

CONSOLIDATED VERSION

VERSION CONSOLIDÉE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2016 IEC, Geneva, Switzerland

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

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

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
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 corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

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 also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 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 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: csc@iec.ch.

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

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

CONSOLIDATED VERSION

VERSION CONSOLIDÉE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.05; 35.110

ISBN 978-2-8322-3498-3

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

REDLINE VERSION

VERSION REDLINE



**Field device tool (FDT) interface specification –
Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2**

**Spécification des interfaces des outils des dispositifs de terrain (FDT) –
Partie 303-1: Intégration des profils de communication – IEC 61784 CP 3/1
et CP 3/2**

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope	8
2 Normative references	8
3 Terms, definitions, symbols, abbreviated terms and conventions	9
3.1 Terms and definitions	9
3.2 Symbols and abbreviated terms	9
3.3 Conventions	10
3.3.1 Data type names and references to data types	10
3.3.2 Vocabulary for requirements.....	10
3.3.3 Use of UML.....	10
4 Bus category	10
5 Access to instance and device data	11
5.1 Process Channel objects provided by DTM.....	11
5.2 DTM services to access instance and device data	11
6 Protocol specific behavior	11
6.1 PROFIBUS device model.....	11
6.2 Configuration and parameterization of PROFIBUS devices	12
6.2.1 General	12
6.2.2 Monolithic DTM for a modular PROFIBUS device.....	13
6.2.3 Modular DTM for a modular PROFIBUS device	13
6.3 Support for DPV0 configuration.....	14
6.4 PROFIBUS slaves operating without a cyclic PROFIBUS master	14
6.5 PROFIBUS-related information of a slave DTM	14
6.5.1 General	14
6.5.2 Bus Master Configuration Part (BMCP).....	15
7 Protocol specific usage of general data types.....	25
8 Protocol specific common data types	27
9 Network management data types	27
9.1 General.....	27
9.1.1 Configuration	27
9.1.2 Process Channel.....	28
9.1.3 Parameterization.....	28
9.2 Master-bus parameter set	29
9.3 Slave bus parameter set	29
9.4 Module and channel data.....	30
9.5 GSD information	33
9.5.1 General	33
9.5.2 GSD for gateway devices	33
10 Communication data types.....	34
10.1 General.....	34
10.2 Error information provided by Communication Channel.....	34
10.3 DPV0 communication.....	34
10.4 DPV1 communication.....	41
11 Channel parameter data types	42

12	Device identification	47
12.1	General	47
12.2	Protocol specific handling of the data type STRING	47
12.3	Common device type identification data types	47
12.4	Topology scan data types	52
12.5	Scan identification data types	52
12.6	Device type identification data types – provided by DTM.....	55
12.7	Identification information in GUI	58
13	ProfiSafe	58
13.1	Motivation	58
13.2	General parameter handling.....	58
13.3	ProfiSafe individual device parameter	59
	Bibliography	61
	Figure 1 – Part 303-1 of the IEC 62453 series	7
	Figure 2 – FDT PROFIBUS device model	12
	Figure 3 – Example for IO data within datagrams.....	31
	Figure 4 – F-Parameter and individual device parameter	59
	Figure 5 – Data structure of ProfiSafe individual device parameters	60
	Table 1 – Protocol identifiers.....	10
	Table 2 – Physical layer identifiers	10
	Table 3 – ~~BMPC~~ BMCP Part1 – General configuration	16
	Table 4 – ~~BMPC~~ BMCP Part2 – Parameter data	16
	Table 5 – ~~BMPC~~ BMCP Part3 – Configuration data	17
	Table 6 – Part 4: Address table and slave user parameters.....	18
	Table 7 – Part 4: Extended Prm data.....	18
	Table 8 – Complete BMCP	19
	Table 9 – Protocol specific usage of general data types.....	25
	Table 10 – Bus parameter set for master device	29
	Table 11 – Bus parameter set for slave device	30
	Table 12 – Signal channels within the data frame	32
	Table 13 – Simple DPV0 communication data types	35
	Table 14 – Structured DPV0 Communication data types	35
	Table 15 – Availability of services for Master Class1 (C1)	40
	Table 16 – Availability of services for Master Class2 (C2)	40
	Table 17 – Simple DPV1 communication data types	41
	Table 18 – Structured DPV1 communication data types	42
	Table 19 – Mapping of DPV1 data types to FDT data types.....	44
	Table 20 – Simple Channel Parameter data types.....	45
	Table 21 – Structured Channel Parameter data types	46
	Table 22 – Identification data types with Profibus DP specific mapping	48
	Table 23 – Identification data types with Profibus I&M specific mapping	49
	Table 24 – Identification data types with Profibus PA specific mapping	51

