect		51
Tableau 10 – Arguments de la méthode Disconnect		52
Tableau 11 – Arguments de la méthode Transfer		53
Tableau 12 – Structure EddDataTypeInfo DataType		53
Tableau 13 – Valeurs de l'EddDataTypeEnum		54
Tableau 14 – Arguments de la méthode SetAddress		55
Tableau 15 – Arguments du service Connect		56
Tableau 16 – Arguments de la méthode Transfer		58
Tableau 17 – Arguments de la méthode SetAddress		61
Tableau A.1 – Éléments de GenericNetworkT		63
Tableau A.2 – Attributs de GenericConnectionPointT		64
Tableau A.3 – Éléments de GenericConnectionPointT		64
Tableau A.4 – Attributs de GenericIdentificationT		65
Tableau B.1 – Attributs de TransferSendDataT		67
Tableau B.2 – Éléments de TransferSendDataT		67
Tableau B.3 – Éléments d'EddDataTypeInfoListT		67
Tableau B.4 – Attributs d'EddDataTypeInfoT		68
Tableau B.5 – Énumérations d'EddDataTypeT		69
Tableau B.6 – Attributs de TransferResultDataT		69

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 100: Profils – Protocoles génériques

AVANT-PROPOS

- 1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62769-100 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

FCG TS62769-100 Edition 1.1, *Field Device Integration Part 100: Profiles – Generic Protocols*, une spécification de FieldComm Group, PROFIBUS Nutzerorganisation e. V., OPC Foundation et FDT Group, a servi de base à l'élaboration de la présente norme.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
65E/739/FDIS	65E/743/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette Norme internationale.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62769, publiées sous le titre général *Intégration des appareils de terrain (FDI)*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 100: Profils – Protocoles génériques

1 Domaine d'application

La présente partie de l'IEC 62769 spécifie un profil FDI de l'IEC 62769 pour les protocoles génériques. Cela signifie que toutes les interfaces sont définies et qu'un hôte peut ajouter la prise en charge de plusieurs protocoles sans modifier sa mise en œuvre. Néanmoins, il est nécessaire de spécifier certaines définitions spécifiques au protocole (PSD) à l'aide de ce profil. L'Annexe C spécifie les PSD qu'il est nécessaire de définir par protocole, de sorte que les Paquetages d'appareils FDI, les Paquetages de communication FDI pour les Passerelles et les Serveurs de communication FDI, les Serveurs de communication FDI, les Passerelles et les Appareils prenant en charge ce type de protocole puissent fonctionner ensemble dans un hôte, quel que soit ce protocole spécifique.

NOTE Un hôte qui utilise un mécanisme propriétaire pour la communication à la place d'un Serveur de communication FDI définit ses propres moyens de gérer ce profil afin de prendre en charge plusieurs protocoles sans modifier sa mise en œuvre. La manière dont le pilote de communication est lié à l'hôte constitue une démarche propriétaire spécifique.

2 Références normatives

Les documents suivants cités dans le texte constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61804 (toutes les parties), *Blocs fonctionnels (FB) pour les procédés industriels et le langage de description électronique de produit (EDDL)*

IEC 61804-3, *Blocs fonctionnels (FB) pour les procédés industriels et le langage de description électronique de produit (EDDL) – Partie 3: Sémantique et syntaxe EDDL*

IEC 62541-100:2015, *Architecture unifiée OPC – Partie 100: Interface d'appareils*

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

IEC 62769-4, *Intégration des appareils de terrain (FDI) – Partie 4: Paquetages FDI*

IEC 62769-5, *Intégration des appareils de terrain (FDI) – Partie 5: Modèle d'Information FDI*

IEC 62769-7, *Intégration des appareils de terrain (FDI) – Partie 7: Appareils de Communication FDI*

3 Termes, définitions, termes abrégés et conventions

Pour les besoins du présent document, les termes et définitions de la série IEC 61804, de l'IEC 62541-100, de l'IEC 62769-2, de l'IEC 62769-4, de l'IEC 62769-5 et de l'IEC 62769-7 s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.1 Termes abrégés

EDD	electronic device description (description électronique de produit)
EDDL	Electronic Device Description Language (langage de description électronique de produit) (voir l'IEC 61804)
FDI™	Field Device Integration™ ¹ (intégration des appareils de terrain)
FCG	FieldComm Group
XML	Extensible Markup Language (langage de balisage extensible) (voir FCG TS62769-100, Édition 1.1, <i>Field Device Integration Part 100: Profiles – Generic Protocols</i> , disponible à l'adresse https://fieldcommgroup.org)

3.2 Conventions

3.2.1 Syntaxe EDDL

Le présent document spécifie le contenu du composant EDD faisant partie intégrante des Paquetages de communication FDI. Le contenu de la spécification utilisant la syntaxe EDDL est rédigé avec la police *Courier New*. La syntaxe EDDL est utilisée pour les déclarations de signature de méthode, de variable, de structure de données et de composant.

3.2.2 Syntaxe XML

Les exemples de syntaxe XML utilisent la police *Courier New*. La syntaxe XML est utilisée pour décrire le schéma de document XML.

Exemple: `<xs:simpleType name="ExampleType">`

3.2.3 Capitalisations

La série IEC 62769 utilise des termes en majuscules pour souligner le fait qu'ils ont une signification spécifique aux FDI.

Certains de ces termes utilisent un sigle comme préfixe, par exemple:

- Client FDI, ou
- Serveur FDI.

