



IEC 62769-103-1

Edition 2.0 2020-04
REDLINE VERSION

INTERNATIONAL STANDARD



**Field device integration (FDI) –
Part 103-1: Profiles – PROFIBUS**





THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2020 IEC, Geneva, Switzerland

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

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

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

About the IEC

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

About IEC publications

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

IEC publications search - webstore.iec.ch/advsearchform

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

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

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

Electropedia - www.electropedia.org

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

IEC Glossary - std.iec.ch/glossary

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



IEC 62769-103-1

Edition 2.0 2020-04
REDLINE VERSION

INTERNATIONAL STANDARD



**Field device integration (FDI) –
Part 103-1: Profiles – PROFIBUS**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-8290-8

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD	4
INTRODUCTION	4
1 Scope	7
2 Normative references	7
3 Terms, definitions, abbreviated terms and acronyms conventions	8
3.1 Terms and definitions	8
3.2 Abbreviated terms and acronyms	8
3.3 Conventions	8
3.3.1 EDDL syntax	8
3.3.2 XML syntax	8
3.3.3 Capitalizations	8
4 Profile for PROFIBUS	9
4.1 General	9
4.2 Catalog profile	9
4.2.1 Protocol support file	9
4.2.2 CommunicationProfile definition	10
4.2.3 Profile device	10
4.2.4 Protocol version information	10
4.3 Associating a Package with a device	11
4.3.1 Device type identification mapping	11
4.3.2 Device type revision mapping	12
4.4 Information Model mapping	13
4.4.1 ProtocolType definition	13
4.4.2 DeviceType mapping	14
4.4.3 FunctionalGroup identification definition	14
4.5 Topology elements	15
4.5.1 ConnectionPoint definition	15
4.5.2 Communication Device definition	17
4.5.3 Communication service provider definition	17
4.5.4 Network definition	18
4.6 Methods	19
4.6.1 Methods for FDI Communication Servers	19
4.6.2 Methods for Gateways	21
Annex A (normative) Topology Scan result schema	27
A.1 General	27
A.2 Network	27
A.3 ProfibusNetworkT	27
A.4 ProfibusConnectionPointT	27
A.5 ProfibusIdentificationT	28
A.6 ProfibusAddressT	29
Annex B (normative) Transfer service parameters	30
B.1 General	30
B.2 sendData	30
B.3 xsreceiveData	30
B.4 xsTransferSendDataT	30
B.5 TransferResultDataT	31

B.6 OperationT	31
Bibliography	32

Figure 1 – Version mapping problem	12
--	----

Table 1 – ProtocolSupportFile for FDI Device Packages	9
Table 2 – ProtocolSupportFile for FDI Communication Packages	9
Table 3 – PROFIBUS CommunicationProfile definition schema	10
Table 4 – Catalog values for profile devices	10
Table 5 – Version mapping examples	11
Table 6 – Device identification information mapping	12
Table 7 – Protocol type Profibus_DP	13
Table 8 – Protocol type Profibus_PA	14
Table 9 – DeviceType property mapping	14
Table 10 – PROFIBUS Device Types identification attributes	14
Table 11 – ConnectionPoint type for Profibus_DP	15
Table 12 – ConnectionPoint type for Profibus_PA	16
Table 13 – Method Connect arguments	19
Table 14 – Method Disconnect arguments	20
Table 15 – Method Transfer arguments	20
Table 16 – Method SetAddress arguments	21
Table 17 – Connect service arguments	22
Table 18 – Method Transfer arguments	24
Table 19 – Method SetAddress arguments	25
Table A.1 – Elements of ProfibusNetworkT	27
Table A.2 – Attributes of ProfibusConnectionPointT	28
Table A.3 – Elements of ProfibusConnectionPointT	28
Table A.4 – Attributes of ProfibusIdentificationT	29
Table B.1 – Attributes of TransferSendDataT	31
Table B.2 – Attributes of TransferResultDataT	31

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE INTEGRATION (FDI) –

Part 103-1: Profiles – PROFIBUS

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 62769-103-1 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/622/CDV	65E/685A/RVC 65E/685/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 Ryttoft
Affolterstrasse 4
Zurich, 8050
Switzerland~~
- ~~b) Phoenix Contact GmbH & Co KG
Intellectual Property, Licenses & Standards
Flachsmarktstrasse 8, 32825 Blomberg
Germany~~
- ~~c) Fisher Controls International LLC
John Dilger, Emerson Process Management LLLP
301 S. 1st Avenue, Marshalltown, 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 103-1: Profiles – PROFIBUS