Table 25 – Simple identification data types with protocol independent semantics	52
Table 26 – Structured identification data types with protocol independent semantics	52
Table 27 – Simple topology scan data types	52
Table 28 – Structured topology scan data types.....	52
Table 29 – Simple scan identification data types.....	53
Table 30 – Structured scan identification data types	53
Table 31 – Structured device identification data types	56
Table 32 – DataLink Layer Identifiers	10

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2

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.

DISCLAIMER

This Consolidated version is not an official IEC Standard and has been prepared for user convenience. Only the current versions of the standard and its amendment(s) are to be considered the official documents.

This Consolidated version of IEC 62453-303-1 bears the edition number 1.1. It consists of the first edition (2009-06) [documents 65E/127/FDIS and 65E/140/RVD] and its amendment 1 (2016-06) [documents 65E/336/CDV and 65E/395A/RVC]. The technical content is identical to the base edition and its amendment.

In this Redline version, a vertical line in the margin shows where the technical content is modified by amendment 1. Additions are in green text, deletions are in strikethrough red text. A separate Final version with all changes accepted is available in this publication.

International Standard IEC 62453-303-1 been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

Each part of the IEC 62453-3xy series is intended to be read in conjunction with IEC 62453-2.

The French version of this standard has not been voted upon.

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

A list of all parts of the IEC 62453 series, under the general title *Field Device Tool (FDT) interface specification*, can be found on the IEC website.

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

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

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 publication using a colour printer.

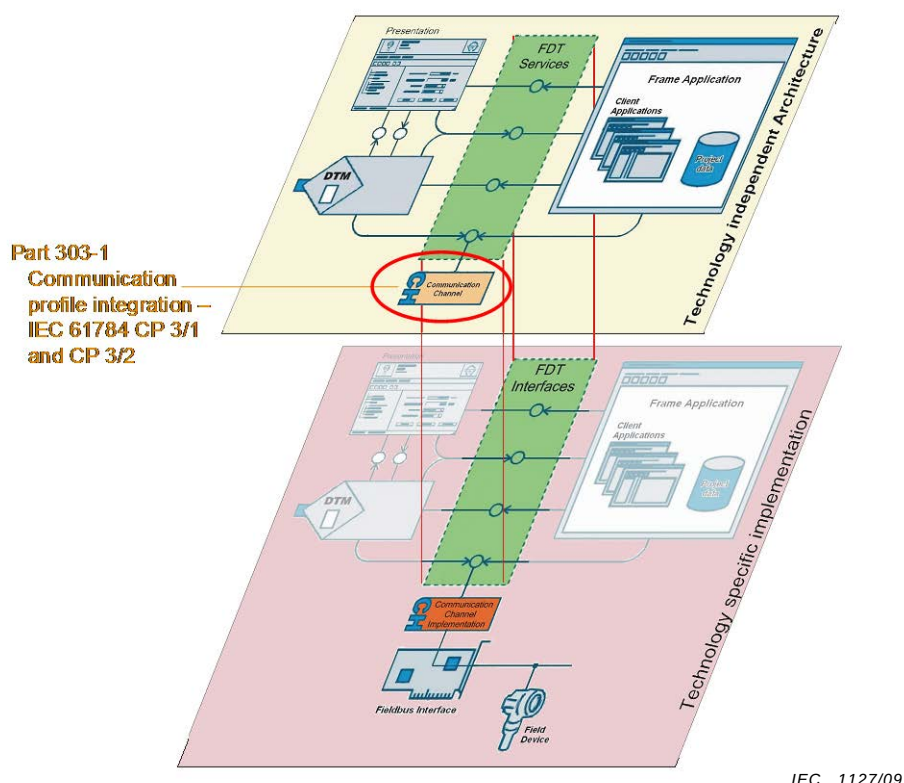
INTRODUCTION

This part of IEC 62453 is an interface specification for developers of FDT (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