D'autres sont des termes composés, par exemple:

- Serveurs de communication, ou
- Paquetage de profil.

Les noms ou attributs de paramètres sont concaténés en un seul terme, les termes d'origine commençant, dans ce terme, par une majuscule, par exemple:

- ProtocolSupportFile ou
- ProtocolType.

¹ Field Device Integration (FDI) est l'appellation commerciale d'un produit distribué par FieldComm Group. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. Des produits équivalents peuvent être utilisés s'il est démontré qu'ils conduisent aux mêmes résultats.

Les noms ou attributs de paramètres peuvent également être construits avec un trait de soulignement afin de concaténer au moins deux termes, par exemple:

- DEVICE_REV ou
- DEVICE_MODEL

4 Profils des protocoles génériques

4.1 Généralités

Le présent document de profil lié à la spécification FDI de l'IEC 62769 spécifie les particularités du protocole nécessaires aux Paquetages FDI décrivant les Serveurs de communication, les Passerelles et les Appareils.

Pour les Serveurs de communication, le présent document définit les particularités du protocole qu'il est nécessaire de prendre en considération dans le Modèle d'information hébergé par les Serveurs de communication.

4.2 Profil du catalogue

4.2.1 Fichier de prise en charge du protocole

4.2.1.1 Paquetage d'appareils FDI

Les pièces jointes spécifiques au protocole sont mentionnées dans le Catalogue de paquetage défini dans l'IEC 62769-5. Bien que le présent document définisse un profil en principe adapté à un grand nombre de protocoles, il ne définit pas d'exigences relatives aux pièces jointes spécifiques au protocole. Toutefois, il n'empêche pas non plus d'utiliser ces pièces jointes. Les PSD (voir l'Annexe C) définissent les exigences relatives à la nécessité de ProtocolSupportFiles pour un protocole particulier. Toutefois, la configuration d'un appareil utilisant un Paquetage d'appareils FDI ne doit pas exiger l'utilisation d'une pièce jointe spécifique au protocole. Le Tableau 1 spécifie les paramètres de ProtocolSupportFile dans le Paquetage d'appareils FDI si un ou un grand nombre sont fournis.

Tableau 1 – ProtocolSupportFile pour les Paquetages d'appareils FDI

Paramètre	Description
Type de contenu	texte clair
Espace de noms racine	vide
Relation de la source	http://fdi-cooperation.com/2010/relationship/attachment-protocol
Nom de fichier	Non défini

4.2.1.2 Paquetages de Communication FDI

Les mêmes règles que celles des Paquetages d'appareils FDI s'appliquent.

4.2.2 Définition de CommunicationProfile

L'IEC 62769-4 définit une chaîne CommunicationProfileT pour le schéma de Catalogue XML. La chaîne est spécifique au protocole et est définie en tant que ProfileIdentifier dans les PSD (voir l'Annexe C).

4.2.3 Appareil de profil

Un Paquetage de profil doit fournir les valeurs de catalogue des appareils de profil, en permettant au Serveur FDI d'optimiser la description d'appareil générique, si une description spécifique n'est pas disponible. Les définitions du Tableau 2 portent sur le contenu du catalogue, qui est indépendant du fournisseur.

Tableau 2 – Valeurs du catalogue pour les appareils de profil

Élément	Attribut	Contenu
PackageType	—	Profil
Manufacturer	—	Vide
DeviceModel	—	Le format de DeviceModel est spécifique au protocole et les détails relatifs au format sont définis dans les PSD (voir l'Annexe C). Pour attribuer un résultat d'analyse avec un paquetage de profil, le ProfileId de ce résultat doit être mappé au DeviceModel du paquetage de profil.

4.2.4 Informations relatives à la version du protocole

L'IEC 62769-4 définit un type d'élément nommé InterfaceT pour le schéma de Catalogue XML. Le type d'élément InterfaceT contient un élément nommé Version qui est réputé donner des informations de version relatives au profil de protocole de communication appliqué. La valeur doit se conformer au schéma d'informations de version indiqué dans l'IEC 62769-4 défini dans le type d'élément VersionT. Les PSD (voir l'Annexe C) définissent le mapping des versions d'un protocole spécifique dans ce champ.

4.3 Association d'un Paquetage à un appareil

4.3.1 Mapping d'identification du type d'appareil

Le mapping d'identification du type d'appareil vise à permettre à des systèmes hôtes FDI de comparer le résultat d'analyse à la représentation topologique dans le Modèle d'information. Les systèmes hôtes FDI doivent également pouvoir déterminer le Paquetage d'appareils FDI qui correspond à une entrée d'appareil contenue dans le résultat d'analyse. Cela permet à l'utilisateur d'un système hôte FDI de synchroniser le Modèle d'information avec l'installation réelle.

Le service d'analyse mis en œuvre par le Serveur de communication (défini en 4.6.1.7) donne le résultat de l'analyse par l'intermédiaire d'un document XML (le schéma est défini à l'Article A.5).

Le service d'analyse mis en œuvre par la Passerelle (défini en 4.6.2.7) donne le résultat de l'analyse au moyen du Modèle d'information contenant les structures de données issues du contenu EDD comme cela est spécifié en 4.6.2.7.

Quel que soit le mode de présentation du résultat d'analyse, ce dernier comporte généralement une identification du type d'appareil et une identification de l'instance d'appareil.