1 Scope

This part of IEC 62769 specifies an FDI profile of IEC 62769 for IEC 61784-1_CP 3/1 (PROFIBUS DP)¹ and IEC 61784-1_CP3/2 (PROFIBUS PA)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 61784-1, Industrial communication networks – Profiles – Part 1: Fieldbus profiles~~

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

IEC 62541-100:2015, *OPC Unified Architecture - Part 100: Device Interface*

IEC 62769-2², *Field Device Integration (FDI) - Part 2: FDI Client*

~~NOTE 1 – IEC 62769-2 is technical identical to FDI-2022.~~

IEC 62769-4³, *Field Device Integration (FDI) – Part 4: FDI Packages*

~~NOTE 2 – IEC 62769-4 is technically identical to FDI-2024.~~

IEC 62769-5⁴, *Field Device Integration (FDI) – Part 5: FDI Information Model*

~~NOTE 3 – IEC 62769-5 is technically identical to FDI-2025.~~

IEC 62769-7⁵, *Field Device Integration (FDI) – Part 7: FDI Communication Devices*

~~NOTE 4 – IEC 62769-7 is technically identical to FDI-2027.~~

PI Order No.: 2.122:2008, *Specification for PROFIBUS – Device Description and Device Integration – Volume 1: GSD, V5.1, July 2008: GSD*; available at <www.profibus.com> [viewed 2018-11-23]

¹ PROFIBUS is the trade name of the non-profit consortium PROFIBUS & PROFINET International. 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.

² Under preparation. Stage at the time of publication: IEC/RFDIS 62769-2:2020.

³ Under preparation. Stage at the time of publication: IEC/RFDIS 62769-4:2020.

⁴ Under preparation. Stage at the time of publication: IEC/RFDIS 62769-5:2020.

⁵ Under preparation. Stage at the time of publication: IEC/RFDIS 62769-7:2020.

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 61784-1, IEC 61804 (all parts), IEC 62541-100, IEC 62769-4, IEC 62769-5, IEC 62769-7 and PI Order No.: 2.122:2008 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 abbreviations apply:

EDD Electronic Device Description
EDDL Electronic Device Description Language (see IEC 61804)
GSD General station description (see PI Order No.: 2.122:2008)
I&M Identification and maintenance function
~~UIP — User Interface Plug-in~~
UUID Universally unique identifier (see ISO/IEC 11578)
XML Extensible markup language (see REC-xml-20081126)

3.3 Conventions

3.3.1 EDDL syntax

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

3.3.2 XML syntax

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

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

3.3.3 Capitalizations

IEC 62769 (all parts) use capitalized terms to emphasize that these terms have a FDI specific meaning.

Some of these terms using 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 PROFIBUS

4.1 General

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

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

4.2 Catalog profile

4.2.1 Protocol support file

4.2.1.1 FDI Device Package

Protocol specific attachments are mentioned in the Package Catalog as defined in IEC 62769-5. A communication feature list (GSD) file according to PI Order No.: 2.122:2008 is a mandatory attachment for FDI Device Packages representing PROFIBUS DP and PROFIBUS PA devices. Table 1 specifies the parameters of the ProtocolSupportFile in the FDI Device Package.

Table 1 – ProtocolSupportFile for FDI Device Packages

Parameter	Description
Content Type	text/plain
Root Namespace	empty
Source Relationship	http://fdi-cooperation.com/2010/relationship/attachment-protocol
Filename	According to PI Order No.: 2.122:2008

4.2.1.2 FDI Communication Packages

A GSD file as specified in PI Order No.: 2.122:2008 is an optional attachment for FDI Communication Packages representing PROFIBUS DP and PROFIBUS PA devices. Table 2 specifies the parameters of ProtocolSupportFile for FDI Communication Packages.

Table 2 – ProtocolSupportFile for FDI Communication Packages

Parameter	Description
Content Type:	text/plain
Root Namespace:	empty
Source Relationship:	http://fdi-cooperation.com/2010/relationship/attachment-protocol
Filename:	According to PI Order No.: 2.122:2008

4.2.2 CommunicationProfile definition

IEC 62769-4 defines a CommunicationProfileT-~~enumeration-type~~ string for the Catalog XML schema. Table 3 defines the PROFIBUS specific values for this-~~enumeration~~ string.