With the integration of fieldbusses into control systems, there are a few other tasks which need to be performed. In addition to fieldbus- and device-specific tools, there is a need to integrate these tools into higher-level system-wide planning- or engineering tools. In particular, for use in extensive and heterogeneous control systems, typically in the area of the process industry, the unambiguous definition of engineering interfaces that are easy to use for all those involved is of great importance.

A device-specific software component, called DTM (Device Type Manager), is supplied by the field device manufacturer with its device. The DTM is integrated into engineering tools via the FDT interfaces defined in this specification. The approach to integration is in general open for all kinds of fieldbusses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

Figure 1 shows how IEC 62453–303-1 is aligned in the structure of the IEC 62453 series.



IEC 1127/09

Figure 1 – Part 303-1 of the IEC 62453 series

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 303-1: Communication profile integration – IEC 61784 CP 3/1 and CP 3/2

1 Scope

Communication Profile 3/1 and Communication Profile 3/2 (commonly known as PROFIBUS™¹) defines communication profiles based on IEC 61158-2 Type 3, IEC 61158-3-3, IEC 61158-4-3, IEC 61158-5-3, and IEC 61158-6-3. The basic profiles CP 3/1 (PROFIBUS DP) and CP 3/2 (PROFIBUS PA) are defined in IEC 61784-1.

This part of IEC 62453 provides information for integrating the PROFIBUS protocol into the FDT interface specification (IEC 62453–2).

This part of the IEC 62453 specifies communication and other services.

This specification neither contains the FDT specification nor modifies it.

2 Normative references

The following referenced documents are indispensable for the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*

IEC 61158-2:2014, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-3-3, *Industrial communication networks – Fieldbus specifications – Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3 *Industrial communication networks – Fieldbus specifications – Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-5-3: *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements*

IEC 61158-6-3, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

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

¹ PROFIBUS™ is a trade names of the non-profit organization PROFIBUS Nutzerorganisation e.V. (PNO). This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this standard does not require use of the registered logos for PROFIBUS™. Use of the registered logos for PROFIBUS™ requires permission of PNO.

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2009, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

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

3.1.1

bus interface module

module of a field device that provides the connection to the fieldbus

3.1.2

CP 3/1

Communication profile of CPF3, featuring asynchronous transmission; RS 485 (ANSI TIA/EIA RS-485-A); optional RS 485-IS; plastic fiber; glass multi mode fiber or glass single mode fiber; PCF fiber

3.1.3

CP 3/2

Communication profile of CPF3, featuring synchronous transmission; manchester coded and bus powered (MBP); optional intrinsically safe (MBP-IS) and lower power (MBP-LP)

3.2 Symbols and abbreviated terms

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

ANSI	American National Standards Institute (http://www.ansi.org)
BIM	Bus Interface Module
BMCP	Bus Master Configuration Part
CFG	Configuration data used during initialization of PROFIBUS slave device
DCS	Distributed Control System
DP	Decentralized Peripherals
EIA	Electronic Industries Alliance
FDL	Fieldbus Data Link layer
FMA	Fieldbus Management layer
FMS	Fieldbus Message Specification
GSD	General Station Description
MBP	Manchester coded Bus Powered
I&M	Identification and maintenance functions
PA	Process Automation
PCF	Polymer Clad Fibre
PROFIBUS	Process Field Bus
RS	Radio Sector / Recommended Standard
TIA	Telecommunications Industry Association

3.3 Conventions

3.3.1 Data type names and references to data types

The conventions for naming and referencing of data types are explained in IEC 62453-2, Clause A.1

3.3.2 Vocabulary for requirements

The following expressions are used when specifying requirements.