Les systèmes hôtes FDI qui comparent la configuration topologique actuelle du réseau à la représentation topologique dans le Modèle d'information doivent être en mesure de gérer les situations suivantes:

- L'instance d'Appareil physique identifiée au niveau d'une adresse d'appareil spécifique n'est pas présente d'un point de vue logique dans le Modèle d'information (en tant qu'Instance): Permettre au système hôte FDI de rechercher le Paquetage d'appareil FDI approprié en fonction des informations du catalogue d'appareil.
- L'instance d'Appareil physique identifiée par l'adresse d'appareil est présente d'un point de vue logique dans le Modèle d'information (en tant qu'Instance): Permettre au système hôte FDI de comparer les informations relatives au type d'appareil présentées dans le résultat de l'analyse (voir l'identification à l'Article A.5) et les informations spécifiques relatives au type d'appareil de l'Instance présentent dans le Modèle d'information.

Le Paquetage d'appareils FDI contient les informations relatives à l'identification du type d'appareil, qui peuvent être comparées aux résultats de l'analyse en fonction du schéma de catalogue de l'IEC 62769-4 définissant les types d'éléments (simples) "DeviceModel" et

"Manufacturer". Ces deux types sont utilisés dans les types d'éléments (complexes) "Protocol" et "RegDeviceType".

En conséquence du déploiement du Paquetage FDI, les informations relatives au Paquetage FDI sont présentes dans le Modèle d'information, comme le FunctionalGroup Identification spécifié contenant SerialNumber et Tag (voir 4.4.3).

Le mapping entre les différentes sources de données d'identification d'appareil est décrit dans le Tableau 3. Étant donné que les résultats de l'analyse fournis par le Serveur de communication ou la Passerelle peuvent transmettre les données générées par l'appareil (micrologiciel), le mapping d'identification du type d'appareil doit être pris en charge en fournissant les données correspondantes dans le Catalogue et le Modèle d'information contenus dans le Paquetage d'appareils FDI.

Tableau 3 – Mapping des informations d'identification d'appareils

Paquetage d'appareils FDI	Modèle d'information	Résultat d'analyse fourni par le Serveur de communication	Résultat d'analyse fourni par la Passerelle
Type spécifié par le catalogue Manufacturer	FunctionalGroup: Identification Nom d'exploration: Manufacturer	Élément (chemin): ConnectionPoint/Identification Attribut: Manufacturer	COLLECTION ConnectionPoint. Identification: Manufacturer
Type spécifié par le catalogue DeviceModel	FunctionalGroup: Identification Nom d'exploration: DeviceModel	Élément (chemin): ConnectionPoint/Identification Attribut: DeviceModel	COLLECTION ConnectionPoint. Identification: DeviceModel

Étant donné que tous les protocoles destinés à être utilisés avec ce profil pour les protocoles génériques ne peuvent pas prendre en charge un mécanisme de reconnaissance obligatoire permettant d'identifier le type d'appareil (Manufacturer et DeviceModel), les résultats d'analyse offrent la possibilité d'exclure l'identification de l'appareil et de ne donner que l'adresse. Dans ce cas, certains mécanismes spécifiques à l'hôte peuvent être utilisés pour attribuer le paquetage FDI souhaité à l'appareil (par interaction de l'utilisateur, par exemple).

Étant donné que certains protocoles ne disposent même pas des capacités obligatoires permettant d'identifier la présence d'un appareil associé à une adresse de protocole spécifique, il convient que les hôtes donnent la possibilité à certains utilisateurs d'ajouter des appareils en donnant manuellement des informations d'adresse.

4.3.2 Mapping de révisions de type d'appareil

L'IEC 62769-4 envisage un concept permettant de déterminer la compatibilité entre un Paquetage d'appareils FDI et un appareil. L'IEC 62769-4 spécifie un processus de gestion du cycle de vie s'appuyant sur une seule information de version fournie pour l'ensemble de l'appareil. Elle est capturée dans le DeviceRevision (voir le Tableau 4). Le mapping des informations de version est spécifique au protocole et il est nécessaire de le définir dans les PSD (voir l'Annexe C).

Tableau 4 – Mapping des informations de révisions d'appareils

Paquetage d'appareils FDI	Modèle d'information	Résultat d'analyse fourni par le Serveur de communication	Résultat d'analyse fourni par la Passerelle
Type spécifié par le catalogue ListOfSupportedDeviceRevisions	FunctionalGroup: Identification Nom d'exploration: DeviceRevision	Élément (chemin): ConnectionPoint/Identification Attribut: DeviceRevision	COLLECTION ConnectionPoint. Identification: DeviceRevision

4.4 Mapping du Modèle d'information

4.4.1 Définition de ProtocolType

Dans le Tableau 5, un sous-type de ProtocolType est défini pour identifier la communication de réseau utilisant ce profil.

Tableau 5 – Type de protocole GenericProtocol

Attribut	Valeur				
BrowseName	GenericProtocol				
IsAbstract	False				
Références	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Sous-type du ProtocolType défini dans l'IEC 62541-100.					
HasProperty	Variable	ProtocolIdentifier	String	PropertyType	Obligatoire

La Variable obligatoire ProtocolIdentifier définit le protocole concret représenté à l'aide du type GenericProtocol. Elle doit correspondre au ProtocolIdentifier défini pour le CommunicationProfile en 4.2.2. La chaîne est spécifique au protocole et est définie en tant que ProfileIdentifier dans les PSD (voir l'Annexe C).