Table 3 – PROFIBUS CommunicationProfile definition schema

Profile Identifier	Protocol
"profibus_dp"	PROFIBUS DP/V0; PROFIBUS DP/V1; PROFIBUS DP/V2
"profibus_pa"	PROFIBUS PA

4.2.3 Profile device

A Profile Package shall provide the catalog values for profile devices, enabling the FDI Server to leverage a generic device description, if a specific one is not available. The definitions in Table 4 focus on catalog content that is vendor independent.

Table 4 – Catalog values for profile devices

Element	Attribute	Content
PackageType	—	Profile
CommunicationProfile Manufacturer	—	Empty
DeviceModel	—	<p>The allowed profile identifier values (PROFILE_ID) are provided by PROFIBUS & PROFINET International (PI). PI provides and maintains an XML file (Profile_ID_Table) containing the assignment of PROFILE_ID to profiles.</p> <p>It is available at <http://www.profibus.com/IM/Profile_ID_Table.xml></p> <p>The file can be downloaded by any engineering or service tool whenever it is connected to the Internet.</p> <p>NOTE More information is provided in PI Order No.: 3.502 (I&M Profile) and related profile definitions are referred therein.</p> <p>The string format shall be hexadecimal starting with 0x, e.g. '0x3D00'.</p>

4.2.4 Protocol version information

IEC 62769-4 defines an element type named InterfaceT for the Catalog XML schema. The element type InterfaceT contains an element named Version which is supposed to provide version information about the applied communication protocol profile. The value has to follow the IEC 62769-4 defined version information schema defined in the element type VersionT. Table 5 describes how to apply the currently known protocol versions defined by the non-profit consortium PROFIBUS & PROFINET International. The general rule is to apply the value "0" for parts of the version information according to IEC 62769-4 that are not used in currently known protocol versions.

Table 5 – Version mapping examples⁶

Protocol / Version	InterfaceT Version value
PROFIBUS DP/V0	0.0.0 ^a
PROFIBUS DP/V1	1.0.0 ^a
PROFIBUS DP/V2	2.0.0 ^a
PROFIBUS PA 3.02	3.2.0 ^b
^a The protocols PROFIBUS DP/V0, PROFIBUS DP/V1 and PROFIBUS DP/V2 contain a single number. This number is considered to be the major version. The minor and built numbers are set to "0".	
^b The currently known PROFIBUS PA profile number is considered to provide major and minor version information. Leading zeros are not considered in version value evaluation since only the actual decimal values are relevant.	

4.3 Associating a Package with a device

4.3.1 Device type identification mapping

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

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

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

Common for both ways of presenting the scan result is that scan results contain device type identification and device instance identification.

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

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

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

As a result of the FDI Package deployment the FDI Package information is then present in the Information Model as the specified FunctionalGroup Identification containing Ident_Number and Manufacturer_ID (see 4.4.3). The Ident_Number matches with the GSD specified Ident_Number. Manufacturer_ID is specified through the I&M profile defined VendorID and DeviceID (see 4.4.3).

⁶ The given table can be considered to be an example only since this document cannot foresee how future protocol versions will be defined.

The mapping between different device identification data sources is described in Table 6. Since scan results provided by the Communication Server or Gateway can convey data that is produced by the device (firmware) the device type identification mapping shall be supported by providing corresponding data in the FDI Device Package contained Catalog and Information Model.

Table 6 – Device identification information mapping

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

4.3.2 Device type revision mapping

IEC 62769-4 envisions a concept that allows to determine the compatibility between an FDI Device Package and a Device. IEC 62769-4 specifies a life cycle management process bearing on a single version information provided for the entire device.

NOTE PROFIBUS related specifications, for example PI Order No.: 2.122:2008 (GSD) and PI Order No.: 3.502:2009 (I&M) splits the device type revision into software and hardware related information. The GSD specifies the attributes Hardware_Release and Software_Release. The I&M specifies HARDWARE_REVISION and SOFTWARE_REVISION. Hardware_Release and HARDWARE_REVISION ~~must~~ will always match. Software_Release and SOFTWARE_REVISION ~~must~~ will always match.

The goal of 4.3.2 is to describe the translation rules between PROFIBUS related specifications, describing their way of providing the version information, and the IEC 62769-4 specified way of containing the version information that can be compared against the version read from the device. The purpose is to determine the compatibility between an FDI Device Package and a Device. Figure 1 depicts the problem.