Usage of “shall” or “mandatory”	No exceptions allowed.
Usage of “should” or “recommended”	Strong recommendation. It may make sense in special exceptional cases to differ from the described behaviour.
Usage of “conditional”	Function or behaviour shall be provided, depending on defined conditions.
Usage of “can” or “optional”	Function or behaviour may be provided, depending on defined conditions.

3.3.3 Use of UML

Figures in this document are using the UML notation as defined in Annex A of IEC 62453–1.

4 Bus category

CP 3/1 and CP 3/2 protocols are identified in the protocolId element of the structured data type 'fdt:BusCategory' by the following unique identifiers (Table 1):

Table 1 – Protocol identifiers

Identifier value	ProtocolId name	Description
036D1497-387B-11D4-86E1-00E0987270B9	'Profibus DP/V0'	Support of Profibus DP V0 protocol
036D1499-387B-11D4-86E1-00E0987270B9	'Profibus DP/V1'	Support of Profibus DP V1 protocol

CP 3/1 AND CP 3/2 protocols are using the following unique identifiers in physicalLayer members within PhysicalLayer data type (Table 2):

Table 2 – Physical layer identifiers

Identifier value	Name	Description
036D1590-387B-11D4-86E1-00E0987270B9	MBP	IEC 61158-2 (MBP, Profibus PA)
036D1591-387B-11D4-86E1-00E0987270B9	RS485	IEC 61158-2:2014, Clause 22 (RS485, PROFIBUS DP)
036D1592-387B-11D4-86E1-00E0987270B9	Fiber Optic	IEC 61158-2:2014, Clause 23 (Fiber Optic cable, PROFIBUS DP)
036D1593-387B-11D4-86E1-00E0987270B9		Ethernet (depreciated, do not use)

Table 32 defines which DataLinkLayer shall be used in combination with the BusCategory values defined in Table 32.

Table 32 – DataLink Layer Identifiers

Identifier value	Name	Description
50A21B35-7EE7-4999-8174-70396929C0B4	PROFIBUS FDL	PROFIBUS FDL
CDF338DC-E9A3-4D13-91AC-60A43DCB2904	PROFIBUS FMA1/2	PROFIBUS FMA1/2

5 Access to instance and device data

5.1 Process Channel objects provided by DTM

The minimum set of provided data should be: Process values modeled as channel objects including the ranges and scaling.

5.2 DTM services to access instance and device data

The services InstanceDataInformation and DeviceDataInformation shall provide access to at least all parameters of the Physical Block and the status and Out value of the Function Blocks shall be exposed.

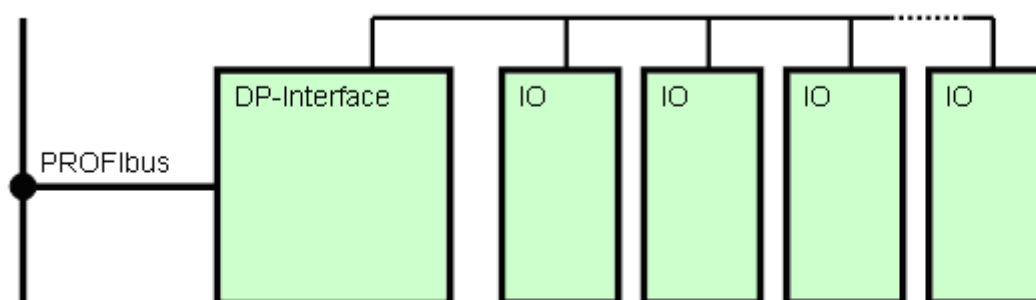
According to IEC 62453-2, at least one set of semantic information (one per supported fieldbus protocol) shall be provided for each accessible data object, using the 'SemanticInformation' general data type. The corresponding data type 'applicationDomain' shall have a value defined for Profibus and the data type 'semanticId' shall have an appropriate value, as specified in Table 9.

