IEC 62769-101-1

Edition 2.0 2020-11
REDLINE VERSION

INTERNATIONAL STANDARD

colour inside

Field device integration (FDI) – Part 101-1: Profiles – Foundation Fieldbus H1

IEC 62769-101-1:2020-11 RLV(en)

IEC Central Office
3, rue de Varembé
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 It also gives information on projects, replaced and withdrawn publications.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.

IEC 62769-101-1

Edition 2.0 2020-11
REDLINE VERSION

INTERNATIONAL STANDARD

Field device integration (FDI) – Part 101-1: Profiles – Foundation Fieldbus H1

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-9104-7

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

FIELD DEVICE INTEGRATION (FDI) –

Part 101-1: Profiles – Foundation Fieldbus H1

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.

This redline version of the official IEC Standard allows the user to identify the changes made to the previous edition. A vertical bar appears in the margin wherever a change has been made. Additions are in green text, deletions are in strikethrough red text.

International Standard IEC 62769-101-1 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2015. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) support for generic protocol extension for faster adoption of other technologies;
b) support for Package developers to build EDDs targeted for today's EDD bases system under a single development tool.

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

CDV	Report on voting
65E/620/CDV	65E/683/RVC

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.

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

INTRODUCTION

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning

a) Method for the Supplying and Installation of Device-Specific Functionalities, see Patent Family DE10357276;

b) Method and device for accessing a functional module of automation system, see Patent Family EP2182418;

c) Methods and apparatus to reduce memory requirements for process control system software applications, see Patent Family US2013232186;

d) Extensible Device Object Model, see Patent Family US12/893,680.

IEC takes no position concerning the evidence, validity and scope of this patent right.

The holders of these patent rights have assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from:

a) ABB Research Ltd
Claes Rytoft
Affolterstrasse 4
Zurich, 8050
Switzerland

b) Phoenix Contact GmbH & Co KG
Intellectual Property, Lycenses & Standards
Flachsmarktstrasse 8, 32825 Blomberg
Germany

c) Fisher Controls International LLC
John Dilger, Emerson Process Management LLLP
301 S. 1st Avenue, Marshaltown, Iowa 50158
USA

d) Rockwell Automation Technologies, Inc.
1 Allen-Bradley Drive
Mayfield Heights, Ohio 44124
USA

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

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

FIELD DEVICE INTEGRATION (FDI) –

Part 101-1: Profiles – Foundation Fieldbus H1

1 Scope

This part of IEC 62769 specifies ~~an FDI profile of~~ the IEC 62769 profile for IEC 61784-1_CP 1/1 (FOUNDATION™ Fieldbus H1)[1].

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 61158-5-9:2014, *Industrial communication networks – Fieldbus specifications – Part 5-9: Application layer service definition – Type 9 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus Profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC/IEEE 8802-3*

IEC 61784-3:~~2010~~2016, *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions*

IEC 61804 (all parts), *Function blocks (FB) for process control and electronic device description language (EDDL)*

IEC 62541-6, *OPC unified architecture – Part 6: Mappings*

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

~~NOTE IEC 62769-1 is technically identical to FDI-2021.~~

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

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

~~NOTE IEC 62769-2 is technically identical to FDI-2022.~~

IEC 62769-4~~:2015~~, *Field Device Integration (FDI) – Part 4: FDI Packages*

~~NOTE IEC 62769-4 is technically identical to FDI-2024.~~

IEC 62769-5~~:2015~~, *Field Device Integration (FDI) – Part 5: FDI Information Model*

---

[1] FOUNDATION™ Fieldbus is the trade name of the non-profit consortium Fieldbus Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade name. Use of the trade name requires permission of the trade name holder.

~~NOTE IEC 62769-5 is technically identical to FDI-2025.~~

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

~~NOTE IEC 62769-6 is technically identical to FDI-2026.~~

IEC 62769-7~~:42015~~, *Field Device Integration (FDI) – Part 7: FDI Communication Devices*

~~NOTE IEC 62769-7 is technically identical to FDI-2027.~~

3 Terms, definitions, abbreviated terms and ~~acronyms~~ conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 61158-5-9, IEC 61784-1, IEC 61784-2, IEC 61784-3, IEC 61804 (all parts), IEC 62541-6, IEC 62541-100, IEC 62769-1, IEC 62769-2, IEC 62769-4, IEC 62769-5, IEC 62769-6, 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

3.2 Abbreviated terms ~~and acronyms~~

For the purposes of this document, the following abbreviated terms apply:

CFF common file format
CP communication profile (see IEC 61784-1 or IEC 61784-2)
CPF communication profile family (see IEC 61784-1 or IEC 61784-2)
EDD Electronic Device Description (see IEC 61804 (all parts))
FB Function Block
IM Information Model
SMIB System Management Information Base
VFD virtual field device

3.3 Conventions

3.3.1 EDDL syntax

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

3.3.2 XML syntax

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

Example `<xsd:simpleType name="Example">`

3.3.3 Capitalizations

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

Some of these terms use an acronym 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 to 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:

- PROFILE_ID, or
- Profibus_PA_Network.

4 Profile for CP 1/1 (FOUNDATION™ H1)

4.1 General

This profile specifies the protocol specifics needed for FDI Packages describing Communication Servers, gateways and devices. Requirements for Direct Access transfer service parameters are given in Annex B.

4.2 Catalog profile

4.2.1 Protocol support file

~~5.2.1.1 Capability file~~

Each CP 1/1 FDI Device Package shall contain a capability file. The capability file part is described in Table 1.

Table 1 – Capability File part

Parameter	Description
Content Type:	txt/plain
Root Namespace:	Not applicable
Source Relationship:	http://fdi-cooperation.com/2010/relationships/attachment-protocol
Filename:	Use file extension .CFF

4.2.2 CommunicationProfile definition

IEC 62769-4 defines a ~~CommunicationProfile enumeration~~ CommunicationProfileT string type for the Catalog XML schema. Table 2 defines the CP 1/1 specific values for this ~~enumeration~~ string.

Table 2 – CommunicationProfile definition

CommunicationProfile	Description
foundation_h1	CP 1/1 device type with a Function Block application

4.2.3 Profile device

Not supported in this document.

4.2.4 Protocol version information

IEC 62769-4 defines an element type named InterfaceT for the Catalog XML Schema. Element type InterfaceT contains an element named Version which is supposed to provide version information about the applied communication protocol profile. The value follows the IEC 62769-4 defined version information schema defined in element type VersionT.

The major version part of VersionT shall be set to the ITK_VER parameter. The minor and builds parts shall be set to 0.

EXAMPLE For ITK_VER 5, the value for InterfaceT is 5.0.0.

4.3 Associating a Package with a CP 1/1 device

4.3.1 Device type identification mapping

CP 1/1 device types are uniquely identified by the parameters MANUFAC_ID, DEVICE_TYPE and DEV_REV found in the Resource Block. These parameters are used to associate a given device instance to an FDI Device Package. These parameters are mapped to the FDI Device Package Catalog according to Table 3.

Table 3 – Device type catalog mapping

Catalog Element	CP Mapping
Manufacturer element of InterfaceT (IEC 62769-4)	MANUFAC_ID String format "0xdddd" where dddd is the MANUFAC_ID number in hexadecimal format.
DeviceModel element of InterfaceT (IEC 62769-4)	DEVICE_TYPE String format "0xdddd" where dddd is the DEVICE_TYPE number in hexadecimal format.
DeviceRevision element ListOfSupportedDeviceRevisionsT (IEC 62769-4)	DEV_REV String format "x.0.0" where x is the DEV_REV in decimal format (no leading zeros).

4.3.2 Device type revision mapping

Each device type is identified according to 4.3.1. A device may also include a parameter COMPATIBILITY_REV from the Resource Block. This parameter specifies the lowest device version (DEV_REV) that a new device can replace while maintaining compatibility with a prior FDI Device Package.

4.4 Information Model mapping

4.4.1 ProtocolType definition

Table 4 defines the ProtocolType used to identify CP 1/1 network communications.

Table 4 – ProtocolType Foundation_H1 definition

Attribute	Value				
BrowseName	Foundation_H1				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Inherits the properties of ProtocolType defined in IEC 62541-100.					

4.4.2 DeviceType mapping

Each device type inherits the properties of the DeviceType. The mapping of the inherited properties from the DeviceType is defined in Table 5.

Table 5 – Inherited DeviceType Property mapping

Property	CP Mapping
SerialNumber	DEV_ID (System Management Information Base)
RevisionCounter	-1 (not defined)
Manufacturer	~~MANUFAC_ID (Resource Block)~~ String obtained from FDI package catalog (ManufacturerName from PackageT)
Model	~~DEV_TYPE (Resource Block)~~ String obtained from FDI package catalog (Name of DeviceTypeT, which is a localized name)
DeviceManual	entry text string (not supported) [a]
DeviceRevision	DEV_REV (Resource Block)
SoftwareRevision	SOFTWARE_REV (if available, otherwise ~~-1~~ empty string)
HardwareRevision	HARDWARE_REV (if available, otherwise ~~-1~~ empty string)
[a] Device manuals are exposed as attachments of the FDI Device Package.	

4.4.3 FunctionalGroup Identification definition

As defined in IEC 62541-100, each device representation in the FDI Server hosted Information Model shall contain a protocol specific FunctionalGroup called Identification. This FunctionalGroup organizes variables found in the Resource Block of the device type instance. The FunctionalGroup Identification for CP 1/1 is defined in Table 6.

Table 6 – Identification Parameters

BrowseName	DataType	Optional/Mandatory
MANUFAC_ID	UInt32	Mandatory
DEV_TYPE	UInt16	Mandatory
DEV_REV	UInt8	Mandatory
HARDWARE_REV	String	Optional
SOFTWARE_REV	String	Optional
COMPATIBILITY_REV	UInt8	Optional
CAPABILITY_LEV	UInt8	Optional
ITK_VER	UInt16	Mandatory
SIF_ITK_VER	UInt16	Optional
FD_VER	UInt16	Optional

4.4.4 BlockType property mapping

CP 1/1 device types are block-oriented according to IEC 62541-100. IEC 62769-5 specifies the mapping of EDDL BLOCK_A elements to block types and instances.

The BLOCK_A maps as a subtype of the topology element BlockType and inherits the properties per IEC 62541-100. The mapping of the inherited properties of the BlockType is specified in Table 7.

Table 7 – Inherited BlockType property mapping

Property	CP Mapping (Block ParameterSet)
RevisionCounter	ST_REV
ActualMode	MODE_BLK.ACTUAL
PermittedMode	MODE_BLK.PERMITTED
NormalMode	MODE_BLK.NORMAL
TargetMode	MODE_BLK.TARGET

4.4.5 Mapping to Block ParameterSet

The ParameterSet is relative to each Block. The ParameterSet includes the CHARACTERISTICS records of the block and all the parameters found in the PARAMETERS, LOCAL_PARAMETERS and LIST_ITEMS.

The browse name of the parameters found in the PARAMETERS and LOCAL_PARAMETERS is the member name in the respective lists. For example, ST_REV is the browse name of the Static Revision parameter. LIST_ITEMS do not have member names; therefore the browse name of each LIST in the LIST_ITEMS is the item name of the list.

4.5 Topology elements

4.5.1 ConnectionPoint definition

The ConnectionPoint type ConnectionPoint_Foundation_H1 shall be used to identify CP 1/1 network communication and is defined in Table 8. The ConnectionPoint_Foundation_H1 type is a sub type of the abstract type ConnectionPointType defined in IEC 62541-100.

The Address property shall be the H1 node address.

The OrdinalNumber property reflects the position of the VFD within the SMIB VFD list. For devices exposing multiple FB VFDs, the OrdinalNumber property is mandatory to address the FB VFD. For devices with a single FB VFD, the OrdinalNumber property can be omitted. Devices exposed as instances of type DeviceType define their connection points as components. Hence Devices with multiple FB VFDs shall contain multiple Connection Points, one per FB VFD.

The SIFConnection property denotes whether a safety instrumented function (SIF) connection is necessary or not according to the functional safety profile (IEC 61784-3:~~2010~~2016, Clause 6). CP 1/1 devices that implement the functional safety profile shall have a connection point as a component that has set this property to true. Devices supporting standard connections and SIF connections shall expose two Connections Points as components.