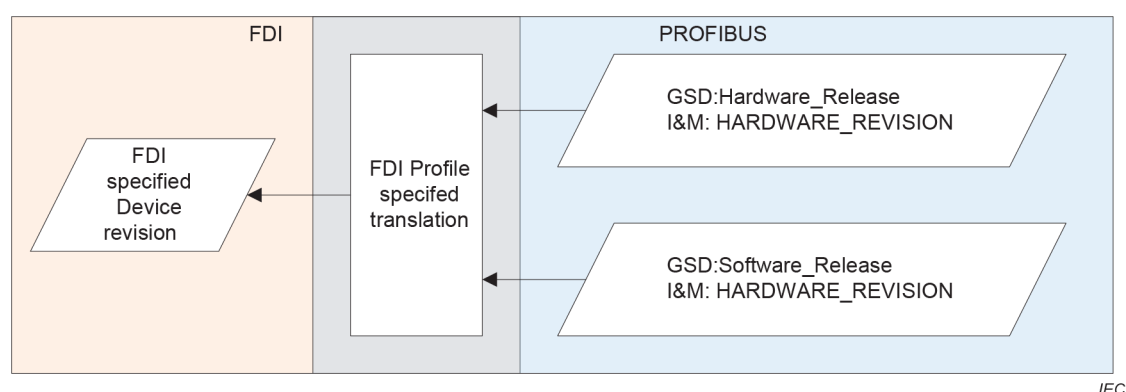


Figure 1 – Version mapping problem

The firmware of a device implements the data exchange interface which shall be described by means of the FDI Device Package content (EDD). A device firmware that implements the PROFIBUS PA profile enables the reading of the values SOFTWARE_REVISION and HARDWARE_REVISION. The access to these values shall be described in the EDD contained in the FDI Device Package.

Firmware modifications that affect the firmware implemented data exchange interface shall be reflected in the FDI Device Package. Such firmware and device description modification shall be visible in the SOFTWARE_REVISION and Software_Release.

Hardware related modifications shall be captured in the HARDWARE_REVISION and Hardware_Release. Hardware related modifications do not necessarily always require a firmware update. Thus HARDWARE_REVISION and Hardware_Release cannot be used to determine compatibility between a device and the FDI Device Package. But if a hardware modification requires firmware modifications both HARDWARE_REVISION and SOFTWARE_REVISION shall be changed. Hardware_Release and Software_Release shall be changed accordingly.

The IEC 62769-4 specifies the Catalog schema and an element DeviceVersion which is used in the element type declaration ListOfSupportedDeviceVersions. The value of the DeviceVersion shall be compared to the device provided SOFTWARE_REVISION or the GSD provided Software_Release in order to determine the compatibility between an FDI Device Package and a device.

The data format for the SOFTWARE_REVISION is a string while the DeviceVersion expects three numbers for major, minor, and revision. Therefore the following rules apply: If the string has the format <integer>.<integer>.<integer> this is transferred to major, minor, and revision (in the same order). <integer> references to simple integer number in the string such as '1' or '12', not to other representations such as hexadecimal format (e.g. 0x001A). If <integer>.<integer> is provided, this is transferred to major and minor and '0' is used for revision. If only an <integer> is provided, this is transferred to major and '0' is used for minor and revision. A leading character or a leading character and whitespace shall be ignored. For a string in any other format, the revision number shall not be considered to select the correct FDI package.

4.4 Information Model mapping

4.4.1 ProtocolType definition

The concept to derive PROFIBUS DP and PROFIBUS PA specific Network Types applies to the protocol type definition.

The protocol type Profibus_DP shall be used to identify the PROFIBUS DP communication. The type Profibus_DP is a sub type of the abstract type ProtocolType in IEC 62541-100. Table 7 specifies the allowed values of the ProtocolType attributes for the protocol type Profibus_DP.

Table 7 – Protocol type Profibus_DP

Attribute	Value				
BrowseName	Profibus_DP				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the ProtocolType defined in IEC 62541-100.					

The network type Profibus_PA_Network shall be used to build PROFIBUS PA network topologies. The type Profibus_DP_Network is a sub type of the abstract type NetworkType in IEC 62541-100. Table 8 specifies the allowed values of the ProtocolType attributes for the protocol type Profibus_PA.

Table 8 – Protocol type Profibus_PA

Attribute	Value				
BrowseName	Profibus_PA				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the ProtocolType defined in IEC 62541-100.					

4.4.2 DeviceType mapping

The DeviceType property mapping of the DeviceType node is defined in Table 9.