6 Protocol specific behavior

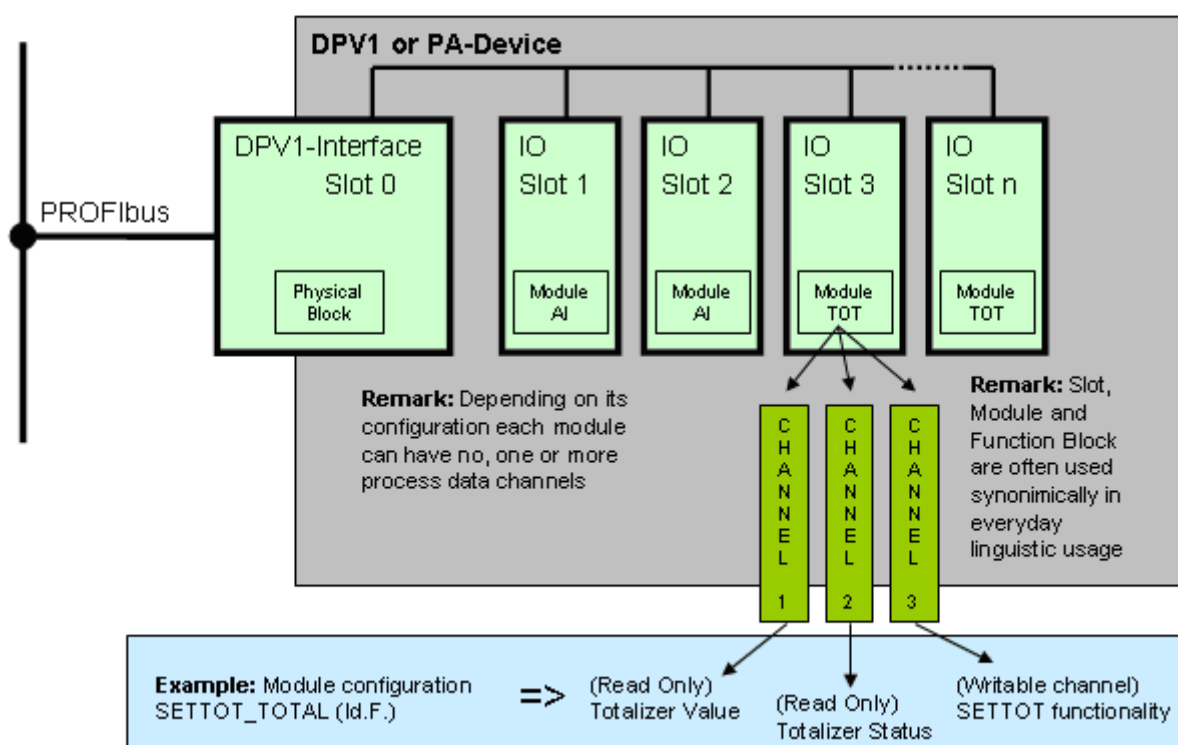
6.1 PROFIBUS device model

FDT extends the PROFIBUS device model by using Process Channels for description of I/O values (see Figure 2).

Classical View of PROFIBUS device



PROFIBUS notations from a monolithic DPV1 or PA device DTMs point of view



IEC 1128/09

Figure 2 – FDT PROFIBUS device model

6.2 Configuration and parameterization of PROFIBUS devices

6.2.1 General

In a GSD-based configuration tool the user defines the configuration and sets the appropriate parameters for the modules. The configuration tool creates the configuration string and the parameter string that are used to set up the slave properly.

With FDT the configuration and parameterization of the devices is no longer executed only by a central component; it moved partly into the DTMs. A DTM is responsible for providing configuration and parameterization information for the cyclic master that puts the PROFIBUS slaves in operation.