Table 8 – ConnectionPointType ConnectionPoint_Foundation_H1 definition

Attribute	Value				
BrowseName	ConnnectionPoint_Foundation_H1				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Inherits the properties of ConnectionPointType defined in IEC 62541-100.					
HasProperty	Variable	Address	Byte	PropertyType	Mandatory
HasProperty	Variable	OrdinalNumber	Int32	PropertyType	Optional
HasProperty	Variable	SIFConnection	Boolean	PropertyType	Optional

The ConnectionPoint type ConnectionPoint_Foundation_H1 shall be described by an EDD element contained in a Communication Device related FDI Package that can drive a CP 1/1 network. Actual ConnectionPoint_Foundation_H1 properties are declared by VARIABLE constructs grouped together in a COLLECTION named Foundation_H1_ConnectionPoint_Properties.

```
COMPONENT ConnectionPoint_Foundation_H1
{
  LABEL "Foundation H1 Connection point";
  CLASSIFICATION NETWORK_CONNECTION_POINT;
  CAN_DELETE FALSE;
  PROTOCOL Foundation_H1;
  CONNECTION_POINT Foundation_H1_ConnectionPoint_Properties;
}

VARIABLE Address
{
  LABEL "H1 Node address";
  HELP "Address of the H1 Node";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (1)
  {
    MIN_VALUE 16;
    MAX_VALUE 255;
  }
  HANDLING READ & WRITE;
}


VARIABLE OrdinalNumber
{
  LABEL "OrdinalNumber address property";
  HELP "OrdinalNumber property to address the Function Block
Application”;
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (4);
  HANDLING READ & WRITE;
}

VARIABLE SIFConnection
{
  LABEL "SIFConnection address property";
  HELP "Connection point supports SIF Connections";
  CLASS DEVICE;
  TYPE ENUMERATED (1)
  {
     {0,”NO_SIFCONNECTION”} ,
```

```
      {1,"SIFCONNECTION"}
  }
  HANDLING READ & WRITE;
}

COLLECTION Foundation_H1_ConnectionPoint_Properties
{
  LABEL "FF H1 Connection Point data";
  MEMBERS
  {
    CONNECTION_POINT_ADDRESS, Address;
    CONNECTION_POINT_ORDINALNUMBER, OrdinalNumber;
    CONNECTION_POINT_SIFCONNECTION , SIFConnection;
  }
}
```

4.5.2 Communication Device definition

According to IEC 62769-7, each FDI Communication Package shall contain an EDD element describing the device. The following EDDL source code is an example describing an FDI Communication Server.

```
COMPONENT Foundation_H1_Communication_Server
{
  LABEL "Foundation H1 communication server";
  PRODUCT_URI "urn:Fieldbus Foundation:Foundation H1 Communication
Server";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Communication_Device_Setup
  }
}

COMPONENT_RELATION Foundation_H1_Communication_Device_Setup
{
  LABEL "Relation between Device and communication device";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING { LinkId }
  COMPONENTS
  {
    Foundation_H1_Communication_Device{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 4;
}

VARIABLE LinkId
{
  LABEL "Link Id address parameter of the Communication device";
  HELP "Link Id address parameter of the Communication device";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (2)
  {
    MIN_VALUE 4096;
    MAX_VALUE 65535;
  }
  HANDLING READ & WRITE;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type CommunicationServerType as described in IEC 62769-7.

The LinkId VARIABLE holds the address value for a Communication device instance. In the Information model the LinkId will be represented as an instance of BaseDataVariableType and as a component of the ParameterSet of the communication device.

According to 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 is an example for a communication device.

```
COMPONENT Foundation_H1_Communication_Device
{
  LABEL "Foundation H1 communication device";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Service_Provider_Relation
  }
}

COMPONENT_RELATION Foundation_H1_Service_Provider_Relation
{
  LABEL "Foundation H1 communication service provider";
  RELATION_TYPE CHILD_COMPONENT;
  COMPONENTS
  {
    Foundation_H1_Service_Provider{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 16;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type ServerCommunicationDeviceType as described in IEC 62769-7.

An instance of ServerCommunicationDeviceType shall contain the following parameter(s) with its ParameterSet. Table 9 shows the definition of Communication device ParameterSet.

Table 9 – Communication device ParameterSet definition

Attribute	Value				
BrowseName	ParameterSet				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
See IEC 62541-100:2015, 5.2.					
HasTypeDefinition	ObjectType	BaseObjectType			
HasComponent	Variable	LinkId	UInt16	BaseDataVariableType	Mandatory
HasComponent	Variable	<ParameterIdentifier>		BaseDataVariableType	Mandatory-Placeholder

The LinkId parameter is an addressing parameter distinguishing multiple H1 Links. If an FDI Communication Server supports multiple physical H1 Links these are mapped within the Information Model to multiple communication device instances. If the FDI Communication Server supports only one H1 Link it shall define only one communication device within the Information Model. The value of the variable can be set to 0 in this case.

The EDD declaration of the variable LinkId is with the ADDRESSING attribute of the COMPONENT_RELATION of the FDI Communication Server definition ~~(see 5.5.2)~~.

4.5.3 Communication service provider definition

According to 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 is an example for a CP 1/1 communication service provider component.

The component reference ConnectionPoint_Foundation_H1 corresponds to the related Connection Point definition in 4.5.1.

```
COMPONENT Foundation_H1_Service_Provider
{
  LABEL "Foundation H1 communication service provider";
  CAN_DELETE FALSE;
  CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Connection_Point_Relation
  }
  BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION
Foundation_H1_Service_Provider_Connection_Point_Relation
{
  LABEL "Relation between communication service provider and
connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING {Address};
  COMPONENTS
  {
    ConnectionPoint_Foundation_H1{ AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 1;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type ServerCommunicationServiceType as described in IEC 62769-7.

4.5.4 Network definition

According to IEC 62769-7 each FDI Communication Package shall contain at least one EDD element describing one Network for each of the protocols that are supported by the Communication Device. The definition supports the network topology engineering.

```
COMPONENT Network_Foundation_H1
{
  LABEL "Foundation H1 Network";
  CAN_DELETE TRUE;
```

```
  CLASSIFICATION NETWORK;
  PROTOCOL Foundation_H1
  COMPONENT_RELATIONS
  {
    Foundation_H1_Network_Connection_Point_Relation
  }

COMPONENT_RELATION Foundation_H1_Network_Connection_Point_Relation
{
  LABEL "Relation between network and connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING {Address}
  COMPONENTS
  {
    ConnectionPoint_Foundation_H1
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 32;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type NetworkType as described in IEC 62541-100.

4.6 Methods

4.6.1 Methods for FDI Communication Servers

4.6.1.1 General

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

4.6.1.2 Connect

Table 10 shows the Method Connect arguments. The connect transaction may need a significant amount of time since configuration of communication endpoints is involved. Ensure the OPC UA timeout is configured appropriately (e.g. 30 seconds).

Signature:

```
Connect(
   [in]  ByteString       CommunicationRelationId,
   [in]  UInt16           LinkId
   [in]  byte             Address,
   [in]  Int32            OrdinalNumber
   [in]  Boolean          SIFConnection
   [in]  UInt32           ServiceID,
   [out] UInt32           DelayForNextCall,
   [out] Int32            ServiceError);
```

Table 10 – Method Connect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the Device ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
LinkId	The argument name shall match with the corresponding BrowseName of the Variable defined as a component of an instance of type ~~CommunicationServerChannelType~~ ServerCommunicationDeviceType (refer to ~~5.5.3~~ 4.5.2). The argument value is passed by the parent instance of a ~~CommunicationServerChannelType~~ ServerCommunicationDeviceType. The value may be obtained by the Scan Method or may be directly configured.
Address	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element. The argument value holds the device's node address.
OrdinalNumber	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in ~~5.4.2~~ 4.5.1. The argument value holds the OrdinalNumber. The OrdinalNumber is the position of the VFD within the SMIB VFD list. If a value 0 is passed with this argument the first FB VFD is selected.
SIFConnection	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in ~~5.4.2~~ 4.5.1. The argument value denotes whether a SIF Connection is necessary or not.
~~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 faster than specified in the argument value.~~
ServiceError	~~0: OK / function started asynchronously, result has to be polled with EndConnect~~ ~~1: OK / execution finished, connection established successfully~~ 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 node address -5: Connect Failed/invalid device identification -6: Connect Failed/invalid LinkId argument -7: Connect Failed/invalid OrdinalNumber argument
NOTE IEC 62769-7 defines the argument AddressData of the Connect Method as an array of Variant. The address arguments defined with the table are represented as entries of the Variant array in the order they are specified above. IEC 62769-7 defines the argument DeviceInformation as a protocol specific argument list in which the Connect Method stores the resulting data. The DeviceInformation argument is defined as an array of Variant. The DeviceInformation argument is not used.	

4.6.1.3 Disconnect

Table 11 shows the Method Disconnect arguments.

Signature:

```
Disconnect(
```

```
    [in]   ByteString        CommunicationRelationId,
    [out]  UInt32            ServiceError)
```

Table 11 – Method Disconnect arguments

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

4.6.1.4 Transfer

Table 12 shows the Method Transfer arguments.

Signature

```
Transfer(
    [in]   ByteString        CommunicationRelationId,
    [in]   String            OPERATION,
    [in]   String            BlockTag,
[in]       UInt32            INDEX,
[in]       UInt32            SUB_INDEX,
    [in]   Byte[]            WriteData,
    [in]   UInt32            ServiceId,
    [out]  Byte[]            ReadData,
    [in]   UInt32            ServiceId,
    [out]  UInt32            DelayForNextCall,
    [out]  Int32             ServiceError);
```

Table 12 – 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.
OPERATION	The argument value indicates the data transfer operation. The allowed values are "READ", "WRITE" and "VIEW_READ".
BlockTag	The argument denotes the Block tag of the block instance being addressed. The value can be obtained by the Method Scan.
INDEX	OPERATION indicates "READ" or "WRITE": The argument denotes the relative index of the block parameter being addressed. The relative index can be calculated by iterating the parameters of a block within the FDI Information Model. A value of 0 addresses the block header record (described by the CHARACTERISTICS attribute within the EDD). The first parameter is addressed with the INDEX 1. OPERATION indicates "VIEW_READ": The argument denotes the view identifier in a range from 1 to 4. For instance a value of 1 requests that View_1 shall be read. Multiple View_3 or View_4 objects are identified by the INDEX and the SUB_INDEX argument.

Argument	Description
SUB_INDEX	OPERATION indicates "READ" or "WRITE": The argument denotes the subindex of a member of the block parameter being addressed if the block parameter is of type RECORD or ARRAY. To address a simple parameter a value of 0 shall be passed with this argument. To address a specific member of a parameter of type RECORD or ARRAY a 1 relative value shall be passed with this argument. To address the parameter of type RECORD or ARRAY as a whole a value of 0 shall be passed with this argument. OPERATION indicates "VIEW_READ": The argument addresses a View_3 or View_4 if multiple views of that type exist. The argument shall be 0 if there are no multiple views of the type addressed with the INDEX argument. Values of 1 up to the number of views of that type address the specific view.
WriteData	Write data encoded as byte array. Encoding of integers shall follow the rules defined in IEC 62541-6. The argument shall be ignored if OPERATION indicates a read transfer or a view read transfer.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
ReadData	With this argument, the read data byte stream is returned as byte array. Encoding of integers shall follow the rules defined in IEC 62541-6. The argument shall be ignored if OPERATION indicates a write transfer.
~~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	~~0: OK / function started asynchronously, result has to be polled with EndTransfer~~ ~~1:OK / execution finished, ReceivedData contains the result~~ 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 -7: Transfer Failed/parameter Check [a] -8: Transfer Failed/exceeds Parameter Limits [a] -9: Transfer Failed/wrong Mode for Request [a] -10: Transfer Failed/write is prohibited by write lock switch or write lock Function Block for SIS devices [a] -11: Transfer Failed/data value is never writeable [a] -12: Transfer Failed/duplicate BlockTag detected -13: Invalid INDEX, SUB_INDEX argument provided with a "VIEW_READ" transfer
The FDI Server maintains an Information Model defined in IEC 62541-100. Hence topology elements representing an FFBlockType are separated from actual block instances. An instance called Blocks of a ConfigurableObjectType is used to implement instantiation rules. Instantiation of blocks is further detailed with IEC 62769-5. According to the rules defined in IEC 62769-5, the FDI Server needs to gather information of the FF Directory object in order to be able to create block instances. This information shall be provided by the Scan Method defined in 4.6.1.7. According to IEC 62769-5, the BlockTag argument denoted above is obtained from the DisplayName attribute of the corresponding Block instance within the FDI Information Model.	

Argument	Description
NOTE 1 IEC 62769-7 defines the argument SendData of the Transfer Method as an array of Variant. The arguments OPERATION, BlockTag, INDEX, SUB_INDEX and WriteData defined in this Table 12 are represented as entries of the Variant array in the order they are specified above. NOTE 2 IEC 62769-7 defines the argument ReceiveData of the Transfer Method as an array of Variant. The argument ReadData defined in this Table 12 is represented as an entry of the Variant array in the order specified above. NOTE 3 Example (for clarification): A block has two views of type View_4. The first view of type View_4 is addressed with the arguments INDEX = 4 and SUB_INDEX = 1. The second view is addressed with the arguments INDEX = 4 and SUB_INDEX = 2.	
[a] A ServiceError value may be returned with a write operation.	
See Annex C for a description on how the communication service arguments for the Transfer Method are obtained from the EDD.	

4.6.1.5 GetPublishedData

CP 1/1 alerts represent unsolicited messages as defined in IEC 62769-7. Table 13 shows the Method GetPublishedData arguments.

NOTE CP 1/1 uses the term "alerts" to refer to alarms and event messages. These are asynchronous, unsolicited messages that deliver state change notifications such as diagnostic conditions. These messages are mapped to the ~~GetPublishData~~ GetPublishedData service. CP 1/1 also uses the term "publish" to refer to synchronous, network scheduled communication for process values. These published messages are not mapped to the GetPublishedData service.