4.4.2 Mapping de DeviceType

Chaque type d'appareil hérite des propriétés de DeviceType. Le mapping des propriétés héritées de DeviceType est défini dans le Tableau 6. Noter que seuls les attributs définis à l'Annexe C, et par conséquent attendus par chaque protocole générique, sont utilisés. Les protocoles spécifiques peuvent fournir, par exemple, un SoftwareRevision, mais comme il n'est pas accessible par l'hôte, ce profil ne l'utilise pas.

Tableau 6 – Mapping de la propriété DeviceType héritée

Propriété	Mapping de protocole générique
SerialNumber	SerialNumber (voir l'Annexe C)
RevisionCounter	-1 (non défini)
Manufacturer	Chaîne issue du catalogue de paquetages FDI (ManufacturerName de PackageT)
Model	Chaîne issue du catalogue de paquetages FDI (Nom de DeviceTypeT, qui est un nom localisé)
DeviceManual	chaîne de texte vide (non prise en charge) ^a
DeviceRevision	DeviceRevision (voir l'Annexe C)
SoftwareRevision	chaîne vide (non définie)
HardwareRevision	chaîne vide (non définie)
^a Les manuels de l'appareil sont présentés sous forme de pièces jointes du Paquetage d'appareils FDI.	

4.4.3 Définition du FunctionalGroup "Identification"

Comme cela est indiqué dans l'IEC 62541-100:2015, 5.3, chaque représentation d'appareil dans le Modèle d'information hébergé par le Serveur FDI doit contenir un FunctionalGroup spécifique au protocole nommé "Identification". Les paramètres de ce FunctionalGroup sont définis pour les types d'appareils de protocole générique, comme suit:

Tableau 7 – Attributs d'identification de types d'appareils de protocole générique

BrowseName	DataType	Obligatoire/Facultatif
Manufacturer	String	Obligatoire
DeviceModel	String	Obligatoire
SerialNumber	String	Facultatif
Tag	String	Facultatif
DeviceRevision	UInt16	Facultatif
ProfileId	String	Facultatif

Les instances BaseDataVariable doivent être créées à partir des déclarations de VARIABLE avec des identifiants correspondant aux noms d'exploration indiqués dans le Tableau 7.

4.5 Éléments topologiques

4.5.1 Définition de ConnectionPoint

Le GenericConnectionPoint de type ConnectionPoint doit être utilisé pour paramétrer les points d'accès au réseau utilisant ces protocoles génériques. Le GenericConnectionPoint de type ConnectionPoint est un sous-type du type abstrait ConnectionPointType défini dans l'IEC 62541-100. Le Tableau 8 spécifie la représentation du GenericConnectionPoint dans l'AddressSpace.

Tableau 8 – Type ConnectionPoint pour les protocoles génériques

Attribut	Valeur				
BrowseName	GenericConnectionPoint				
IsAbstract	False				
Références	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Sous-type du ConnectionPointType défini dans l'IEC 62541-100.					
HasProperty	Variable	Address	String	PropertyType	Obligatoire
HasProperty	Variable	ProtocolIdentifier	String	PropertyType	Obligatoire

Le GenericConnectionPoint de type ConnectionPoint doit être décrit par un élément EDD contenu dans un Paquetage FDI lié à l'Appareil de communication qui peut gérer un réseau de protocole générique. Les propriétés réelles de ConnectionPoint sont déclarées par des constructions de VARIABLE regroupées dans une COLLECTION nommée ConnectionPoint. Ce profil ne doit contenir que le CONNECTION_POINT_ADDRESS, mappé à OPC UA Property Address. De plus, le PROTOCOL spécifié par le COMPONENT doit être mappé à la propriété ProtocolIdentifier. Le code source EDDL suivant est un exemple représentatif du Connection Point d'un ExampleProtocol. Le ProtocolIdentifier défini par les PSD (voir l'Annexe C) doit être utilisé comme le nom PROTOCOL dans l'EDD.

```

COMPONENT ConnectionPoint_Generic
{
  LABEL "Generic Connection Point";
  CLASSIFICATION NETWORK_CONNECTION_POINT;
  CAN_DELETE FALSE;
  PROTOCOL ExampleProtocol;
  CONNECTION_POINT ConnectionPoint;
}

```

```

VARIABLE Address

```

```

{
    LABEL "Address";
    HELP "Address of the device";
    TYPE EUC(<protocol-specific>);
    CLASS LOCAL;
}

COLLECTION ConnectionPoint
{
    LABEL "Connection Point";
    MEMBERS
    {
        CONNECTION_POINT_ADDRESS, Address;
    }
}

```

4.5.2 Définition de l'appareil de communication

Conformément à l'IEC 62769-7, chaque Paquetage de communication FDI doit contenir un élément EDD qui décrit l'appareil de communication. Le code source EDDL suivant est un exemple représentatif d'un Serveur de communication.

```

COMPONENT Generic_Communication_Server
{
    LABEL "Generic communication server";
    PRODUCT_URI "urn:Company Name:Product Name";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS
    {
        Generic_Communication_Device_Setup
    }
}

COMPONENT_RELATION Generic_Communication_Device_Setup
{
    LABEL "Relation between Device and communication device";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        Generic_Communication_Device{AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 4;
}

```

Conformément à l'IEC 62769-7, chaque Paquetage de communication FDI doit contenir au moins un élément EDD qui décrit au moins un composant de l'appareil de communication. Le code source EDDL suivant est un exemple pour un appareil de communication de protocole générique:

```

COMPONENT Generic_Communication_Device
{
    LABEL "Generic communication device";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { Generic_Service_Provider_Relation }
}

COMPONENT_RELATION Generic_Service_Provider_Relation
{
    LABEL "Relation to communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        Generic_Service_Provider{AUTO_CREATE 1;}
    }
}