Table 9 – DeviceType property mapping

Property	PROFIBUS Mapping
SerialNumber	SERIAL_NUMBER (see Table 10)
RevisionCounter	REV_COUNTER (see Table 10)
Manufacturer	MANUFACTURER_ID (see Table 10) String taken from FDI package catalog (ManufacturerName from PackageT)
Model	ORDER_ID (see Table 10) String taken from FDI package catalog (Name of DeviceTypeT, which is a localized name)
DeviceRevision	Not supported
DeviceManual	Not supported
SoftwareRevision	SOFTWARE_REVISION (see Table 10)
HardwareRevision	HARDWARE_REVISION (see Table 10)

4.4.3 FunctionalGroup identification definition

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

Table 10 – PROFIBUS Device Types identification attributes

BrowseName	DataType	Mandatory/Optional
Ident_Number	UInt16	Mandatory
MANUFACTURER_ID	UInt16	Mandatory
ORDER_ID	String	Optional
SERIAL_NUMBER	String	Optional
HARDWARE_REVISION	UInt16	Optional
SOFTWARE_REVISION	String	Optional
REV_COUNTER	UInt16	Optional
PROFILE_ID	UInt16	Optional
PROFILE_SPECIFIC_TYPE	UInt16	Optional
IM_VERSION	ByteString	Optional
IM_SUPPORTED	UInt16	Optional

The BaseDataVariable instances, except Ident_Number, shall be created from VARIABLE declarations with identifiers that correspond to the browse names listed in Table 10. The BaseDataVariable instances Ident_Number shall be created from the GSD file attribute Ident_Number.

4.5 Topology elements

4.5.1 ConnectionPoint definition

In order to support different network topology engineering needs related to different physical layers used by PROFIBUS DP and PROFIBUS PA, two different ConnectionPoint types shall be defined.

The ConnectionPoint type Profibus_DP shall be used to parameterize PROFIBUS DP network access points. The ConnectionPoint type Profibus_DP is a sub type of the abstract type ConnectionPointType defined in IEC 62541-100. Table 11 specifies the allowed values of the ConnectionPoint attributes for the protocol type Profibus_DP.

Table 11 – ConnectionPoint type for Profibus_DP

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

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

```
COMPONENT ConnectionPoint_Profibus_DP
{
    LABEL "PROFIBUS DP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL PROFIBUS_DP;
    CONNECTION_POINT ConnectionPoint;
}
```

```
VARIABLE Address
{
    LABEL "Station address";
    HELP "Address of the PROFIBUS slave";
    TYPE UNSIGNED_INTEGER(1)
    {
        INITIAL_VALUE 126;
        MIN_VALUE 0;
        MAX_VALUE 126;
    }
    HANDLING READ & WRITE;
    CLASS LOCAL;
}
```

```

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

```

The ConnectionPoint type Profibus_PA shall be used to parameterize PROFIBUS PA network access points. The ConnectionPoint type Profibus_PA is a sub type of the abstract type ConnectionPointType defined in IEC 62541-100. Table 12 specifies the allowed values of the ConnectionPoint attributes for the protocol type Profibus_PA.

Table 12 – ConnectionPoint type for Profibus_PA

Attribute	Value				
BrowseName	ConnectionPoint_Profibus_PA				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
Subtype of the ConnectionPointType defined in IEC 62541-100.					
HasProperty	Variable	Address	Byte	PropertyType	Mandatory

The Property Address allowed values are 0..126.

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

```

COMPONENT ConnectionPoint_Profibus_PA
{
  LABEL "PROFIBUS PA Connection Point";
  CLASSIFICATION NETWORK_CONNECTION_POINT;
  CAN_DELETE FALSE;
  PROTOCOL PROFIBUS_PA;
}

```

```

VARIABLE Address
{
  LABEL "Station address";
  HELP "Address of the PROFIBUS slave";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(1)
  {
    INITIAL_VALUE 126;
    MIN_VALUE 0;
    MAX_VALUE 126;
  }
  HANDLING READ & WRITE;
  CLASS LOCAL;
}

```

```

COLLECTION ConnectionPoint
{
  LABEL "PROFIBUS PA Connection Point data";
  MEMBERS

```

```

    {
        CONNECTION_POINT_ADDRESS, Address;
    }
}

```

4.5.2 Communication Device definition

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

```

COMPONENT Profibus_Communication_Server
{
    LABEL "PROFIBUS communication server";
    PRODUCT_URI "urn:PROFIBUS International:PROFIBUS Communication      Server";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS
    {
        Profibus_Communication_Device_Setup
    }
}

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