Signature:

```
GetPublishedData(
    [in] ByteString CommunicationRelationId,
    [out] String BlockTag,
    [out] Byte[] AlarmEventData,
    [out] NodeId AlarmEventType,
    [out] DateTime TimeStamp
    [in] UInt32 ServiceId,
    [out] UInt32 DelayForNextCall,
    [out] Int32 ServiceError);
```

Table 13 – Method GetPublishedData 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.
BlockTag	The output argument denotes the Block tag of the block instance that issued the alarm or event.
AlarmEventData	With this argument, the alarm/event data byte stream is returned as a byte array. Encoding of integers shall follow the rules defined in IEC 62541-6.
AlarmEventType	NodeId of the alarm or event type node defined within the FDI Information Model to decode the alarm/event data stream. The alarm and event types shall be read from the EDD by the FDI Server when creating the Information Model.
TimeStamp	Denotes the time the alarm or event was detected by the device.

~~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 EndGetPublishedData invocation cycle that shall not be faster than specified in the argument value.~~
ServiceError	~~0: OK / function started asynchronously, result has to be polled with EndGetPublishedData~~ ~~1: OK / execution finished, ReceivedData and TimeStamp contains the result~~ 0: OK/execution finished. -1: GetPublishedData Failed/canceled by caller -2: Call Failed/unknown service ID -3: GetPublishedData Failed/not supported -4: GetPublishedData Failed/no existing communication relation -5: GetPublishedData Failed/invalid communication relation identifier -8: GetPublishedData Failed/no alarm/event data published -9: GetPublishedData Failed/invalid AlarmEventType
The FDI Server maintains an Information Model defined in IEC 62541-100. Hence topology elements representing an FFBlockType are separated from actual block instances. An instance called Blocks of a ConfigurableObjectType is used to implement instantiation rules. Instantiation of blocks is further detailed with IEC 62769-5. According to the rules defined in IEC 62769-5, the FDI Server needs to gather information of the FF Directory object in order to be able to create block instances. This information shall be provided by the Scan Method defined in 4.6.1.7. According to IEC 62769-5, the BlockTag argument denoted above is obtained from the DisplayName attribute of the corresponding Block instance within the FDI Information Model. A ServiceError value may be returned with a write operation.	
NOTE 1 IEC 62769-7 defines the argument ReceiveData of the GetPublishedData Method as an array of Variant. The arguments BlockTag, AlarmEventData and AlarmEventType defined in this Table 13 are represented as entries of the Variant array in the order they are specified above. NOTE 2 IEC 62769-7 defines the argument SendData of the Transfer Method as an array of Variant. The arguments OPERATION, BlockTag, INDEX, SUB_INDEX and WriteData defined in the table are represented as entries of the Variant array in the order they are specified above. NOTE 3 IEC 62769-7 defines the argument ReceiveData of the Transfer Method as an array of Variant. The argument ReadData defined in the table is represented as an entry of the Variant array in the order specified above.	

4.6.1.6 SetAddress

Table 14 shows the Method SetAddress arguments.

NOTE Modifying the address of a device will have an impact on the communications of a distributed control system (DCS) if present. Setting the address of a device will take a significant amount of time.

Signature:

```
SetAddress(
    [in]   String           OPERATION,
    [in]   UInt16           LinkId,
    [in]   byte             OldAddress,
    [in]   byte             NewAddress,
    [in]   String           NewPDTag
    [in]   UInt32           ServiceId,
    [out]  UInt32           DelayForNextCall,
    [out]  Int32            ServiceError);
```

Table 14 – Method SetAddress arguments

Argument	Description
OPERATION [a]	The argument value indicates the type of addressing operation. The allowed values are "SETASSIGNMENT", "CLEARASSIGNMENT". Argument values given with the arguments below may be ignored depending on the value of the OPERATION argument.
LinkId [a]	The argument name shall match with the corresponding BrowseName of the Variable defined as a component of an instance of type ~~CommunicationServerChannelType~~ ServerCommunicationDeviceType (refer to ~~5.5.3~~ 4.5.2). The argument value is passed by the parent instance of a ~~CommunicationServerChannelType~~ ServerCommunicationDeviceType. The value may be obtained by the Scan Method or may be directly configured.
OldAddress [a]	The argument value holds the current address of a device. Allowed values are 16...255.
NewAddress [b]	The argument value holds the new address for a device. Allowed values are 0 and 16...247. The value is 0 if the service is not being used to change the H1 device's address. The argument value is ignored if the OPERATION argument value is "CLEARASSIGNMENT".
NewPDTag [b]	The argument value holds the new PD-Tag to set for the device. The argument value is ignored if the OPERATION argument value is "CLEARASSIGNMENT".
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	~~0: OK / function started asynchronously, result has to be polled with EndSetAddress~~ ~~1: OK / execution finished successfully~~ 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

[a] IEC 62769-7 defines the argument OldAddress of the SetAddress Method as an array of Variant. The arguments OPERATION, OldAddress and LinkId defined in this Table 14 are represented as entries of the Variant array in the order they are specified above.

[b] IEC 62769-7 defines the argument NewAddress of the SetAddress Method as an array of Variant. The arguments NewAddress and NewPDTag defined in the table are represented as entries of the Variant array in the order they are specified above.

4.6.1.7 Scan

The Method signature specified in IEC 62769-7 applies. The corresponding topologyScanResult schema is specified in Annex A.

4.6.1.8 ResetScan

The Method signature specified in IEC 62769-7 applies.

4.6.2 Methods for Gateways

Not supported in this document.

Annex A
(normative)

Topology scan schema

A.1 General

The topology scan result schema specified in Annex A describes the CP 1/1 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 FDI Server hosted Information Model.

A.2 FoundationH1AddressT

A simple type that defines the address structure for CP 1/1.

The XML schema for a FoundationH1AddressT type is:

```
<xsd:simpleType name="FoundationH1AddressT">
      <xsd:restriction base="xsd:unsignedByte">
            <xsd:minInclusive value="16"/>
            <xsd:maxInclusive value="255"/>
      </xsd:restriction>
</xsd:simpleType>
```

A.3 FoundationH1ConnectionPointT

A complex type that defines the Connection Point for CP 1/1.

The XML schema for a FoundationH1ConnectionPointT type is:

```
<xsd:complexType name="FoundationH1ConnectionPointT">
      <xsd:sequence>
            <xsd:element name="Identification"
type="ff:FoundationIdentificationT"/>
            <xsd:element name="BlockScanInstance"
type="ff:FoundationBlockIdentificationT" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="Address" type="ff:FoundationH1AddressT"
use="required"/>
      <xsd:attribute name="SIFConnection" type="xsd:boolean"
use="required"/>
      <xsd:attribute name="OrdinalNumber" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>
```

The attributes of a FoundationH1ConnectionPointT type are described in Table A.1.

Table A.1 – Attributes of FoundationH1ConnectionPointT

Attribute	Description
Address	The Attribute value holds the address of the network connected device.
SIFConnection	SIFConnection denotes whether a SIF Connection is necessary or not.
OrdinalNumber	The OrdinalNumber property reflects the position of the VFD within the System Management VFD list. Multiple VFDs are mapped to multiple ScanItem elements.

The elements of a FoundationH1ConnectionPointT type are described in Table A.2.

Table A.2 – Elements of FoundationH1ConnectionPointT

Element	Description
Identification	The element data holds the device type identification data. Compared to the Information Model (IEC 62769-5) 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.
BlockScanInstance	Block instance information of the scanned device VFD. Used to create Block instances within FDI Server IM. See IEC 62769-5.

A.4 FoundationH1NetworkT

A complex type that defines the network for CP 1/1.

The XML schema for a FoundationH1NetworkT type is:

```
<xsd:complexType name="FoundationH1NetworkT">
      <xsd:sequence>
            <xsd:element name="ConnectionPoint"
type="ff:FoundationH1ConnectionPointT" maxOccurs="unbounded"/>
      </xsd:sequence>
</xsd:complexType>
```

The elements of a FoundationH1NetworkT type are described in Table A.3.

Table A.3 – Elements of FoundationH1NetworkT

Element	Description
ConnectionPoint	CP 1/1 Connection Point.

A.5 Network

The root element that is used to return the scan result of a CP 1/1 network.

The XML schema for a Network element is:

```
<xsd:element name="Network" type="ff:FoundationH1NetworkT"/>
```

A.6 FoundationBlockIdentificationT

A complex type that defines the block instance information of the scanned device.

The XML schema for a FoundationBlockIdentificationT type is:

```
<xsd:complexType name="FoundationBlockIdentificationT">
      <xsd:attribute name="BlockTag" use="required"/>
      <xsd:attribute name="DDItem" use="required"/>
      <xsd:attribute name="DirectoryPosition" use="required"/>
</xsd:complexType>
```

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

Table A.4 – Attributes of FoundationBlockIdentificationT

Attribute	Description
BlockTag	The BlockTag attribute shall be mapped to the DisplayName of a block instance to be created within the FDI Server IM.
DDItem	This attribute is used to find the correct block type of a block instance to be created within the FDI Server IM. The block type is looked up within the SupportedTypes Folder in the Blocks component of a DeviceType.
DirectoryPosition	This attribute denotes the relative position of the block instance within the Directory object. The first block instance has a value of 0. See block instantiation rules in IEC 62769-5.

A.7 FoundationIdentificationT

A complex type that defines the content corresponds to the FunctionalGroup Identification.

The XML schema for a FoundationIdentificationT type is:

```
<xsd:complexType name="FoundationIdentificationT">
      <xsd:attribute name="MANUFAC_ID" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="DEV_TYPE" type="xsd:unsignedShort"
use="required"/>
      <xsd:attribute name="DEV_REV" type="xsd:unsignedShort"
use="required"/>
      <xsd:attribute name="ITK_VER" type="xsd:unsignedShort"
use="required"/>
      <xsd:attribute name="HARDWARE_REV" type="xsd:string"
use="optional"/>
      <xsd:attribute name="SOFTWARE_REV" type="xsd:string"
use="optional"/>
      <xsd:attribute name="COMPATIBILITY_REV" type="xsd:unsignedInt"
use="optional"/>
      <xsd:attribute name="CAPABILITY_LEV" type="xsd:unsignedByte"
use="optional"/>
      <xsd:attribute name="SIF_ITK_VER" type="xsd:unsignedShort"
use="optional"/>
      <xsd:attribute name="FD_VER" type="xsd:unsignedShort"
use="optional"/>
</xsd:complexType>
```

The attributes of a FoundationIdentificationT type are described in Table A.5.

Table A.5 – Attributes of FoundationIdentificationT

Attribute	Description
MANUFAC_ID	Manufacturer identification number.
DEV_TYPE	Manufacturer model number associated with the resource.
DEV_REV	Manufacturer revision number associated with the resource. Conditional: Shall be available if the device exposes a Function Block VFD.
ITK_VER	ITK Profile Number. Conditional: Shall be available if the device exposes a Function Block VFD.
HARDWARE_REV	Manufacturer hardware revision.
SOFTWARE_REV	Manufacturer software revision.
COMPATIBILITY_REV	This parameter is optionally used when replacing field devices. The correct usage of this parameter presumes the COMPATIBILITY_REV value of the replacing device should be equal to or lower than the DEV_REV value of the replaced device.
CAPABILITY_LEV	This parameter may be included in a device to indicate the capability level supported by a device.
SIF_ITK_VER	SIF ITK Profile Number
FD_VER	A parameter equal to the value of the major version of the Field Diagnostics specification that this device was designed for.

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 62769-6 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 receiveData

An element contains data that is returned through IDirectAccess function Transfer defined argument receiveData.

The XML schema for a receiveData element is:

```
<xsd:element name="receiveData">
      <xsd:complexType>
            <xsd:complexContent>
                  <xsd:extension base="ff:TransferResultDataT">
                        <xsd:sequence>
                              <xsd:element name="ResponseCode"
type="ff:ResponseCodeT" minOccurs="0"/>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
</xsd:element>
```

The elements of a receiveData element are described in Table B.1.

Table B.1 – Elements of receiveData

Element	Description
ResponseCode	Optional element that holds the return values for a negative service response

B.3 sendData

An element contains data that is submitted through the IDirectAccess function Transfer defined argument sendData.

The XML schema for a sendData element is:

```
<xsd:element name="sendData" type="ff:TransferSendDataT"/>
```

B.4 OperationT

A simple type that defines service operations.

The XML schema for an OperationT enumeration type is:

```
<xsd:simpleType name="OperationT">
      <xsd:restriction base="xsd:string">
            <xsd:enumeration value="READ"/>
            <xsd:enumeration value="WRITE"/>
            <xsd:enumeration value="GETOD"/>
      </xsd:restriction>
</xsd:simpleType>
```

The enumeration values of an OperationT enumeration type are described in Table B.2.

Table B.2 – Enumerations of OperationT

Enumeration	Description
READ	Read Service according to IEC 61158-5-9:~~2007~~2014, 6.3.5.3.2
WRITE	Write Service according to IEC 61158-5-9:~~2007~~2014, 6.3.5.3.3
GETOD	GetOD (long form) service according to IEC 61158-5-9:~~2007~~2014, 6.3.2.3.2

B.5 ResponseCodeT

A complex type that defines negative response error information.

The XML schema for a ResponseCodeT type is:

```
<xsd:complexType name="ResponseCodeT">
      <xsd:attribute name="ErrorClass" type="xsd:unsignedShort"
use="required"/>
      <xsd:attribute name="AdditionalCode" type="xsd:short"
use="optional"/>
      <xsd:attribute name="AdditionalDescription" type="xsd:string"
use="optional"/>
</xsd:complexType>
```

The attributes of a ResponseCodeT type are described in Table B.3.