```

```

    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

Dans un appareil de communication réel, il est nécessaire d'adapter le `ConnectionPoint_Generic` en fonction du protocole pris en charge et des définitions du point de connexion associé données en 4.5. L'attribut `BYTE_ORDER` ne doit pas être utilisé pour ce profil car le traitement de l'ordre des octets doit être assuré dans la logique applicative de la passerelle.

4.5.3 Définition du fournisseur de services de communication

Conformément à l'IEC 62769-7, chaque Paquetage de communication FDI doit contenir au moins un élément EDD qui décrit au moins un composant du fournisseur de services de communication. Le code source EDDL ci-dessous est un exemple pour un composant de fournisseur de services de communication de protocole générique.

La référence du composant (`ConnectionPoint_Generic`) correspond à la définition du point de connexion associé du 4.5). L'attribut `BYTE_ORDER` ne doit pas être utilisé pour ce profil car le traitement de l'ordre des octets doit être assuré dans la logique applicative de la passerelle.

```

COMPONENT Generic_Service_Provider
{
    LABEL "Generic Protocol communication service provider";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
    COMPONENT_RELATIONS
    {
        Generic_Service_Provider_Connection_Point_Relation
    }
}

COMPONENT_RELATION Generic_Service_Provider_Connection_Point_Relation
{
    LABEL "Relation between communication service provider and Connection Point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {Address}
    COMPONENTS
    {
        ConnectionPoint_Generic{ AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

4.5.4 Définition du réseau

Conformément à l'IEC 62769-7, chaque Paquetage de communication FDI doit contenir au moins un élément EDD qui décrit les contraintes de configuration du réseau à l'aide de la construction de composant.

```

COMPONENT Network_Generic
{
    LABEL "Generic Network";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK;
    COMPONENT_RELATIONS
    {
        Generic_Network_Connection_Point_Relation
    }
}

COMPONENT_RELATION Generic_Network_Connection_Point_Relation
{
    LABEL "Relation between network and Connection Point";
    RELATION_TYPE CHILD_COMPONENT;
}

```

```

ADDRESSING {Address}
COMPONENTS
{
    ConnectionPoint_Generic
}
MINIMUM_NUMBER 1;
MAXIMUM_NUMBER 32;
}

```

4.6 Méthodes

4.6.1 Méthodes pour les Serveurs de communication FDI

4.6.1.1 Généralités

Le Serveur de communication contenu dans le Modèle d'information doit mettre en œuvre les services selon les signatures de méthode décrites en 4.6.1.

4.6.1.2 Connect

Signature:

```

Connect (
    [in]  ByteString      CommunicationRelationId,
    [in]  String          Address,
    [out] Int32           ServiceError);

```

Le Tableau 9 décrit les arguments.

Tableau 9 – Arguments de la méthode Connect

Argument	Description
CommunicationRelationId	La valeur de l'argument contient le nodeId du ConnectionPoint qui représente la connexion entre un appareil et un réseau physique qui est directement connecté au matériel du Serveur de communication. Le nodeId permet de déterminer la relation parent-enfant directe.
Address	Le nom de l'argument doit correspondre au nom d'attribut correspondant défini pour le ConnectionPoint qui est décrit par un élément EDD correspondant spécifié en 4.5. La valeur de l'argument contient l'adresse de réseau de l'appareil spécifique au protocole, qui est unique dans le segment de réseau.
ServiceError	0: OK / exécution terminée, connexion établie avec succès -1: Échec de la connexion / annulée par l'appelant -3: Échec de la connexion / appareil introuvable -4: Échec de la connexion/ adresse de l'appareil non valide -5: Échec de la connexion/ identification de l'appareil non valide

4.6.1.3 Disconnect

Signature:

```

Disconnect (
    [in]  ByteString      CommunicationRelationId,
    [out] Int32           ServiceError);

```

Le Tableau 10 décrit les arguments.

Tableau 10 – Arguments de la méthode Disconnect

Argument	Description
CommunicationRelationId	La valeur de l'argument contient le nodeId du ConnectionPoint qui représente la connexion entre un appareil et un réseau physique directement connecté au matériel du Serveur de communication. Le nodeId permet de déterminer la relation parent-enfant directe.
ServiceError	0: OK / la déconnexion a abouti -1: Échec de la déconnexion / il n'existe aucune relation de communication -2: Échec de la déconnexion / identifiant de relation de communication non valide

4.6.1.4 Transfer

Signature

```

Transfer (
    [in]   ByteString      CommunicationRelationId,
    [in]   String          Header,
    [in]   ByteString      RequestData,
    [in]   EddDataTypeInfo[] RequestDataTypes,
    [in]   EddDataTypeInfo[] ResponseDataTypes,
    [out]  ByteString      ResponseData,
    [out]  ByteString      RESPONSE_CODES,
    [out]  Int32            ServiceError);

```

Le Tableau 11 décrit les arguments.