```

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 in is an example for a PROFIBUS DP communication device:

```

COMPONENT Profibus_Communication_Device
{
    LABEL "PROFIBUS communication device";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { Profibus_Service_Provider_Relation }
    BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION Profibus_Service_Provider_Relation
{
    LABEL "Relation to communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        Profibus_Service_Provider{AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

In an actual communication device the value "ConnectionPoint_Profibus_DP" needs to be adapted according to the supported protocol and the related connection point definitions given in 4.5. The attribute BYTE_ORDER value is to be set according to the protocol.

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 below is an example for a PROFINET IO communication service provider component:

The component reference (ConnectionPoint_Profibus_DP) corresponds to the related connection point definition in 4.5. The attribute BYTE_ORDER value is to be set according to the protocol.

```
COMPONENT Profibus_Service_Provider
{
    LABEL "PROFIBUS communication service provider";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
    COMPONENT_RELATIONS
    {
        Profibus_Service_Provider_Connection_Point_Relation
    }
    BYTE_ORDER BIG_ENDIAN; // EDDL extension
}

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

4.5.4 Network definition

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

```
COMPONENT Network_Profibus_DP
{
    LABEL "PROFIBUS DP Network";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK;
    COMPONENT_RELATIONS
    {
        Profibus_DP_Network_Connection_Point_Relation
    }
}

COMPONENT_RELATION Profibus_DP_Network_Connection_Point_Relation
{
    LABEL "Relation between network and Connection Point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {Address}
    COMPONENTS
    {
        ConnectionPoint_Profibus_DP
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 32;
}
```

4.6 Methods

4.6.1 Methods for FDI Communication Servers

4.6.1.1 General

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

4.6.1.2 Connect

Signature:

```
Connect (
    [in]   ByteString      CommunicationRelationId,
    [in]   byte            Address,
    [in]   UInt16          ManufacturerId,
    [out]  Int32            ServiceError);
```

Table 13 provides the description of the arguments.

Table 13 – Method Connect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the Communication Server hardware. The nodeId allows finding the direct parent-child relation.
Address	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 4.5. The argument value holds the device's network address.
ManufacturerId	The argument shall correspond to BaseDataVariable MANUFACTURER_ID that is organized in the FunctionalGroup Identification (see 5.4.3).
ServiceError	0: OK/execution finished, connection established successfully -1: Connect Failed/canceled by caller -3: Connect Failed/device not found -4: Connect Failed/invalid device address -5: Connect Failed/invalid device identification

4.6.1.3 Disconnect

Signature:

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

Table 14 provides the description of the arguments.

Table 14 – Method Disconnect arguments

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

4.6.1.4 Transfer

Signature

```

Transfer (
    [in]   ByteString   CommunicationRelationId,
    [in]   String       OPERATION,
    [in]   unsigned char SLOT,
    [in]   unsigned char INDEX,
    [in]   ByteString   REQUEST,
    [out]  ByteString   REPLY,
    [out]  ByteString   RESPONSE_CODES,
    [out]  Int32         ServiceError);

```

Table 15 provides the description of the arguments.

Table 15 – 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 data transfer direction. Allowed values are "READ" and "WRITE".
SLOT	The argument name shall match with the corresponding COMMAND – attribute name SLOT. The argument value shall come from the attribute value of COMMAND – attribute SLOT of the corresponding COMMAND that shall be processed.
INDEX	The argument name shall match with the corresponding COMMAND – attribute name INDEX. The argument value shall come from attribute value of COMMAND – attribute INDEX of the corresponding COMMAND that shall be processed.
REQUEST	The argument name shall match with the corresponding COMMAND sub-element name REQUEST. The byte stream submitted through the argument is created from definitions provided by the REQUEST element of the corresponding COMMAND that shall be processed.
REPLY	The argument name shall match with the corresponding COMMAND sub-element name REPLY. The byte stream returned by this argument applies to definitions provided by the REPLY element of the corresponding COMMAND that shall be processed.
RESPONSE_CODES	The argument name shall match with the COMMAND sub-element name RESPONSE_CODES. The argument value conveys the PROFIBUS specific communication service response bytes.
ServiceError	0: OK/execution finished, REPLY and RESPONSE_CODES contain the results -1: Transfer Failed/canceled by caller -3: Transfer Failed/no existing communication relation. -4: Transfer Failed/invalid communication relation identifier -5: Transfer Failed/invalid sendData content -6: Transfer Failed/invalid receiveData format

4.6.1.5 GetPublishedData

This method is not supported by PROFIBUS.