Table B.3 – Attributes of ResponseCodeT

Attribute	Description
ErrorClass	Class of error reported by the negative service response
AdditionalCode	Optional reason code provided by the Function Block application
AdditionalDescription	Optional text description of the negative service response

B.6 TransferResultDataT

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

The XML schema for a TransferResultDataT type is:

```
<xsd:complexType name="TransferResultDataT">
      <xsd:attribute name="DATA" type="xsd:hexBinary" use="optional"/>
</xsd:complexType>
```

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

Table B.4 – Attributes of TransferResultDataT

Attribute	Description
DATA	Data received after the service operation. This attribute is required for the GETOD and READ service operations.

B.7 TransferSendDataT

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

The XML schema for a TransferSendDataT type is:

```
<xsd:complexType name="TransferSendDataT">
      <xsd:attribute name="OPERATION" type="ff:OperationT"
use="required"/>
      <xsd:attribute name="OD_INDEX" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="OD_SUB_INDEX" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="DATA" type="xsd:hexBinary" use="optional"/>
</xsd:complexType>
```

The attributes of a TransferSendDataT type are described in Table B.5.

Table B.5 – Attributes of TransferSendDataT

Attribute	Description
OPERATION	Data transfer operation
OD_INDEX	Object dictionary index relative to the Function Block application
OD_SUB_INDEX	To address a simple parameter a value of 0 shall be passed with this attribute. To address a specific member of a parameter of type RECORD or ARRAY ~~a~~, 1-relative value shall be passed with this attribute. To address the parameter of type RECORD or ARRAY as a whole, a value of 0 shall be passed with this attribute.
DATA	Mandatory attribute for data to be transferred in the service WRITE operation. This attribute is not used for the READ or GETOD service operation.

Annex C
(informative)

Communication service arguments for Transfer Method

IEC 62769-3 details that communication service arguments for the Transfer Method (see 4.6.1.4) are obtained from COMMAND elements associated to the VARIABLE element. For variable read or write access the FDI server shall obtain the related COMMAND description and obtain the communication service arguments for the Transfer Method from the attributes of the COMMAND description via name matching.

Since the CPF1 EDD profile does not provide a COMMAND EDD item this approach requires additional considerations. In order to keep the FDI Server as generic as possible, the following solution is proposed.

The COMMAND construct is introduced but only at a virtual level. This means there will be no means within the CPF1 EDD profile grammar to define a COMMAND item.

The COMMAND item will have the following attributes:

- INDEX
- SUB_INDEX

The COMMAND item shall be related to the PARAMETER of the block.

The COMMAND item representation will be created automatically (on the fly) for each block parameter by the FDI (EDD) engine when a block is loaded by the engine.

NOTE With today's EDD services, the creation of the COMMAND items for PARAMETERs can be accomplished during ddi_get_item() for the block when the list of parameters is created.

For parameters of the type RECORD or ARRAY COMMAND DD items shall be created for each member of the RECORD or ARRAY. This is to ease operation for the FDI Server.

When the EDD is loaded by the FDI Server the BlockTypes are created within the SupportedTypes Folder in the Blocks component of the DeviceType (see IEC 62769-5 and IEC 62541-100). For each parameter of the BlockType the COMMAND description is provided by the FDI (EDD) engine.

Block instances are created with the result of the Scan Method as described in 4.6.1.7. Block instances are created as (child) components of the Blocks component. According to IEC 62769-5, the DisplayName of the Block instance is the BlockTag.

For variable read or write access, the FDI Server shall obtain the communication service arguments INDEX and SUB_INDEX from the COMMAND description of the parameter via name matching. For the communication service argument BlockTag the DisplayName of the block instance shall be used.

Bibliography

IEC 62769-3~~:2015~~, *Field Device Integration (FDI) – Part 3: FDI Server*

NOTE IEC 62769-3 is technically identical to FDI-2023

FIELDBUS FOUNDATION. *FOUNDATION Specification Common File Format.* FF-103, Version 1.9. 4 June 2010.

FIELDBUS FOUNDATION. *FOUNDATION Specification Function Block Application Process – Part 1. FF-890*, Version 1.10. 2 Aug. 2012.

FIELDBUS FOUNDATION. *FOUNDATION Specification Function Block Application Process – Part 2. FF-891*, Version 1.10. 2 Aug. 2012.

FIELDBUS FOUNDATION. *FF-880: FOUNDATION Specification System Management.* FF-980, Version 1.6. 1 Dec. 2005.

_____

IEC 62769-101-1

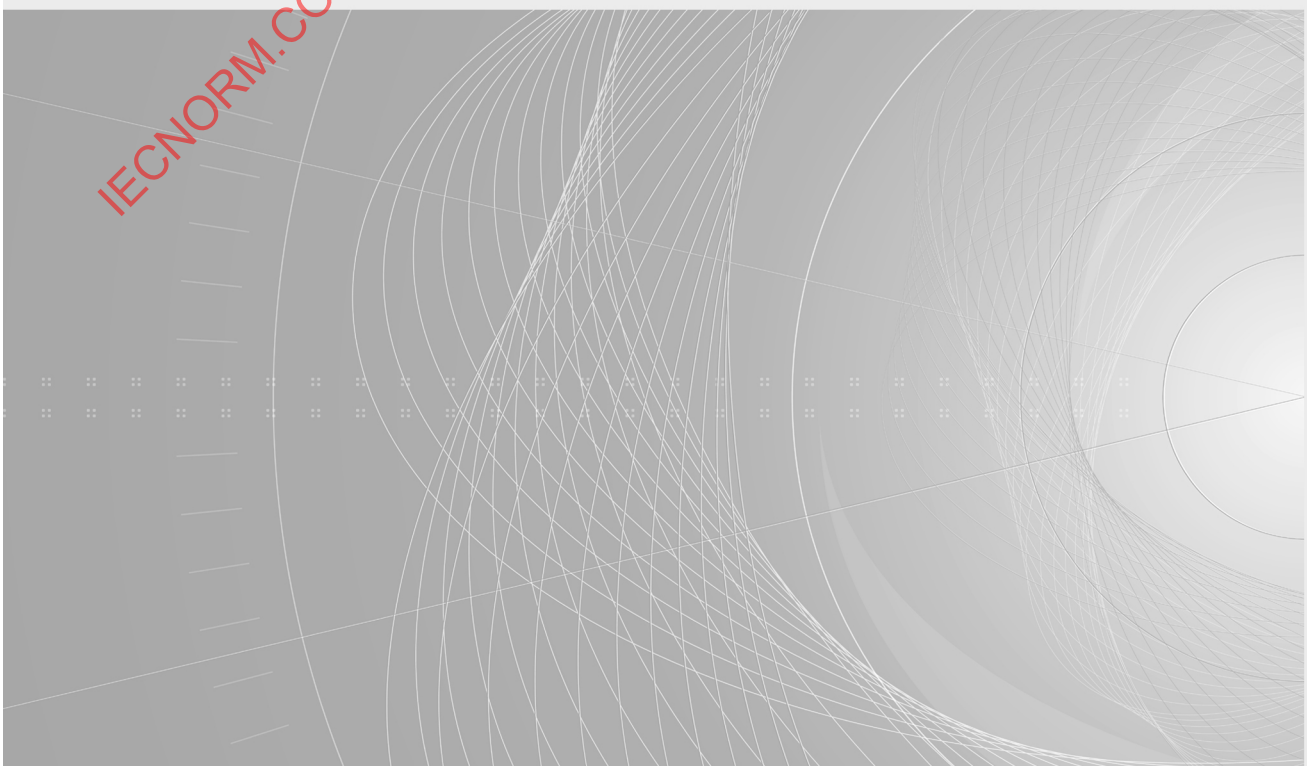
Edition 2.0 2020-11

INTERNATIONAL STANDARD

NORME INTERNATIONALE

Field device integration (FDI) – Part 101-1: Profiles – Foundation Fieldbus H1

Intégration des appareils de terrain (FDI) – Partie 101-1: Profils – Foundation Fieldbus H1



IEC 62769-101-1:2020-11(en-fr)

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

FIELD DEVICE INTEGRATION (FDI) –

Part 101-1: Profiles – Foundation Fieldbus H1

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 62769-101-1 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2015. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) support for generic protocol extension for faster adoption of other technologies;
b) support for Package developers to build EDDs targeted for today's EDD bases system under a single development tool.

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

CDV	Report on voting
65E/620/CDV	65E/683/RVC

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.

FIELD DEVICE INTEGRATION (FDI) –

Part 101-1: Profiles – Foundation Fieldbus H1

1 Scope

This part of IEC 62769 specifies the IEC 62769 profile for IEC 61784-1_CP 1/1 (FOUNDATION™ Fieldbus H1)[1].

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 61158-5-9:2014, *Industrial communication networks – Fieldbus specifications – Part 5-9: Application layer service definition – Type 9 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus Profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC/IEEE 8802-3*

IEC 61784-3:2016, *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions*

IEC 61804 (all parts), *Function blocks (FB) for process control and electronic device description language (EDDL)*

IEC 62541-6, *OPC unified architecture – Part 6: Mappings*

IEC 62541-100:2015, *OPC unified architecture – Part 100: Device Interface*

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

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

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

---

[1] FOUNDATION™ Fieldbus is the trade name of the non-profit consortium Fieldbus Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade name. Use of the trade name requires permission of the trade name holder.

3 Terms, definitions, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 61158-5-9, IEC 61784-1, IEC 61784-2, IEC 61784-3, IEC 61804 (all parts), IEC 62541-6, IEC 62541-100, IEC 62769-1, IEC 62769-2, IEC 62769-4, IEC 62769-5, IEC 62769-6, 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

3.2 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply:

CFF common file format
CP communication profile (see IEC 61784-1 or IEC 61784-2)
CPF communication profile family (see IEC 61784-1 or IEC 61784-2)
EDD Electronic Device Description (see IEC 61804 (all parts))
FB Function Block
IM Information Model
SMIB System Management Information Base
VFD virtual field device

3.3 Conventions

3.3.1 EDDL syntax

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

3.3.2 XML syntax

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

Example `<xsd:simpleType name="Example">`

3.3.3 Capitalizations

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

Some of these terms use an acronym 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 to 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:

- PROFILE_ID, or
- Profibus_PA_Network.

4 Profile for CP 1/1 (FOUNDATION™ H1)

4.1 General

This profile specifies the protocol specifics needed for FDI Packages describing Communication Servers, gateways and devices. Requirements for Direct Access transfer service parameters are given in Annex B.

4.2 Catalog profile

4.2.1 Protocol support file

Each CP 1/1 FDI Device Package shall contain a capability file. The capability file part is described in Table 1.

Table 1 – Capability File part

Parameter	Description
Content Type:	txt/plain
Root Namespace:	Not applicable
Source Relationship:	http://fdi-cooperation.com/2010/relationships/attachment-protocol
Filename:	Use file extension .CFF

4.2.2 CommunicationProfile definition

IEC 62769-4 defines a CommunicationProfileT string type for the Catalog XML schema. Table 2 defines the CP 1/1 specific values for this string.

Table 2 – CommunicationProfile definition

CommunicationProfile	Description
foundation_h1	CP 1/1 device type with a Function Block application

4.2.3 Profile device

Not supported in this document.

4.2.4 Protocol version information

IEC 62769-4 defines an element type named InterfaceT for the Catalog XML Schema. Element type InterfaceT contains an element named Version which is supposed to provide version information about the applied communication protocol profile. The value follows the IEC 62769-4 defined version information schema defined in element type VersionT.

The major version part of VersionT shall be set to the ITK_VER parameter. The minor and builds parts shall be set to 0.

EXAMPLE For ITK_VER 5, the value for InterfaceT is 5.0.0.

4.3 Associating a Package with a CP 1/1 device

4.3.1 Device type identification mapping

CP 1/1 device types are uniquely identified by the parameters MANUFAC_ID, DEVICE_TYPE and DEV_REV found in the Resource Block. These parameters are used to associate a given device instance to an FDI Device Package. These parameters are mapped to the FDI Device Package Catalog according to Table 3.

Table 3 – Device type catalog mapping

Catalog Element	CP Mapping
Manufacturer element of InterfaceT (IEC 62769-4)	MANUFAC_ID String format "0xdddd" where dddd is the MANUFAC_ID number in hexadecimal format.
DeviceModel element of InterfaceT (IEC 62769-4)	DEVICE_TYPE String format "0xdddd" where dddd is the DEVICE_TYPE number in hexadecimal format.
DeviceRevision element ListOfSupportedDeviceRevisionsT (IEC 62769-4)	DEV_REV String format "x.0.0" where x is the DEV_REV in decimal format (no leading zeros).

4.3.2 Device type revision mapping

Each device type is identified according to 4.3.1. A device may also include a parameter COMPATIBILITY_REV from the Resource Block. This parameter specifies the lowest device version (DEV_REV) that a new device can replace while maintaining compatibility with a prior FDI Device Package.

4.4 Information Model mapping

4.4.1 ProtocolType definition

Table 4 defines the ProtocolType used to identify CP 1/1 network communications.

Table 4 – ProtocolType Foundation_H1 definition

Attribute	Value				
BrowseName	Foundation_H1				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Inherits the properties of ProtocolType defined in IEC 62541-100.					

4.4.2 DeviceType mapping

Each device type inherits the properties of the DeviceType. The mapping of the inherited properties from the DeviceType is defined in Table 5.