IECNORM.COM : Click to view the full PDF of IEC 62769-100:2020

Tableau 11 – Arguments de la méthode Transfer

Argument	Description
CommunicationRelationId	La valeur de l'argument contient le nodeId du ConnectionPoint qui représente la connexion entre un appareil et un réseau physique à l'intérieur du Modèle d'information.
Header	Informations spécifiques au protocole relatives aux données à transférer. Le Header est utilisé dans les COMMAND de l'EDD, et les PSD (voir l'Annexe C) définissent le format de cette chaîne en fonction du protocole. Le premier service de communication qui traite cette chaîne doit la valider en fonction de son format défini dans les PSD.
RequestData	Le nom de l'argument doit correspondre au nom REQUEST du sous-élément COMMAND connexe. Le flux d'octets soumis par l'intermédiaire de cet argument est créé à partir des définitions fournies par l'élément REQUEST du COMMAND correspondant qui doit être traité.
RequestDataTypes	Cette matrice contient des informations relatives aux types de données utilisés pour créer la RequestData ByteString. L'ordre des octets utilisés dans le RequestData est spécifique à l'EDD, les informations permettant à la passerelle ou au serveur de communication de le remplacer par un ordre d'octets spécifique au protocole des types de données ou d'appliquer d'autres caractéristiques spécifiques au protocole (l'ordre des bits, par exemple). La longueur globale de tous les types de données fournis dans le RequestDataTypes doit correspondre à la longueur du RequestData.
ResponseDataTypes	Cette matrice contient des informations relatives aux types de données attendus lors de la réception du ResponseData ByteString. Les types de données attendus se trouvent dans l'EDD de l'appareil et, par conséquent, il est nécessaire que la passerelle ou le serveur de communication extraie ces informations afin d'appliquer les caractéristiques spécifiques au protocole aux données de réponse et qu'il/elle sache comment séparer ces données dans une matrice de ByteString. La longueur globale de tous les types de données fournis dans le ResponseDataTypes doit correspondre à la longueur du ResponseData.
ResponseData	Le nom de l'argument doit correspondre au nom REPLY du sous-élément COMMAND connexe. Le flux d'octets renvoyé par cet argument s'applique aux définitions fournies par l'élément REPLY du COMMAND correspondant qui doit être traité.
RESPONSE_CODES	Le nom de l'argument doit correspondre au nom RESPONSE_CODES du sous-élément COMMAND. La valeur de l'argument transmet les octets spécifiques de la réponse du service de communication.
ServiceError	0: OK / exécution terminée -1: Échec de Transfer / annulé par l'appelant -3: Échec de Transfer / il n'existe aucune relation de communication -4: Échec de Transfer / identifiant de relation de communication non valide -5: Échec de Transfer / contenu de sendData non valide -6: Échec de Transfer / format de receiveData non valide

L'EddDataTypeInfo DataType définit les informations relatives à un type de données EDD d'un COMMAND. Ses éléments sont définis dans le Tableau 12.

Tableau 12 – Structure EddDataTypeInfo DataType

Nom	Type	Description
EddDataTypeInfo	structure	Cette structure spécifie les informations relatives au type de données utilisé dans un EDD COMMAND.
eddDataType	EddDataTypeEnum	EddDataType utilisé.
size	UInt32	Taille de l'eddDataType. Elle doit toujours être indiquée, même si cela n'est pas explicitement spécifié par l'EDD. Dans ce cas, la valeur par défaut de ce type de données doit être fournie. Pour les types de données dont la taille ne peut pas être définie (BOOLEAN, par exemple), la valeur doit être 0.

L'EddDataTypeEnum DataType est une énumération qui définit les types de données EDD possibles. Ses valeurs sont définies dans le Tableau 13.

Tableau 13 – Valeurs de l'EddDataTypeEnum

Valeur	Description
BOOLEAN	Type de données conforme à l'IEC 61804-3
DOUBLE	Type de données conforme à l'IEC 61804-3
FLOAT	Type de données conforme à l'IEC 61804-3
INTEGER	Type de données conforme à l'IEC 61804-3
UNSIGNED_INTEGER	Type de données conforme à l'IEC 61804-3
DATE	Type de données conforme à l'IEC 61804-3
DATE_AND_TIME	Type de données conforme à l'IEC 61804-3
DURATION	Type de données conforme à l'IEC 61804-3
TIME	Type de données conforme à l'IEC 61804-3
TIME_VALUE	Type de données conforme à l'IEC 61804-3
BIT_ENUMERATED	Type de données conforme à l'IEC 61804-3
ENUMERATED	Type de données conforme à l'IEC 61804-3
ASCII	Type de données conforme à l'IEC 61804-3
BITSTRING	Type de données conforme à l'IEC 61804-3
EUC	Type de données conforme à l'IEC 61804-3
OCTET	Type de données conforme à l'IEC 61804-3
PACKED_ASCII	Type de données conforme à l'IEC 61804-3
PASSWORD	Type de données conforme à l'IEC 61804-3
VISIBLE	Type de données conforme à l'IEC 61804-3

4.6.1.5 GetPublishedData

Cette méthode n'est pas prise en charge par le profil de protocole générique.

4.6.1.6 SetAddress

Signature

```
SetAddress (
    [in] String OldAddress,
    [in] String NewAddress,
    [out] Int32 ServiceError);
```

Le Tableau 14 décrit les arguments.