A DTM is used to adjust a field device to its specific application. Within PROFIBUS, there are three different aspects of adjustment:

- parameterization: usr prm data (used in the PROFIBUS service SET_PRM for setting up the cyclic communication and the specific behavior of the device);

- application parameterization: application specific parameters (transmitted via acyclic read/write PROFIBUS services);
- configuration: configuration data (used in the PROFIBUS service CHK_CFG for definition of the format and length of the input/output data that are transmitted within cyclic communication).

The application parameterization transmitted via acyclic communication is not in the scope of this document. Within this document the term parameterization represents communication parameterization (SET_PRM).

6.2.2 Monolithic DTM for a modular PROFIBUS device

A monolithic DTM is one single DTM that represents the complete device with its Bus Interface Module (BIM) and its I/O modules. In general, such a DTM offers a configuration dialog (presentation object) that allows definition of the used BIM and modules. The configuration dialog must be available via the FDT standard function “Configure” (see [1] 4.3 Operation Configuration).

Not all PROFIBUS devices require a configuration dialog. That is why not all DTMs provide the “Configure” function. This is valid only for non-modular PROFIBUS devices if the Usr_Prm-Data cannot be changed.

The configuration dialog allows changing the data only in offline mode if the data set can be locked.

6.2.3 Modular DTM for a modular PROFIBUS device

Separate DTMs represent the BIM (Device DTM) and the particular I/O modules (Module DTMs). The effort developing such a modular DTM is normally higher than in the case of a monolithic DTM, because:

- a private protocol has to be implemented between BIM and I/O modules to ensure that only a Module DTM can be added to the BIM DTM. This requires an own protocol ID and the adaptation / creation of communication;
- in some cases, additional private interfaces are necessary to exchange information between Device DTM for BIM and Module DTMs.

Implementing a Modular DTM results in the following advantages:

- the project represents the device structure;
- the user is able to access module-related information directly as a function of the Module DTM;
- the FDT specification defines a mechanism to identify DTMs. With these mechanisms it is possible to provide support for scanning the modules below the BIM and generate the topology automatically;
- supporting a new type of BIM or I/O module requires an additional DTM “only” and does not affect existing components. This may result in reduced test effort.

The configuration data to set up the PROFIBUS configuration must be provided by the Device DTM (representing the BIM). This configuration data may be generated from information of the instantiated Child DTMs and by using a configuration dialog.

Modular DTMs should be provided for modular devices (e.g. a plant operator may add/remove modules). Monolithic DTMs are used to represent devices that show no modularity (e.g. PA devices).

6.3 Support for DPV0 configuration

A PROFIBUS slave is configured by a cyclic master and communicates via PROFIBUS DP. In addition to this the slave may support DPV1 communication.

A Gateway DTM for a PROFIBUS slave does not have to provide communication for the DPV0 communication schema. For example, there is a remote I/O system with HART modules. It may have a Gateway DTM that requires the DPV1 protocol and provides the HART protocol (defined in the information document and in the parameter document). This enables HART Device DTMs to communicate with their devices via the Gateway DTM and via Communication DTM for DPV1. Following the specification the Gateway DTM delivers channel parameter documents for both protocols DPV1 and HART. The protocolId is a member of NetworkManagementInfo data type.

The Process Channels ~~must~~ shall provide Channel Parameter documents for DPV1 including all information to allow integration into the control system (e.g. DPAddress of the IO value if available).

6.4 PROFIBUS slaves operating without a cyclic PROFIBUS master

In most cases, a PROFIBUS slave is configured and parameterized by a cyclic PROFIBUS master device. So a running master device in the network is required.

Some slaves are able to allow acyclic communication without a running cyclic master. Especially in the case of gateway functionality this is an eminent advantage because they allow the parameterization of field devices connected to them by using an acyclic bus master. So instrument specialists are able to work with field devices also in case the controller is not yet working.

If a master starts communication, these devices start to detect bus speed and settings to react properly. This may take some time.

In the following, two cases are described that a user may keep in mind when working with such devices.

Use case 1:

The user performs a network scan. The Communication DTM tries to read diagnostic data via a GetDiagnose Request but does not receive a response. The device is not detected by the Communication DTM. This occurs mostly when the device has a low PROFIBUS address. The reason is that the device has not completed bus speed / bus setting detection as it was asked for their diagnostic data. The workaround is to give these devices a higher PROFIBUS address.