Table 5 – Inherited DeviceType Property mapping

Property	CP Mapping
SerialNumber	DEV_ID (System Management Information Base)
RevisionCounter	-1 (not defined)
Manufacturer	String obtained from FDI package catalog (ManufacturerName from PackageT)
Model	String obtained from FDI package catalog (Name of DeviceTypeT, which is a localized name)
DeviceManual	entry text string (not supported) [a]
DeviceRevision	DEV_REV (Resource Block)
SoftwareRevision	SOFTWARE_REV (if available, otherwise empty string)
HardwareRevision	HARDWARE_REV (if available, otherwise empty string)
[a] Device manuals are exposed as attachments of the FDI Device Package.	

4.4.3 FunctionalGroup Identification definition

As defined in IEC 62541-100, each device representation in the FDI Server hosted Information Model shall contain a protocol specific FunctionalGroup called Identification. This FunctionalGroup organizes variables found in the Resource Block of the device type instance. The FunctionalGroup Identification for CP 1/1 is defined in Table 6.

Table 6 – Identification Parameters

BrowseName	DataType	Optional/Mandatory
MANUFAC_ID	UInt32	Mandatory
DEV_TYPE	UInt16	Mandatory
DEV_REV	UInt8	Mandatory
HARDWARE_REV	String	Optional
SOFTWARE_REV	String	Optional
COMPATIBILITY_REV	UInt8	Optional
CAPABILITY_LEV	UInt8	Optional
ITK_VER	UInt16	Mandatory
SIF_ITK_VER	UInt16	Optional
FD_VER	UInt16	Optional

4.4.4 BlockType property mapping

CP 1/1 device types are block-oriented according to IEC 62541-100. IEC 62769-5 specifies the mapping of EDDL BLOCK_A elements to block types and instances.

The BLOCK_A maps as a subtype of the topology element BlockType and inherits the properties per IEC 62541-100. The mapping of the inherited properties of the BlockType is specified in Table 7.

Table 7 – Inherited BlockType property mapping

Property	CP Mapping (Block ParameterSet)
RevisionCounter	ST_REV
ActualMode	MODE_BLK.ACTUAL
PermittedMode	MODE_BLK.PERMITTED
NormalMode	MODE_BLK.NORMAL
TargetMode	MODE_BLK.TARGET

4.4.5 Mapping to Block ParameterSet

The ParameterSet is relative to each Block. The ParameterSet includes the CHARACTERISTICS records of the block and all the parameters found in the PARAMETERS, LOCAL_PARAMETERS and LIST_ITEMS.

The browse name of the parameters found in the PARAMETERS and LOCAL_PARAMETERS is the member name in the respective lists. For example, ST_REV is the browse name of the Static Revision parameter. LIST_ITEMS do not have member names; therefore the browse name of each LIST in the LIST_ITEMS is the item name of the list.

4.5 Topology elements

4.5.1 ConnectionPoint definition

The ConnectionPoint type ConnectionPoint_Foundation_H1 shall be used to identify CP 1/1 network communication and is defined in Table 8. The ConnectionPoint_Foundation_H1 type is a sub type of the abstract type ConnectionPointType defined in IEC 62541-100.

The Address property shall be the H1 node address.

The OrdinalNumber property reflects the position of the VFD within the SMIB VFD list. For devices exposing multiple FB VFDs, the OrdinalNumber property is mandatory to address the FB VFD. For devices with a single FB VFD, the OrdinalNumber property can be omitted. Devices exposed as instances of type DeviceType define their connection points as components. Hence Devices with multiple FB VFDs shall contain multiple Connection Points, one per FB VFD.

The SIFConnection property denotes whether a safety instrumented function (SIF) connection is necessary or not according to the functional safety profile (IEC 61784-3:2016, Clause 6). CP 1/1 devices that implement the functional safety profile shall have a connection point as a component that has set this property to true. Devices supporting standard connections and SIF connections shall expose two Connections Points as components.

Table 8 – ConnectionPointType ConnectionPoint_Foundation_H1 definition

Attribute	Value				
BrowseName	ConnnectionPoint_Foundation_H1				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Inherits the properties of ConnectionPointType defined in IEC 62541-100.					
HasProperty	Variable	Address	Byte	PropertyType	Mandatory
HasProperty	Variable	OrdinalNumber	Int32	PropertyType	Optional
HasProperty	Variable	SIFConnection	Boolean	PropertyType	Optional

The ConnectionPoint type ConnectionPoint_Foundation_H1 shall be described by an EDD element contained in a Communication Device related FDI Package that can drive a CP 1/1 network. Actual ConnectionPoint_Foundation_H1 properties are declared by VARIABLE constructs grouped together in a COLLECTION named Foundation_H1_ConnectionPoint_Properties.

```
COMPONENT ConnectionPoint_Foundation_H1
{
  LABEL "Foundation H1 Connection point";
  CLASSIFICATION NETWORK_CONNECTION_POINT;
  CAN_DELETE FALSE;
  PROTOCOL Foundation_H1;
  CONNECTION_POINT Foundation_H1_ConnectionPoint_Properties;
}

VARIABLE Address
{
  LABEL "H1 Node address";
  HELP "Address of the H1 Node";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (1)
  {
    MIN_VALUE 16;
    MAX_VALUE 255;
  }
  HANDLING READ & WRITE;
}


VARIABLE OrdinalNumber
{
  LABEL "OrdinalNumber address property";
  HELP "OrdinalNumber property to address the Function Block
Application";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (4);
  HANDLING READ & WRITE;
}

VARIABLE SIFConnection
{
  LABEL "SIFConnection address property";
  HELP "Connection point supports SIF Connections";
  CLASS DEVICE;
  TYPE ENUMERATED (1)
  {
    {0,"NO_SIFCONNECTION"} ,
    {1,"SIFCONNECTION"}
  }
  HANDLING READ & WRITE;
}

COLLECTION Foundation_H1_ConnectionPoint_Properties
{
  LABEL "FF H1 Connection Point data";
  MEMBERS
  {
    CONNECTION_POINT_ADDRESS, Address;
    CONNECTION_POINT_ORDINALNUMBER, OrdinalNumber;
    CONNECTION_POINT_SIFCONNECTION , SIFConnection;
  }
}
```

4.5.2 Communication Device definition

According to IEC 62769-7, each FDI Communication Package shall contain an EDD element describing the device. The following EDDL source code is an example describing an FDI Communication Server.

```
COMPONENT Foundation_H1_Communication_Server
{
  LABEL "Foundation H1 communication server",
  PRODUCT_URI "urn:Fieldbus Foundation:Foundation H1 Communication
Server";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Communication_Device_Setup
  }
}

COMPONENT_RELATION Foundation_H1_Communication_Device_Setup
{
  LABEL "Relation between Device and communication device";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING { LinkId }
  COMPONENTS
  {
    Foundation_H1_Communication_Device{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 4;
}

VARIABLE LinkId
{
  LABEL "Link Id address parameter of the Communication device";
  HELP "Link Id address parameter of the Communication device";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (2)
  {
    MIN_VALUE 4096;
    MAX_VALUE 65535;
  }
  HANDLING READ & WRITE;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type CommunicationServerType as described in IEC 62769-7.

The LinkId `VARIABLE` holds the address value for a Communication device instance. In the Information model the LinkId will be represented as an instance of BaseDataVariableType and as a component of the ParameterSet of the communication device.

According to 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 is an example for a communication device.

```
COMPONENT Foundation_H1_Communication_Device
{
  LABEL "Foundation H1 communication device";
```

```
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Service_Provider_Relation
  }
}

COMPONENT_RELATION Foundation_H1_Service_Provider_Relation
{
  LABEL "Foundation H1 communication service provider”;
  RELATION_TYPE CHILD_COMPONENT;
  COMPONENTS
  {
    Foundation_H1_Service_Provider{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 16;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type ServerCommunicationDeviceType as described in IEC 62769-7.

An instance of ServerCommunicationDeviceType shall contain the following parameter(s) with its ParameterSet. Table 9 shows the definition of Communication device ParameterSet.

Table 9 – Communication device ParameterSet definition

Attribute	Value				
BrowseName	ParameterSet				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
See IEC 62541-100:2015, 5.2.					
HasTypeDefinition	ObjectType	BaseObjectType			
HasComponent	Variable	LinkId	UInt16	BaseDataVariableType	Mandatory
HasComponent	Variable	<ParameterIdentifier>		BaseDataVariableType	Mandatory-Placeholder

The LinkId parameter is an addressing parameter distinguishing multiple H1 Links. If an FDI Communication Server supports multiple physical H1 Links these are mapped within the Information Model to multiple communication device instances. If the FDI Communication Server supports only one H1 Link it shall define only one communication device within the Information Model. The value of the variable can be set to 0 in this case.

The EDD declaration of the variable LinkId is with the ADDRESSING attribute of the COMPONENT_RELATION of the FDI Communication Server definition.

4.5.3 Communication service provider definition

According to 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 is an example for a CP 1/1 communication service provider component.

The component reference ConnectionPoint_Foundation_H1 corresponds to the related Connection Point definition in 4.5.1.

```
COMPONENT Foundation_H1_Service_Provider
{
  LABEL "Foundation H1 communication service provider";
  CAN_DELETE FALSE;
  CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Connection_Point_Relation
  }
  BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION
Foundation_H1_Service_Provider_Connection_Point_Relation
{
  LABEL "Relation between communication service provider and
connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING {Address}
  COMPONENTS
  {
    ConnectionPoint_Foundation_H1{ AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 1;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type ServerCommunicationServiceType as described in IEC 62769-7.

4.5.4 Network definition

According to IEC 62769-7 each FDI Communication Package shall contain at least one EDD element describing one Network for each of the protocols that are supported by the Communication Device. The definition supports the network topology engineering.

```
COMPONENT Network_Foundation_H1
{
  LABEL "Foundation H1 Network";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK;
  PROTOCOL Foundation_H1
  COMPONENT_RELATIONS
  {
    Foundation_H1_Network_Connection_Point_Relation
  }

COMPONENT_RELATION Foundation_H1_Network_Connection_Point_Relation
{
  LABEL "Relation between network and connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING {Address}
  COMPONENTS
  {
    ConnectionPoint_Foundation_H1
  }
```

```
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 32;
}
```

Semantics of the EDDL constructs shown with the EDDL source code above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type NetworkType as described in IEC 62541-100.

4.6 Methods

4.6.1 Methods for FDI Communication Servers

4.6.1.1 General

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

4.6.1.2 Connect

Table 10 shows the Method Connect arguments. The connect transaction may need a significant amount of time since configuration of communication endpoints is involved. Ensure the OPC UA timeout is configured appropriately (e.g. 30 seconds).

Signature:

```
Connect(
    [in]   ByteString       CommunicationRelationId,
    [in]   UInt16           LinkId,
    [in]   byte             Address,
    [in]   Int32            OrdinalNumber
    [in]   Boolean          SIFConnection
    [out]  Int32            ServiceError);
```

Table 10 – Method Connect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the Device ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
LinkId	The argument name shall match with the corresponding BrowseName of the Variable defined as a component of an instance of type ServerCommunicationDeviceType (refer to 4.5.2). The argument value is passed by the parent instance of a ServerCommunicationDeviceType. The value may be obtained by the Scan Method or may be directly configured.
Address	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element. The argument value holds the device's node address.
OrdinalNumber	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 4.5.1. The argument value holds the OrdinalNumber. The OrdinalNumber is the position of the VFD within the SMIB VFD list. If a value 0 is passed with this argument the first FB VFD is selected.
SIFConnection	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 4.5.1. The argument value denotes whether a SIF Connection is necessary or not.

Argument	Description
ServiceError	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 node address -5: Connect Failed/invalid device identification -6: Connect Failed/invalid LinkId argument -7: Connect Failed/invalid OrdinalNumber argument
NOTE IEC 62769-7 defines the argument AddressData of the Connect Method as an array of Variant. The address arguments defined with the table are represented as entries of the Variant array in the order they are specified above. IEC 62769-7 defines the argument DeviceInformation as a protocol specific argument list in which the Connect Method stores the resulting data. The DeviceInformation argument is defined as an array of Variant. The DeviceInformation argument is not used.	

4.6.1.3 Disconnect

Table 11 shows the Method Disconnect arguments.

Signature:

```
Disconnect(
   [in]   ByteString        CommunicationRelationId,
   [out]  UInt32            ServiceError)
```

Table 11 – Method Disconnect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the Device ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI 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

Table 12 shows the Method Transfer arguments.

Signature