Tableau 14 – Arguments de la méthode SetAddress

Argument	Description
OldAddress	La valeur de l'argument contient l'adresse actuelle d'un appareil. La même sémantique que celle du paramètre Address de la méthode Connect s'applique à ce champ.
NewAddress	La valeur de l'argument contient la nouvelle adresse d'un appareil. La même sémantique que celle du paramètre Address de la méthode Connect s'applique à ce champ.
ServiceError	0: OK / exécution terminée avec succès -1: Échec de SetAddress / annulé par l'appelant -3: Échec de SetAddress / non initialisé -4: Échec de SetAddress / non connecté à un réseau -5: Échec de SetAddress / aucun appareil détecté répondant à oldAddress -6: Échec de SetAddress / erreur d'adresse dupliquée -7: Échec de SetAddress / l'appareil n'a pas accepté la nouvelle adresse -8: Échec de SetAddress / oldAddress non valide (en matière de syntaxe, de type de données, de format de données, etc.) -9: Échec de SetAddress / newAddress non valide (en matière de syntaxe, de type de données, de format de données, etc.) -10: Échec de SetAddress / impossible à l'état connecté

4.6.1.7 Scan

La signature de méthode spécifiée dans l'IEC 62769-7 s'applique. Le schéma topologyScanResult correspondant est spécifié à l'Article A.2.

4.6.1.8 ResetScan

La signature de méthode spécifiée dans l'IEC 62769-7 s'applique.

4.6.2 Méthodes pour les Passerelles

4.6.2.1 Généralités

Les signatures de méthode définies en 4.6.2 s'appliquent. Les méthodes doivent être mises en œuvre dans l'élément EDD (IEC 62769-4) contenu dans un Paquetage FDI lié à une Passerelle contenant les définitions de l'appareil de communication.

4.6.2.2 Connect

Le présent paragraphe décrit la mise en œuvre spécifique au protocole générique du service Connect spécifié dans l'IEC 62769-7.

```

METHOD BeginConnect (
    DD_STRING          CommunicationRelationId,
    DD_STRING          Address,
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

```

METHOD EndConnect (
    DD_STRING          CommunicationRelationId,
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

METHOD CancelConnect (
    DD_STRING          CommunicationRelationId,
    unsigned long      ServiceId,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

Le Tableau 15 décrit les arguments.

Tableau 15 – Arguments du service Connect

Argument	Description
CommunicationRelationId	La valeur de l'argument contient le nodeId du ConnectionPoint qui représente la connexion entre un appareil et un réseau physique directement connecté au matériel du Serveur de communication. Le nodeId permet de déterminer la relation parent-enfant directe.
Address	Le nom de l'argument doit correspondre au nom d'attribut correspondant défini pour le ConnectionPoint qui est décrit par un élément EDD correspondant spécifié en 4.5. La valeur de l'argument contient l'adresse de réseau de l'appareil spécifique au protocole, qui est unique dans le segment de réseau.
ServiceId	Le code de transaction de service établit la relation entre la demande de service et la réponse correspondante.
DelayForNextCall	La valeur spécifie un délai en ms pour limiter le cycle d'appel d'EndConnect qui ne doit pas être plus court que spécifié dans la valeur de l'argument.
ServiceError	1: OK / la fonction a démarré de manière asynchrone, le résultat doit être interrogé avec EndConnect 0: OK / exécution terminée, connexion établie avec succès -1: Échec de la connexion / annulé par l'appelant -2: Échec de l'appel / ID de service inconnu -3: Échec de la connexion / appareil introuvable -4: Échec de la connexion / adresse de l'appareil non valide -5: Échec de la connexion / identification de l'appareil non valide

4.6.2.3 Disconnect

Le présent paragraphe décrit la mise en œuvre spécifique au protocole générique du service Disconnect spécifié dans l'IEC 62769-7.

```

METHOD Disconnect(
    DD_STRING CommunicationRelationId,
    long      &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

Tous les arguments de la méthode Disconnect sont décrits dans le Tableau 10.

4.6.2.4 Transfer

Le présent paragraphe décrit la mise en œuvre spécifique au protocole générique du service Transfer spécifié dans l'IEC 62769-7. En raison des limitations de capacités des méthodes EDD (impossibilité de transférer toutes les collections d'une taille dynamique), ce profil exige certaines manipulations supplémentaires en cas de transfert par l'intermédiaire de Passerelles. Outre les appels de la méthode EDD, un hôte a en premier lieu besoin de fournir les informations relatives au type de données de la requête et de la réponse. Il est nécessaire que l'hôte respecte l'ordre suivant:

- 1) Ajouter les informations relatives au type de données de la requête et de la réponse dans une liste définie par l'EDD de la passerelle (appelée DATA_TYPE_INFO_PER_SERVICE_ID). L'entrée doit inclure le ServiceId également utilisé dans les appels de méthode.
- 2) Appeler BeginTransfer à l'aide du même ServiceId et suivre le modèle d'appel asynchrone défini dans l'IEC 62769-7.
- 3) Lorsque l'hôte reçoit la réponse, il lui revient de retirer l'entrée de la liste. Ainsi, seul l'hôte manipule la liste, la logique applicative de la passerelle accédant uniquement à la liste sans la modifier afin d'extraire les informations relatives aux types de données.

```

METHOD BeginTransfer(
    DD_STRING      CommunicationRelationId,
    DD_STRING      Header,
    DD_STRING      RequestData,
    unsigned long   ServiceId,
    DD_STRING      &ResponseData,
    DD_STRING      &RESPONSE_CODES,
    long           &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

METHOD EndTransfer(
    DD_STRING      CommunicationRelationId,
    unsigned long   ServiceId,
    DD_STRING      &ResponseData,
    DD_STRING      &RESPONSE_CODES,
    unsigned long   &DelayForNextCall,
    long           &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

```

METHOD CancelTransfer (
    DD_STRING          CommunicationRelationId,
    DD_STRING          &ResponseData,
    DD_STRING          &RESPONSE_CODES,
    unsigned long      ServiceId,
    long               &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}
    
```

Le Tableau 16 décrit les arguments.