4.6.1.6 SetAddress

Signature

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

Table 16 provides the description of the arguments.

Table 16 – Method SetAddress arguments

Argument	Description
OldAddress	The argument value holds the current address of a device. Allowed values are 0..126.
NewAddress	The argument value holds the new address for a device. Allowed values are 0..125.
ServiceError	0: OK/execution finished successfully -1: SetAddress Failed/canceled by caller -3: SetAddress Failed/not initialized -4: SetAddress Failed/not connected to a network -5: SetAddress Failed/no device found responding to oldAddress -6: SetAddress Failed/duplicate address error -7: SetAddress Failed/device did not accept new address -8: SetAddress Failed/invalid oldAddress (in terms of syntax, data type, data format, and so on) -9: SetAddress Failed/invalid newAddress (in terms of syntax, data type, data format, and so on) -10: SetAddress Failed/not possible in status connected

4.6.1.7 Scan

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

4.6.1.8 ResetScan

The method signature specified in IEC 62769-7 applies.

4.6.2 Methods for Gateways

4.6.2.1 General

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

4.6.2.2 Connect

Subclause 4.6.2.2 describes the PROFIBUS Gateway specific implementation of the service Connect specified in IEC 62769-7.

```

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

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

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

```

Table 17 provides the description of the arguments.

Table 17 – Connect service arguments

Argument	Description
CommunicationRelationId	The argument value contains the noId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the Communication Server hardware. The noId allows finding the direct parent-child relation.
Address	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 4.5. The argument value holds the device's network address.
ManufacturerId	The argument shall correspond to BaseDataVariable MANUFACTURER_ID that is organized in the FunctionalGroup Identification (see 5.4.3).
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	1: OK/function started asynchronously, result has to be polled with EndConnect 0: OK/execution finished, connection established successfully -1: Connect Failed/canceled by caller -2: Call Failed/unknown service ID -3: Connect Failed/device not found -4: Connect Failed/invalid device address -5: Connect Failed/invalid device identification

4.6.2.3 Disconnect

Subclause 4.6.2.3 describes the PROFIBUS specific implementation of the service Disconnect specified in IEC 62769-7.

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

All the arguments of the Disconnect method are described in Table 14.

4.6.2.4 Transfer

Subclause 4.6.2.4 describes the PROFIBUS specific implementation of the service Transfer specified in IEC 62769-7.

```
METHOD BeginTransfer(
    DD_STRING      CommunicationRelationId,
    DD_STRING      OPERATION,
    unsigned char   char-SLOT,
    unsigned char   char-INDEX,
    DD_STRING      REQUEST,
    DD_STRING      &REPLY,
    DD_STRING      &RESPONSE_CODES,
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError)
{
    ACCESS ONLINE;
    DEFINITION{<Gateway specific implementation>}
}

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

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

Table 18 provides the description of the arguments.

Table 18 – 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 direction. Allowed values are "READ" and "WRITE".
SLOT	The argument name shall match with the corresponding COMMAND – attribute name SLOT. The argument value shall come from the attribute value of COMMAND – attribute SLOT of the corresponding COMMAND that shall be processed.
INDEX	The argument name shall match with the corresponding COMMAND – attribute name INDEX. The argument value shall come from the attribute value of COMMAND – attribute INDEX of the corresponding COMMAND that shall be processed.
REQUEST	The argument name shall match with corresponding COMMAND sub-element name REQUEST. The byte stream submitted through argument is created from definitions provided by the REQUEST element of the corresponding COMMAND that shall be processed.
REPLY	The argument name shall match with corresponding COMMAND sub-element name REPLY. The byte stream returned by this argument applies to definitions provided by the REPLY element of the corresponding COMMAND that shall be processed.
RESPONSE_CODES	The argument name shall match with the COMMAND sub-element name RESPONSE_CODES. The argument value conveys the PROFIBUS specific communication service response bytes.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndTransfer invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK/function started asynchronously, result has to be polled with EndTransfer 0: OK/execution finished, ReceivedData contains the result -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

4.6.2.5 GetPublishedData

This method is not supported in PROFIBUS.

4.6.2.6 SetAddress

Subclause 4.6.2.6 describes the PROFIBUS specific implementation of the service SetAddress specified in IEC 62769-7.

```

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

```

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

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

```

Table 19 provides the description of the arguments.