```
Transfer(
   [in]   ByteString        CommunicationRelationId,
   [in]   String            OPERATION,
   [in]   String            BlockTag,
[in]      UInt32            INDEX,
[in]      UInt32            SUB_INDEX,
   [in]   Byte[]            WriteData,
   [in]   UInt32            ServiceId,
   [out]  Byte[]            ReadData,
   [out]  Int32             ServiceError);
```

Table 12 – 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.
OPERATION	The argument value indicates the data transfer operation. The allowed values are "READ", "WRITE" and "VIEW_READ".
BlockTag	The argument denotes the Block tag of the block instance being addressed. The value can be obtained by the Method Scan.
INDEX	OPERATION indicates "READ" or "WRITE": The argument denotes the relative index of the block parameter being addressed. The relative index can be calculated by iterating the parameters of a block within the FDI Information Model. A value of 0 addresses the block header record (described by the CHARACTERISTICS attribute within the EDD). The first parameter is addressed with the INDEX 1. OPERATION indicates "VIEW_READ": The argument denotes the view identifier in a range from 1 to 4. For instance a value of 1 requests that View_1 shall be read. Multiple View_3 or View_4 objects are identified by the INDEX and the SUB_INDEX argument.
SUB_INDEX	OPERATION indicates "READ" or "WRITE": The argument denotes the subindex of a member of the block parameter being addressed if the block parameter is of type RECORD or ARRAY. To address a simple parameter a value of 0 shall be passed with this argument. To address a specific member of a parameter of type RECORD or ARRAY a 1 relative value shall be passed with this argument. To address the parameter of type RECORD or ARRAY as a whole a value of 0 shall be passed with this argument. OPERATION indicates "VIEW_READ": The argument addresses a View_3 or View_4 if multiple views of that type exist. The argument shall be 0 if there are no multiple views of the type addressed with the INDEX argument. Values of 1 up to the number of views of that type address the specific view.
WriteData	Write data encoded as byte array. Encoding of integers shall follow the rules defined in IEC 62541-6. The argument shall be ignored if OPERATION indicates a read transfer or a view read transfer.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
ReadData	With this argument, the read data byte stream is returned as byte array. Encoding of integers shall follow the rules defined in IEC 62541-6. The argument shall be ignored if OPERATION indicates a write transfer.
ServiceError	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 -7: Transfer Failed/parameter Check [a] -8: Transfer Failed/exceeds Parameter Limits [a] -9: Transfer Failed/wrong Mode for Request [a] -10: Transfer Failed/write is prohibited by write lock switch or write lock Function Block for SIS devices [a] -11: Transfer Failed/data value is never writeable [a] -12: Transfer Failed/duplicate BlockTag detected -13: Invalid INDEX, SUB_INDEX argument provided with a "VIEW_READ" transfer

Argument	Description
The FDI Server maintains an Information Model defined in IEC 62541-100. Hence topology elements representing an FFBlockType are separated from actual block instances. An instance called Blocks of a ConfigurableObjectType is used to implement instantiation rules. Instantiation of blocks is further detailed with IEC 62769-5. According to the rules defined in IEC 62769-5, the FDI Server needs to gather information of the FF Directory object in order to be able to create block instances. This information shall be provided by the Scan Method defined in 4.6.1.7. According to IEC 62769-5, the BlockTag argument denoted above is obtained from the DisplayName attribute of the corresponding Block instance within the FDI Information Model.	
NOTE 1 IEC 62769-7 defines the argument SendData of the Transfer Method as an array of Variant. The arguments OPERATION, BlockTag, INDEX, SUB_INDEX and WriteData defined in this Table 12 are represented as entries of the Variant array in the order they are specified above. NOTE 2 IEC 62769-7 defines the argument ReceiveData of the Transfer Method as an array of Variant. The argument ReadData defined in this Table 12 is represented as an entry of the Variant array in the order specified above. NOTE 3 Example (for clarification): A block has two views of type View_4. The first view of type View_4 is addressed with the arguments INDEX = 4 and SUB_INDEX = 1. The second view is addressed with the arguments INDEX = 4 and SUB_INDEX = 2.	
[a] A ServiceError value may be returned with a write operation.	
See Annex C for a description on how the communication service arguments for the Transfer Method are obtained from the EDD.	

4.6.1.5 GetPublishedData

CP 1/1 alerts represent unsolicited messages as defined in IEC 62769-7. Table 13 shows the Method GetPublishedData arguments.

NOTE CP 1/1 uses the term "alerts" to refer to alarms and event messages. These are asynchronous, unsolicited messages that deliver state change notifications such as diagnostic conditions. These messages are mapped to the GetPublishedData service. CP 1/1 also uses the term "publish" to refer to synchronous, network scheduled communication for process values. These published messages are not mapped to the GetPublishedData service.

Signature:

```
GetPublishedData(
    [in] ByteString CommunicationRelationId,
    [out] String BlockTag,
    [out] Byte[] AlarmEventData,
    [out] NodeId AlarmEventType,
    [out] DateTime TimeStamp,
    [out] Int32 ServiceError);
```

Table 13 – Method GetPublishedData 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.
BlockTag	The output argument denotes the Block tag of the block instance that issued the alarm or event.
AlarmEventData	With this argument, the alarm/event data byte stream is returned as a byte array. Encoding of integers shall follow the rules defined in IEC 62541-6.
AlarmEventType	NodeId of the alarm or event type node defined within the FDI Information Model to decode the alarm/event data stream. The alarm and event types shall be read from the EDD by the FDI Server when creating the Information Model.
TimeStamp	Denotes the time the alarm or event was detected by the device.

ServiceError	0: OK/execution finished. -1: GetPublishedData Failed/canceled by caller -2: Call Failed/unknown service ID -3: GetPublishedData Failed/not supported -4: GetPublishedData Failed/no existing communication relation -5: GetPublishedData Failed/invalid communication relation identifier -8: GetPublishedData Failed/no alarm/event data published -9: GetPublishedData Failed/invalid AlarmEventType
The FDI Server maintains an Information Model defined in IEC 62541-100. Hence topology elements representing an FFBlockType are separated from actual block instances. An instance called Blocks of a ConfigurableObjectType is used to implement instantiation rules. Instantiation of blocks is further detailed with IEC 62769-5. According to the rules defined in IEC 62769-5, the FDI Server needs to gather information of the FF Directory object in order to be able to create block instances. This information shall be provided by the Scan Method defined in 4.6.1.7. According to IEC 62769-5, the BlockTag argument denoted above is obtained from the DisplayName attribute of the corresponding Block instance within the FDI Information Model. A ServiceError value may be returned with a write operation.	
NOTE 1 IEC 62769-7 defines the argument ReceiveData of the GetPublishedData Method as an array of Variant. The arguments BlockTag, AlarmEventData and AlarmEventType defined in this Table 13 are represented as entries of the Variant array in the order they are specified above. NOTE 2 IEC 62769-7 defines the argument SendData of the Transfer Method as an array of Variant. The arguments OPERATION, BlockTag, INDEX, SUB_INDEX and WriteData defined in the table are represented as entries of the Variant array in the order they are specified above. NOTE 3 IEC 62769-7 defines the argument ReceiveData of the Transfer Method as an array of Variant. The argument ReadData defined in the table is represented as an entry of the Variant array in the order specified above.	

4.6.1.6 SetAddress

Table 14 shows the Method SetAddress arguments.

NOTE Modifying the address of a device will have an impact on the communications of a distributed control system (DCS) if present. Setting the address of a device will take a significant amount of time.

Signature:

```
SetAddress(
    [in]   String           OPERATION,
    [in]   UInt16           LinkId,
    [in]   byte             OldAddress,
    [in]   byte             NewAddress,
    [in]   String           NewPDTag
    [in]   UInt32           ServiceId,
    [out]  Int32            ServiceError);
```

Table 14 – Method SetAddress arguments

Argument	Description
OPERATION [a]	The argument value indicates the type of addressing operation. The allowed values are "SETASSIGNMENT", "CLEARASSIGNMENT". Argument values given with the arguments below may be ignored depending on the value of the OPERATION argument.
LinkId [a]	The argument name shall match with the corresponding BrowseName of the Variable defined as a component of an instance of type ServerCommunicationDeviceType (refer to 4.5.2). The argument value is passed by the parent instance of a ServerCommunicationDeviceType. The value may be obtained by the Scan Method or may be directly configured.
OldAddress [a]	The argument value holds the current address of a device. Allowed values are 16…255.
NewAddress [b]	The argument value holds the new address for a device. Allowed values are 0 and 16…247. The value is 0 if the service is not being used to change the H1 device's address. The argument value is ignored if the OPERATION argument value is "CLEARASSIGNMENT".
NewPDTag [b]	The argument value holds the new PD-Tag to set for the device. The argument value is ignored if the OPERATION argument value is "CLEARASSIGNMENT".
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response
ServiceError	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

[a] IEC 62769-7 defines the argument OldAddress of the SetAddress Method as an array of Variant. The arguments OPERATION, OldAddress and LinkId defined in this Table 14 are represented as entries of the Variant array in the order they are specified above.

[b] IEC 62769-7 defines the argument NewAddress of the SetAddress Method as an array of Variant. The arguments NewAddress and NewPDTag defined in the table are represented as entries of the Variant array in the order they are specified above.

4.6.1.7 Scan

The Method signature specified in IEC 62769-7 applies. The corresponding topologyScanResult schema is specified in Annex A.

4.6.1.8 ResetScan

The Method signature specified in IEC 62769-7 applies.

4.6.2 Methods for Gateways

Not supported in this document.

Annex A
(normative)

Topology scan schema

A.1 General

The topology scan result schema specified in Annex A describes the CP 1/1 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 FDI Server hosted Information Model.

A.2 FoundationH1AddressT

A simple type that defines the address structure for CP 1/1.

The XML schema for a FoundationH1AddressT type is:

```
<xsd:simpleType name="FoundationH1AddressT">
      <xsd:restriction base="xsd:unsignedByte">
            <xsd:minInclusive value="16"/>
            <xsd:maxInclusive value="255"/>
      </xsd:restriction>
</xsd:simpleType>
```

A.3 FoundationH1ConnectionPointT

A complex type that defines the Connection Point for CP 1/1.

The XML schema for a FoundationH1ConnectionPointT type is:

```
<xsd:complexType name="FoundationH1ConnectionPointT">
      <xsd:sequence>
            <xsd:element name="Identification"
type="ff:FoundationIdentificationT"/>
            <xsd:element name="BlockScanInstance"
type="ff:FoundationBlockIdentificationT" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="Address" type="ff:FoundationH1AddressT"
use="required"/>
      <xsd:attribute name="SIFConnection" type="xsd:boolean"
use="required"/>
      <xsd:attribute name="OrdinalNumber" type="xsd:unsignedInt"
use="required"/>
</xsd:complexType>
```

The attributes of a FoundationH1ConnectionPointT type are described in Table A.1.

Table A.1 – Attributes of FoundationH1ConnectionPointT

Attribute	Description
Address	The Attribute value holds the address of the network connected device.
SIFConnection	SIFConnection denotes whether a SIF Connection is necessary or not.
OrdinalNumber	The OrdinalNumber property reflects the position of the VFD within the System Management VFD list. Multiple VFDs are mapped to multiple ScanItem elements.

The elements of a FoundationH1ConnectionPointT type are described in Table A.2.

Table A.2 – Elements of FoundationH1ConnectionPointT

Element	Description
Identification	The element data holds the device type identification data. Compared to the Information Model (IEC 62769-5) 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.
BlockScanInstance	Block instance information of the scanned device VFD. Used to create Block instances within FDI Server IM. See IEC 62769-5.

A.4 FoundationH1NetworkT

A complex type that defines the network for CP 1/1.

The XML schema for a FoundationH1NetworkT type is:

```
<xsd:complexType name="FoundationH1NetworkT">
      <xsd:sequence>
            <xsd:element name="ConnectionPoint"
type="ff:FoundationH1ConnectionPointT" maxOccurs="unbounded"/>
      </xsd:sequence>
</xsd:complexType>
```

The elements of a FoundationH1NetworkT type are described in Table A.3.

Table A.3 – Elements of FoundationH1NetworkT

Element	Description
ConnectionPoint	CP 1/1 Connection Point.

A.5 Network

The root element that is used to return the scan result of a CP 1/1 network.

The XML schema for a Network element is:

```
<xsd:element name="Network" type="ff:FoundationH1NetworkT"/>
```

A.6 FoundationBlockIdentificationT

A complex type that defines the block instance information of the scanned device.

The XML schema for a FoundationBlockIdentificationT type is:

```
<xsd:complexType name="FoundationBlockIdentificationT">
      <xsd:attribute name="BlockTag" use="required"/>
      <xsd:attribute name="DDItem" use="required"/>
      <xsd:attribute name="DirectoryPosition" use="required"/>
</xsd:complexType>
```

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

Table A.4 – Attributes of FoundationBlockIdentificationT

Attribute	Description
BlockTag	The BlockTag attribute shall be mapped to the DisplayName of a block instance to be created within the FDI Server IM.
DDItem	This attribute is used to find the correct block type of a block instance to be created within the FDI Server IM. The block type is looked up within the SupportedTypes Folder in the Blocks component of a DeviceType.
DirectoryPosition	This attribute denotes the relative position of the block instance within the Directory object. The first block instance has a value of 0. See block instantiation rules in IEC 62769-5.

A.7 FoundationIdentificationT

A complex type that defines the content corresponds to the FunctionalGroup Identification.

The XML schema for a FoundationIdentificationT type is:

```
<xsd:complexType name="FoundationIdentificationT">
      <xsd:attribute name="MANUFAC_ID" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="DEV_TYPE" type="xsd:unsignedShort"
use="required"/>
      <xsd:attribute name="DEV_REV" type="xsd:unsignedShort"
use="optional"/>
      <xsd:attribute name="ITK_VER" type="xsd:unsignedShort"
use="optional"/>
      <xsd:attribute name="HARDWARE_REV" type="xsd:string"
use="optional"/>
      <xsd:attribute name="SOFTWARE_REV" type="xsd:string"
use="optional"/>
      <xsd:attribute name="COMPATIBILITY_REV" type="xsd:unsignedInt"
use="optional"/>
      <xsd:attribute name="CAPABILITY_LEV" type="xsd:unsignedByte"
use="optional"/>
      <xsd:attribute name="SIF_ITK_VER" type="xsd:unsignedShort"
use="optional"/>
      <xsd:attribute name="FD_VER" type="xsd:unsignedShort"
use="optional"/>
</xsd:complexType>
```

The attributes of a FoundationIdentificationT type are described in Table A.5.

Table A.5 – Attributes of FoundationIdentificationT

Attribute	Description
MANUFAC_ID	Manufacturer identification number.
DEV_TYPE	Manufacturer model number associated with the resource.
DEV_REV	Manufacturer revision number associated with the resource. Conditional: Shall be available if the device exposes a Function Block VFD.
ITK_VER	ITK Profile Number. Conditional: Shall be available if the device exposes a Function Block VFD.
HARDWARE_REV	Manufacturer hardware revision.
SOFTWARE_REV	Manufacturer software revision.
COMPATIBILITY_REV	This parameter is optionally used when replacing field devices. The correct usage of this parameter presumes the COMPATIBILITY_REV value of the replacing device should be equal to or lower than the DEV_REV value of the replaced device.
CAPABILITY_LEV	This parameter may be included in a device to indicate the capability level supported by a device.
SIF_ITK_VER	SIF ITK Profile Number
FD_VER	A parameter equal to the value of the major version of the Field Diagnostics specification that this device was designed for.

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 62769-6 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 receiveData

An element contains data that is returned through IDirectAccess function Transfer defined argument receiveData.

The XML schema for a receiveData element is:

```
<xsd:element name="receiveData">
      <xsd:complexType>
            <xsd:complexContent>
                  <xsd:extension base="ff:TransferResultDataT">
                        <xsd:sequence>
                              <xsd:element name="ResponseCode"
type="ff:ResponseCodeT" minOccurs="0"/>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
</xsd:element>
```

The elements of a receiveData element are described in Table B.1.

Table B.1 – Elements of receiveData

Element	Description
ResponseCode	Optional element that holds the return values for a negative service response

B.3 sendData

An element contains data that is submitted through the IDirectAccess function Transfer defined argument sendData.

The XML schema for a sendData element is:

```
<xsd:element name="sendData" type="ff:TransferSendDataT"/>
```

B.4 OperationT

A simple type that defines service operations.

The XML schema for an OperationT enumeration type is:

```
<xsd:simpleType name="OperationT">
      <xsd:restriction base="xsd:string">
            <xsd:enumeration value="READ"/>
            <xsd:enumeration value="WRITE"/>
            <xsd:enumeration value="GETOD"/>
      </xsd:restriction>
</xsd:simpleType>
```

The enumeration values of an OperationT enumeration type are described in Table B.2.

Table B.2 – Enumerations of OperationT

Enumeration	Description
READ	Read Service according to IEC 61158-5-9:2014, 6.3.5.3.2
WRITE	Write Service according to IEC 61158-5-9:2014, 6.3.5.3.3
GETOD	GetOD (long form) service according to IEC 61158-5-9:2014, 6.3.2.3.2

B.5 ResponseCodeT

A complex type that defines negative response error information.

The XML schema for a ResponseCodeT type is:

```
<xsd:complexType name="ResponseCodeT">
      <xsd:attribute name="ErrorClass" type="xsd:unsignedShort"
use="required"/>
      <xsd:attribute name="AdditionalCode" type="xsd:short"
use="optional"/>
      <xsd:attribute name="AdditionalDescription" type="xsd:string"
use="optional"/>
</xsd:complexType>
```

The attributes of a ResponseCodeT type are described in Table B.3.

Table B.3 – Attributes of ResponseCodeT

Attribute	Description
ErrorClass	Class of error reported by the negative service response
AdditionalCode	Optional reason code provided by the Function Block application
AdditionalDescription	Optional text description of the negative service response

B.6 TransferResultDataT

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

The XML schema for a TransferResultDataT type is:

```
<xsd:complexType name="TransferResultDataT">
      <xsd:attribute name="DATA" type="xsd:hexBinary" use="optional"/>
</xsd:complexType>
```

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

Table B.4 – Attributes of TransferResultDataT

Attribute	Description
DATA	Data received after the service operation. This attribute is required for the GETOD and READ service operations.

B.7 TransferSendDataT

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

The XML schema for a TransferSendDataT type is:

```
<xsd:complexType name="TransferSendDataT">
      <xsd:attribute name="OPERATION" type="ff:OperationT"
use="required"/>
      <xsd:attribute name="OD_INDEX" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="OD_SUB_INDEX" type="xsd:unsignedInt"
use="required"/>
      <xsd:attribute name="DATA" type="xsd:hexBinary" use="optional"/>
</xsd:complexType>
```

The attributes of a TransferSendDataT type are described in Table B.5.

Table B.5 – Attributes of TransferSendDataT

Attribute	Description
OPERATION	Data transfer operation
OD_INDEX	Object dictionary index relative to the Function Block application
OD_SUB_INDEX	To address a simple parameter a value of 0 shall be passed with this attribute. To address a specific member of a parameter of type RECORD or ARRAY, 1-relative value shall be passed with this attribute. To address the parameter of type RECORD or ARRAY as a whole, a value of 0 shall be passed with this attribute.
DATA	Mandatory attribute for data to be transferred in the service WRITE operation. This attribute is not used for the READ or GETOD service operation.

Annex C
(informative)

Communication service arguments for Transfer Method

IEC 62769-3 details that communication service arguments for the Transfer Method (see 4.6.1.4) are obtained from COMMAND elements associated to the VARIABLE element. For variable read or write access the FDI server shall obtain the related COMMAND description and obtain the communication service arguments for the Transfer Method from the attributes of the COMMAND description via name matching.

Since the CPF1 EDD profile does not provide a COMMAND EDD item this approach requires additional considerations. In order to keep the FDI Server as generic as possible, the following solution is proposed.

The COMMAND construct is introduced but only at a virtual level. This means there will be no means within the CPF1 EDD profile grammar to define a COMMAND item.

The COMMAND item will have the following attributes:

- INDEX
- SUB_INDEX

The COMMAND item shall be related to the PARAMETER of the block.

The COMMAND item representation will be created automatically (on the fly) for each block parameter by the FDI (EDD) engine when a block is loaded by the engine.

NOTE With today's EDD services, the creation of the COMMAND items for PARAMETERs can be accomplished during ddi_get_item() for the block when the list of parameters is created.

For parameters of the type RECORD or ARRAY COMMAND DD items shall be created for each member of the RECORD or ARRAY. This is to ease operation for the FDI Server.

When the EDD is loaded by the FDI Server the BlockTypes are created within the SupportedTypes Folder in the Blocks component of the DeviceType (see IEC 62769-5 and IEC 62541-100). For each parameter of the BlockType the COMMAND description is provided by the FDI (EDD) engine.

Block instances are created with the result of the Scan Method as described in 4.6.1.7. Block instances are created as (child) components of the Blocks component. According to IEC 62769-5, the DisplayName of the Block instance is the BlockTag.

For variable read or write access, the FDI Server shall obtain the communication service arguments INDEX and SUB_INDEX from the COMMAND description of the parameter via name matching. For the communication service argument BlockTag the DisplayName of the block instance shall be used.

Bibliography

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

NOTE IEC 62769-3 is technically identical to FDI-2023

FIELDBUS FOUNDATION. *FOUNDATION Specification Common File Format.* FF-103, Version 1.9. 4 June 2010.

FIELDBUS FOUNDATION. *FOUNDATION Specification Function Block Application Process – Part 1. FF-890*, Version 1.10. 2 Aug. 2012.

FIELDBUS FOUNDATION. *FOUNDATION Specification Function Block Application Process – Part 2. FF-891*, Version 1.10. 2 Aug. 2012.

FIELDBUS FOUNDATION. *FF-880: FOUNDATION Specification System Management.* FF-980, Version 1.6. 1 Dec. 2005.

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 101-1: Profils – Foundation Fieldbus H1

AVANT-PROPOS

1) La Commission Electrotechnique 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. A 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 62769-101-1 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.

Cette deuxième édition annule et remplace la première édition parue en 2015. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

a) prise en charge d'extensions de protocoles génériques, pour une adoption plus rapide d'autres technologies;
b) capacité offerte aux développeurs de Paquetages d'élaborer des EDD ciblant les systèmes actuels de bases EDD, en exploitant un seul outil de développement.

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

CDV	Rapport de vote
65E/620/CDV	65E/683/RVC

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. A cette date, le document sera

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

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 101-1: Profils – Foundation Fieldbus H1

1 Domaine d'application

La présente partie de l'IEC 62769 spécifie le profil de l'IEC 62769 pour le profil de communication CP 1/1 (FOUNDATION™ Fieldbus H1)[1] défini dans l'IEC 61784-1.

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 61158-5-9:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-9: Définition des services de la couche application – Eléments de type 9*

IEC 61784-1, *Réseaux de communication industriels – Profils – Partie 1: Profils de bus de terrain*

IEC 61784-2, *Réseaux de communication industriels – Profils – Partie 2: Profils de bus de terrain supplémentaires pour les réseaux en temps réel fondés sur l'ISO/IEC/IEEE 8802-3*

IEC 61784-3:2016, *Réseaux de communication industriels – Profils – Partie 3: Bus de terrain de sécurité fonctionnelle – Règles générales et définitions de profils*

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

IEC 62541-6, *Architecture unifiée OPC – Partie 6: Correspondances*

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

IEC 62769-1, *Intégration des appareils de terrain (FDI) – Partie 1: Vue d'ensemble*

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 62769-6, *Intégration des appareils de terrain (FDI) – Partie 6: Mapping de technologies FDI*

---

[1] FOUNDATION™ Fieldbus est l'appellation commerciale du consortium Fieldbus Foundation, une organisation à but non lucratif. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'organisation détentrice de l'appellation commerciale, ni un quelconque de ses produits. La conformité n'exige pas l'utilisation de l'appellation commerciale. L'utilisation de l'appellation commerciale exige l'autorisation du détenteur de l'appellation commerciale.

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

3 Termes, définitions, termes abrégés et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans IEC 61158-5-9, IEC 61784-1, IEC 61784-2, IEC 61784-3, IEC 61804 (toutes les parties), IEC 62541-6, IEC 62541-100, IEC 62769-1, IEC 62769-2, IEC 62769-4, IEC 62769-5, IEC 62769-6, et 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

3.2 Termes abrégés

Pour les besoins du présent document, les termes abrégés suivants s'appliquent:

CFF format de fichier commun (common file format)

CP profil de communication (communication profile, voir l'IEC 61784-1 ou l'IEC 61784-2)

CPF famille de profils de communication (communication profile family, voir l'IEC 61784-1 ou l'IEC 61784-2)

EDD Description d'Appareil Electronique (Electronic Device Description, voir l'IEC 61804 (toutes les parties))

FB Bloc Fonctionnel (Function Block)

IM Modèle d'Information (Information Model)

SMIB Base d'Informations de Gestion du Système (System Management Information Base)

VFD appareil de terrain virtuel (virtual field device)

3.3 Conventions

3.3.1 Syntaxe EDDL

Le présent document spécifie le contenu du composant EDD qui fait partie des Paquetages de Communication FDI. La syntaxe EDDL utilise la police Courier New. La syntaxe EDDL est utilisée pour les déclarations des signatures de méthodes, des variables, des structures de données et des composants.

3.3.2 Syntaxe XML

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

Exemple `<xsd:simpleType name="Example">`

3.3.3 Majuscules

La série IEC 62769 utilise des termes en majuscules pour souligner que ces termes ont une signification spécifique à la FDI.

Certains de ces termes utilisent un acronyme comme suffixe, par exemple:

- Client FDI, ou

- Serveur FDI.

Certains de ces termes sont des termes composés, par exemple:

- Serveurs de Communication, ou
- Paquetage de Profil.

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

- ProtocolSupportFile, ou
- ProtocolType.

Les noms de paramètres ou attributs peuvent aussi être construits en utilisant le caractère de soulignement pour concaténer au moins deux termes, par exemple:

- PROFILE_ID, ou
- Profibus_PA_Network.

4 Profil pour CP 1/1 (FOUNDATION™ H1)

4.1 Généralités

Ce profil spécifie les spécificités du protocole nécessaires aux Paquetages FDI décrivant des Serveurs de Communication, des passerelles et des appareils. Les exigences pour les paramètres du service Transfer des Services d'Accès Direct sont données dans l'Annexe B.

4.2 Profil de catalogue

4.2.1 Fichier de prise en charge de protocole

Chaque Paquetage d'Appareil FDI CP 1/1 doit contenir un fichier de capacité. La partie fichier de capacité est définie dans le Tableau 1.

Tableau 1 – Partie fichier de capacité

Paramètre	Description
Type de contenu:	texte/texte brut
Espace de noms racine:	Non applicable
Relation source:	http://fdi-cooperation.com/2010/relationships/attachment-protocol
Nom de fichier:	Utiliser l'extension de fichier .CFF

4.2.2 Définition de CommunicationProfile

L'IEC 62769-4 définit un type de chaîne CommunicationProfile pour le schéma XML Catalog. Le Tableau 2 définit les valeurs spécifiques au CP 1/1 pour cette chaîne.

Tableau 2 – Définition de CommunicationProfile

CommunicationProfile	Description
foundation_h1	Type d'appareil CP 1/1 avec une application de Bloc Fonctionnel

4.2.3 Appareil de profil

Non pris en charge dans le présent document.

4.2.4 Informations relatives à la version de protocole

L'IEC 62769-4 définit un type d'élément nommé InterfaceT pour le schéma XML Catalog. Le type d'élément InterfaceT contient un élément nommé Version qui est censé fournir des informations de version relatives au profil de protocole de communication appliqué. La valeur suit le schéma d'informations de version défini par l'IEC 62769-4, lequel schéma est défini dans le type d'élément VersionT.

La partie version majeure de VersionT doit être définie sur le paramètre ITK_VER. Les parties version mineure et numéro de build (mouture) doivent être mises à 0.

EXEMPLE Pour ITK_VER 5, la valeur pour InterfaceT est 5.0.0.

4.3 Association d'un Paquetage avec un appareil CP 1/1

4.3.1 Mapping d'identification du type d'appareil

Les types d'appareils CP 1/1 sont identifiés de manière unique par les paramètres MANUFAC_ID, DEVICE_TYPE et DEV_REV disponibles dans le Bloc de ressources. Ces paramètres sont utilisés pour associer une instance d'appareil donnée avec un Paquetage d'Appareil FDI. Ces paramètres sont mappés au Catalogue de Paquetage d'Appareil FDI conformément au Tableau 3.

Tableau 3 – Mapping dans le catalogue des types d'appareils

Elément du catalogue	Mapping des types d'appareils CP
Elément Manufacturer d'InterfaceT (IEC 62769-4)	MANUFAC_ID Format de chaîne "0xdddd", où dddd est le numéro MANUFAC_ID au format hexadécimal.
Elément DeviceModel d'InterfaceT (IEC 62769-4)	DEVICE_TYPE Format de chaîne "0xdddd" où dddd est le numéro DEVICE_TYPE au format hexadécimal.
Elément DeviceRevision de ListOfSupportedDeviceRevisionsT (IEC 62769-4)	DEV_REV Format de chaîne "x.0.0" où x est le numéro DEV_REV au format décimal (pas de zéros de tête).

4.3.2 Mapping des révisions de type d'appareil

Chaque type d'appareil est identifié conformément à 4.3.1. Un appareil peut également inclure un paramètre COMPATIBILITY_REV du Bloc de ressources. Ce paramètre spécifie la version d'appareil la plus ancienne (DEV_REV) qu'un nouvel appareil peut remplacer tout en maintenant la compatibilité avec un précédent Paquetage d'Appareil FDI.

4.4 Mapping du Modèle d'Information

4.4.1 Définition de ProtocolType

Le Tableau 4 définit le ProtocolType utilisé pour identifier des communications réseau CP 1/1.

Tableau 4 – Définition de ProtocolType Foundation_H1

Attribut	Valeur				
BrowseName	Foundation_H1				
IsAbstract	False				
Références	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Hérite des propriétés du ProtocolType définies dans l'IEC 62541-100.					

4.4.2 Mapping de DeviceType

Chaque type d'appareil hérite des propriétés du DeviceType. Le mapping des propriétés héritées du DeviceType est défini dans le Tableau 5.

Tableau 5 – Mapping des propriétés héritées du DeviceType

Propriété	Mapping des types d'appareils CP
SerialNumber	DEV_ID (Base d'Informations de Gestion du Système)
RevisionCounter	-1 (non défini)
Manufacturer	Chaîne obtenue à partir du catalogue de paquetages FDI (ManufacturerName issu de PackageT)
Model	Chaîne obtenue à partir du catalogue de paquetages FDI (élément Name de DeviceTypeT, qui est un nom localisé)
DeviceManual	chaîne textuelle d'entrée (non prise en charge) [a]
DeviceRevision	DEV_REV (Bloc de ressources)
SoftwareRevision	SOFTWARE_REV (si disponible, sinon chaîne vide)
HardwareRevision	HARDWARE_REV (si disponible, sinon chaîne vide)
[a] Les manuels d'appareils sont exposés comme pièces jointes du Paquetage d'Appareil FDI.	

4.4.3 Définition du FunctionalGroup «Identification»

Comme défini dans l'IEC 62541-100, chaque représentation d'appareil dans le Modèle d'Information hébergé sur le Serveur FDI doit contenir un FunctionalGroup spécifique à un protocole, appelé Identification. Ce FunctionalGroup organise les variables qui se trouvent dans le Bloc de ressources de l'instance de type d'appareil. Le FunctionalGroup Identification pour le CP 1/1 est défini dans le Tableau 6.

Tableau 6 – Paramètres du FunctionalGroup Identification

BrowseName	DataType	Facultatif/Obligatoire
MANUFAC_ID	UInt32	Obligatoire
DEV_TYPE	UInt16	Obligatoire
DEV_REV	UInt8	Obligatoire
HARDWARE_REV	String	Facultatif
SOFTWARE_REV	String	Facultatif
COMPATIBILITY_REV	UInt8	Facultatif
CAPABILITY_LEV	UInt8	Facultatif
ITK_VER	UInt16	Obligatoire
SIF_ITK_VER	UInt16	Facultatif
FD_VER	UInt16	Facultatif

4.4.4 Mapping des propriétés de BlockType

Les types d'appareils CP 1/1 sont orientés bloc conformément à l'IEC 62541-100. L'IEC 62769-5 spécifie le mapping des éléments EDDL BLOCK_A aux types et instances de blocs.

Le BLOCK_A est mappé comme un sous-type de l'élément de topologie BlockType et hérite des propriétés indiquées dans l'IEC 62541-100. Le mapping des propriétés héritées du BlockType est spécifié dans le Tableau 7.

Tableau 7 – Mapping des propriétés héritées du BlockType

Propriété	Mapping des types d'appareils CP (ParameterSet du Bloc)
RevisionCounter	ST_REV
ActualMode	MODE_BLK.ACTUAL
PermittedMode	MODE_BLK.PERMITTED
NormalMode	MODE_BLK.NORMAL
TargetMode	MODE_BLK.TARGET

4.4.5 Mapping sur le ParameterSet du Bloc

Le ParameterSet est relatif à chaque Bloc. Le ParameterSet inclut les enregistrements CHARACTERISTICS du bloc et tous les paramètres qui se trouvent dans PARAMETERS, LOCAL_PARAMETERS et LIST_ITEMS.

Le nom d'exploration des paramètres qui se trouve dans PARAMETERS et LOCAL_PARAMETERS est le nom du membre dans les listes respectives. Par exemple, ST_REV est le nom d'exploration du paramètre Static Revision. LIST_ITEMS n'a pas de noms de membres; par conséquent, le nom d'exploration de chaque LIST dans LIST_ITEMS est le nom d'article de la liste.

4.5 Eléments de topologie

4.5.1 Définition de ConnectionPoint

Le type de ConnectionPoint ConnectionPoint_Foundation_H1 doit être utilisé pour identifier la communication de réseau CP 1/1 et est défini dans le Tableau 8. Le type ConnectionPoint_Foundation_H1 est un sous-type du type abstrait ConnectionPointType défini dans l'IEC 62541-100.

La propriété Address doit être à l'adresse du nœud H1.

La propriété OrdinalNumber reflète la position du VFD dans la liste des VFD du SMIB. Pour les appareils qui exposent plusieurs FB VFD, la propriété OrdinalNumber est obligatoire pour adresser le FB VFD. Pour les appareils ayant un FB VFD unique, la propriété OrdinalNumber peut être omise. Les appareils exposés comme instances du type DeviceType définissent leurs points de connexion comme des composants. Par conséquent, les appareils ayant des FB VFD multiples doivent contenir des Points de connexion multiples, un par FB VFD.

La propriété SIFConnection indique si une connexion de fonction instrumentée de sécurité (SIF, Safety Instrumented Function) est nécessaire ou non, selon le profil de sécurité fonctionnelle (IEC 61784-3:2016, Article 6). Les appareils CP 1/1 qui mettent en œuvre le profil de sécurité fonctionnelle doivent disposer d'un point de connexion défini comme un composant, qui a mis cette propriété sur True. Les appareils qui prennent en charge les connexions normalisées et les connexions SIF doivent exposer deux Points de connexion comme des composants.

Tableau 8 – Définition du ConnectionPointType ConnectionPoint_Foundation_H1

Attribut	Valeur				
BrowseName	ConnnectionPoint_Foundation_H1				
IsAbstract	False				
Références	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Hérite des propriétés de ConnectionPointType définies dans l'IEC 62541-100.					
HasProperty	Variable	Address	Byte	PropertyType	Obligatoire
HasProperty	Variable	OrdinalNumber	Int32	PropertyType	Facultatif
HasProperty	Variable	SIFConnection	Boolean	PropertyType	Facultatif

Le type de ConnectionPoint ConnectionPoint_Foundation_H1 doit être décrit par un élément EDD contenu dans un Paquetage FDI associé à l'Appareil de Communication, en mesure de piloter un réseau CP 1/1. Les propriétés réelles de ConnectionPoint_Foundation_H1 sont déclarées par les constructions VARIABLE regroupées dans une COLLECTION nommée Foundation_H1_ConnectionPoint_Properties.