Tableau 16 – Arguments de la méthode Transfer

Argument	Description
CommunicationRelationId	La valeur de l'argument contient le nodelid du ConnectionPoint qui représente la connexion entre un appareil et un réseau physique à l'intérieur du Modèle d'information.
Header	Informations spécifiques au protocole relatives aux données à transférer. Le Header est utilisé dans les COMMAND de l'EDD, et les PSD (voir l'Annexe C) définissent le format de cette chaîne en fonction du protocole. Le premier service de communication qui traite cette chaîne doit la valider en fonction de son format défini dans les PSD.
RequestData	Le nom de l'argument doit correspondre au nom REQUEST du sous-élément COMMAND connexe. Le flux d'octets soumis par l'intermédiaire de cet argument est créé à partir des définitions fournies par l'élément REQUEST du COMMAND correspondant qui doit être traité. Le flux d'octets est séparé en plusieurs parties pour les différents types de données demandés. Cela permet d'envisager un ordre d'octets spécifique au protocole des types de données particuliers dans la passerelle ou le serveur de communication.
ResponseData	Le nom de l'argument doit correspondre au nom REPLY du sous-élément COMMAND connexe. Le flux d'octets renvoyé par cet argument s'applique aux définitions fournies par l'élément REPLY du COMMAND correspondant qui doit être traité. Les données sont séparées en plusieurs chaînes d'octets en fonction du champ ResponseDataTypes. Cela permet à la passerelle ou au serveur de communication d'envisager un ordre d'octets spécifique au protocole des types de données.
RESPONSE_CODES	Le nom de l'argument doit correspondre au nom RESPONSE_CODES du sous-élément COMMAND. La valeur de l'argument transmet les octets spécifiques de la réponse du service de communication.
ServiceId	Le code de transaction de service établit la relation entre la demande de service et la réponse correspondante.
DelayForNextCall	La valeur spécifie un délai en ms pour limiter le cycle d'appel d'EndTransfer qui ne doit pas être plus court que spécifié dans la valeur de l'argument.
ServiceError	1: OK / la fonction a démarré de manière asynchrone, le résultat doit être interrogé avec EndTransfer 0: OK / exécution terminée -1: Échec de Transfer / annulé par l'appelant -2: Échec de l'appel / ID de service inconnu -3: Échec de Transfer / il n'existe aucune relation de communication -4: Échec de Transfer / identifiant de relation de communication non valide -5: Échec de Transfer / contenu de sendData non valide -6: Échec de Transfer / format de receiveData non valide

La liste utilisée pour ajouter les informations relatives au type de données pour que la logique applicative de la passerelle puisse procéder aux transferts de type de données nécessaires doit s'appeler DATA_TYPE_INFO_LIST. Elle est définie comme suit:

VARIABLE EddDataTypeEnum

```
{
  LABEL "Edd Data Type";
  CLASS LOCAL;
  TYPE ENUMERATED
  {
    { 0x00, "BOOLEAN" }
    { 0x01, "DOUBLE" }
    { 0x02, "FLOAT" }
    { 0x03, "INTEGER" }
    { 0x04, "UNSIGNED_INTEGER" }
    { 0x05, "DATE" }
    { 0x06, "DATE_AND_TIME" }
    { 0x07, "DURATION" }
    { 0x08, "TIME" }
    { 0x09, "TIME_VALUE" }
    { 0x0a, "BIT_ENUMERATED" }
    { 0x0b, "ENUMERATED" }
    { 0x0c, "ASCII" }
    { 0x0d, "BITSTRING" }
    { 0x0e, "EUC" }
    { 0x10, "OCTET" }
    { 0x11, "PACKED_ASCII" }
    { 0x12, "PASSWORD" }
    { 0x13, "VISIBLE" }
  }
}
```

VARIABLE SizeV

```
{
  LABEL "Size of data type";
  TYPE UNSIGNED_INTEGER(4);
  CLASS LOCAL;
}
```

COLLECTION DataTypeInfo

```
{
  MEMBERS
  {
    Size, SizeV;
    EddDataType, EddDataTypeEnum;
  }
}
```

LIST DataTypeInfoList

```
{
  TYPE DataTypeInfo;
}
```

VARIABLE ServiceIdentification

```
{
  LABEL "Identification of service call";
  TYPE UNSIGNED_INTEGER(4);
  CLASS LOCAL;
}
```

COLLECTION DataTypeInfoPerServiceId

```
{
  MEMBERS
  {
    ServiceId, ServiceIdentification;
    RequestDataTypes, DataTypeInfoList;
    ResponseDataTypes, DataTypeInfoList;
  }
}
```

```

    }
}
LIST DATA_TYPE_INFO_PER_SERVICE_ID
{
    TYPE DataTypeInfoPerServiceId;
}

```

4.6.2.5 GetPublishedData

Cette méthode n'est pas prise en charge dans le profil de protocole générique.

4.6.2.6 SetAddress

Le présent paragraphe décrit la mise en œuvre spécifique au protocole générique du service SetAddress spécifié dans l'IEC 62769-7.

```

BeginSetAddress (
    DD_STRING          OldAddress,
    DD_STRING          NewAddress,
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long              &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

EndSetAddress (
    unsigned long      ServiceId,
    unsigned long      &DelayForNextCall,
    long              &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

```

Le Tableau 17 décrit les arguments.