Table 19 – Method SetAddress arguments

Argument	Description
OldAddress	The argument value holds the current address of a device. Allowed values are 0..126.
NewAddress	The argument value holds the new address for a device. Allowed values are 0..125.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndTransfer invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK/function started asynchronously, result has to be polled with EndSetAddress 0: OK/execution finished successfully -1: SetAddress Failed/canceled by caller -2: Call Failed/unknown service ID -3: SetAddress Failed/not initialized -4: SetAddress Failed/not connected to a network -5: SetAddress Failed/no device found responding to oldAddress -6: SetAddress Failed/duplicate address error -7: SetAddress Failed/device did not accept new address -8: SetAddress Failed/invalid oldAddress (in terms of syntax, data type, data format, and so on) -9: SetAddress Failed/invalid newAddress (in terms of syntax, data type, data format, and so on) -10: SetAddress Failed/not possible in status connected

4.6.2.7 Scan

The method signature specified in IEC 62769-7 applies. The PROFIBUS gateway business logic shall create the scan result following IEC 62769-7. The following definitions shall be present in the COMPONENT declaration that holds the definitions for a communication device. The data structure corresponds to the data structure defined in the XML schema in 4.2. The SCAN_LIST attribute inside the COMPONENT declaration shall refer to LIST TopologyScanResult.

```
VARIABLE DP_Address
{
    LABEL "Station address";
    TYPE UNSIGNED_INTEGER(1);
    CLASS LOCAL;
}

VARIABLE DP_DeviceID
{
    LABEL "Device ID";
    TYPE UNSIGNED_INTEGER(2);
    CLASS LOCAL;
}

COLLECTION ScanItemType
{
    MEMBERS
    {
        DP_ADDRESS_ID, DP_Address;
        DP_DEVICE_ID, DP_DeviceID;
    }
}

LIST TopologyScanResult
{
    TYPE ScanItemType;
    CAPACITY 126;
}
```

4.6.2.8 ScanNext

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

Annex A (normative)

Topology Scan result schema

A.1 General

The topology scan result schema specified in Annex A describes the PROFIBUS 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 Network

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

The XML schema for a Network element is:

```
<xs:element name="Network" type="PI:ProfibusNetworkT"/>
```

A.3 ProfibusNetworkT

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

The XML schema for a ProfibusNetworkT type is:

```
<xs:complexType name="ProfibusNetworkT">
  <xs:sequence>
    <xs:element name="ConnectionPoint"
      type="PI:ProfibusConnectionPointT" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

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

Table A.1 – Elements of ProfibusNetworkT

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

A.4 ProfibusConnectionPointT

The XML schema for a ProfibusConnectionPointT type is:

```
<xs:complexType name="ProfibusConnectionPointT">
  <xs:sequence>
    <xs:element name="Identification"
      type="PI:ProfibusIdentificationT"/>
  </xs:sequence>
</xs:complexType>
```

```

</xs:sequence>
<xs:attribute name="Address" type="PI:ProfibusAddressT"
  use="required"/>
</xs:complexType>

```

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

Table A.2 – Attributes of ProfibusConnectionPointT

Attribute	Description
Address	The Attribute value holds the address of the network connected device.

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

Table A.3 – Elements of ProfibusConnectionPointT

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

A.5 ProfibusIdentificationT

The element content corresponds to the "FunctionalGroup Identification".

The XML schema for a ProfibusIdentificationT type is:

```

<xs:complexType name="ProfibusIdentificationT">
  <xs:attribute name="Ident_Number" type="xs:unsignedShort "
    use="required"/>
  <xs:attribute name="MANUFACTURER_ID" type="xs:unsignedShort "
    use="optional"/>
  <xs:attribute name="ORDER_ID" type="xs:string" use="optional"/>
  <xs:attribute name="SERIAL_NUMBER" type="xs:string"
    use="optional"/>
  <xs:attribute name="HARDWARE_REVISION" type="xs:unsignedShort "
    use="optional"/>
  <xs:attribute name="SOFTWARE_REVISION" type="xs:string"
    use="optional"/>
  <xs:attribute name="REV_COUNTER" type="xs:unsignedShort "
    use="optional"/>
  <xs:attribute name="PROFILE_ID" type="xs:unsignedShort "
    use="optional"/>
  <xs:attribute name="PROFILE_SPECIFIC_TYPE" type="xs:unsignedShort "
    use="optional"/>
  <xs:attribute name="IM_VERSION" type="xs:string" use="optional"/>
  <xs:attribute name="IM_SUPPORTED" type="xs:unsignedShort "
    use="optional"/>
</xs:complexType>

```

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