```
COMPONENT ConnectionPoint_Foundation_H1
{
  LABEL "Foundation H1 Connection point";
  CLASSIFICATION NETWORK_CONNECTION_POINT;
  CAN_DELETE FALSE;
  PROTOCOL Foundation_H1;
  CONNECTION_POINT Foundation_H1_ConnectionPoint_Properties;
}

VARIABLE Address
{
  LABEL "H1 Node address";
  HELP "Address of the H1 Node";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (1)
  {
    MIN_VALUE 16;
    MAX_VALUE 255;
  }
  HANDLING READ & WRITE;
}


VARIABLE OrdinalNumber
{
  LABEL "OrdinalNumber address property";
  HELP "OrdinalNumber property to address the Function Block
Application”;
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (4);
  HANDLING READ & WRITE;
}

VARIABLE SIFConnection
{
  LABEL "SIFConnection address property";
  HELP "Connection point supports SIF Connections";
  CLASS DEVICE;
  TYPE ENUMERATED (1)
  {
     {0,”NO_SIFCONNECTION”} ,
```

```
      {1,"SIFCONNECTION"}
  }
  HANDLING READ & WRITE;
}

COLLECTION Foundation_H1_ConnectionPoint_Properties
{
  LABEL "FF H1 Connection Point data";
  MEMBERS
  {
    CONNECTION_POINT_ADDRESS, Address;
    CONNECTION_POINT_ORDINALNUMBER, OrdinalNumber;
    CONNECTION_POINT_SIFCONNECTION , SIFConnection;
  }
}
```

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. Le code source EDDL ci-après est un exemple décrivant un Serveur de Communication FDI.

```
COMPONENT Foundation_H1_Communication_Server
{
  LABEL "Foundation H1 communication server";
  PRODUCT_URI "urn:Fieldbus Foundation:Foundation H1 Communication
Server";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Communication_Device_Setup
  }
}

COMPONENT_RELATION Foundation_H1_Communication_Device_Setup
{
  LABEL "Relation between Device and communication device";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING { LinkId }
  COMPONENTS
  {
    Foundation_H1_Communication_Device{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 4;
}

VARIABLE LinkId
{
  LABEL "Link Id address parameter of the Communication device";
  HELP "Link Id address parameter of the Communication device";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER (2)
  {
    MIN_VALUE 4096;
    MAX_VALUE 65535;
  }
  HANDLING READ & WRITE;
}
```

La sémantique des constructions EDDL représentées dans le code source EDDL ci-dessus est décrite dans l'IEC 62769-7. L'EDDL COMPONENT est utilisé par le Serveur FDI et le Serveur de Communication FDI pour créer une instance du type CommunicationServerType décrit dans l'IEC 62769-7.

Le LinkId VARIABLE contient la valeur d'adresse d'une instance d'Appareil de Communication. Dans le Modèle d'Information, le LinkId est représenté comme une instance de BaseDataVariableType et comme un composant du ParameterSet de l'Appareil 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 de l'Appareil de Communication. Le code source EDDL ci-après est un exemple décrivant un appareil de communication.

```
COMPONENT Foundation_H1_Communication_Device
{
  LABEL "Foundation H1 communication device";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS
  {
    Foundation_H1_Service_Provider_Relation
  }
}

COMPONENT_RELATION Foundation_H1_Service_Provider_Relation
{
  LABEL "Foundation H1 communication service provider";
  RELATION_TYPE CHILD_COMPONENT;
  COMPONENTS
  {
    Foundation_H1_Service_Provider{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 16;
}
```

La sémantique des constructions EDDL représentées dans le code source EDDL ci-dessus est décrite dans l'IEC 62769-7. L'EDDL COMPONENT est utilisé par le Serveur FDI et le Serveur de Communication FDI pour créer une instance du type ServerCommunicationDeviceType décrit dans l'IEC 62769-7.

Une instance de ServerCommunicationDeviceType doit contenir le ou les paramètres suivants avec son ParameterSet. Le Tableau 9 donne la définition du ParameterSet de l'Appareil de Communication.

Tableau 9 – Définition du ParameterSet de l'Appareil de Communication

Attribut	**Valeur**				
BrowseName	ParameterSet				
Références	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Voir l'IEC 62541-100:2015, 5.2.					
HasTypeDefinition	ObjectType	BaseObjectType			
HasComponent	Variable	LinkId	UInt16	BaseDataVariableType	Obligatoire
HasComponent	Variable	<ParameterIdentifier>		BaseDataVariableType	Obligatoire-Espace réservé