Use Cases 2:

The user tries to connect a field device linked to the gateway that supports DPV1 without a running cyclic master. This can lead to an error message because the gateway device has not completed bus speed / bus setting detection as it was asked for a connection. So the user has to try to connect again. This happens only in very rare situations.

6.5 PROFIBUS-related information of a slave DTM

6.5.1 General

The information used by a cyclic master device to set up the PROFIBUS network properly and allow cyclic communication between control system and slave devices is provided by a DTM in

- Bus Master Configuration Part (BMCP);
- GSD information;
- internal topology,
- process channels.

A DTM of a PROFIBUS slave must deliver these parts of PROFIBUS-related information to get integrated into a FDT-based Engineering System. In the next subclauses, a more detailed description is given on how to generate and how to provide this information. This depends on the kind of DTM (see 6.2 Configuration and parameterization of PROFIBUS devices).

6.5.2 Bus Master Configuration Part (BMCP)

6.5.2.1 BMCP introduction

The BMCP of one single DTM instance describes the actual parameter and configuration data of the corresponding PROFIBUS slave. Each DTM representing a PROFIBUS slave device must provide a Bus Master Configuration Part. The BMCP is provided in the `busMasterConfigurationPart` member of `NetworkManagementInfo` data type. This information is obtained by calling service `NetworkManagementInfoRead`.

The BMCP includes information about the configuration and the parameters for the slave. The BMCP is provided by the DTM and is required in order to generate the master configuration.

The BMCP contains data which might be changed during master configuration. That means that the BMCP may be changed and transferred back to the slave DTM by calling `NetworkManagementInfoWrite`. A Slave DTM must accept the new information and recompute the configuration / internal parameters to match the new BMCP.

DTM must check whether the new values are according to the capabilities of the device. The `NetworkManagementInfoWrite` call will be refused if the device can not handle the new values.

6.5.2.2 Creating the BMCP

This subclause explains the meaning of the individual elements of the BMCP in details.

The BCMP may be generated from the GSD information of a PROFIBUS device.

The BMCP is divided into four parts that are explained in the following subclauses.

The explanations use the GSD keywords and reference the PROFIBUS specification. See also the table with complete BMCP in 6.5.2.2.1 .

6.5.2.2.1 Part 1: From Slave_Para_Len to Octet 15

The first part consists of a fixed set of bytes described in the table below.

Table 3 – ~~BMPC~~ BMCP Part1 – General configuration

Byte	Name	Defined in	Notice
0	Slave_Para_Len	[5] 6.2.12.1	Length of the BMCP including this value
1			
2	SI_Flag	[5] 6.2.12.2	The following GSD values are used [6]: • Bit 0 : reserved, • Bit 1: Extra_Alarm_SAP, • Bit 2: DPV1_Data_Types, • Bit 3: DPV1_Supported, • Bit 4: Publisher_Enable, • Bit 5 : Fail_Safe, . The following bits are not based on GSD values: • Bit 6 : New_Prm, • Bit 7 : Active
3	Slave Type	[5] 6.2.12.3	Value is 0 (= DP-Slave)
4	Max_Diag_Data_Len	[5] 6.2.12.4	The following GSD value is used [6]: Max_Diag_Data_Len
5	Max_Alarm_Len	[5] 6.2.12.5	
6	Max_Channel_Data_Len	[5] 6.2.12.6	This field defines how much data can be transferred between slave and master. In this case, the maximum of these GSD values [6] must be calculated: • Max_Data_Len, • C1_Max_Data_Len plus 4 Bytes (Function Num, Slot_Number, Length)
7	Diag_Update_Delay	[5] 6.2.12.7	The following GSD value is used [6]:Diag_Update_Delay
8	Alarm_Mode	[5] 6.2.12.8	
9	Add_SI_Flag	[5] 6.2.12.9	
10	C1_Timeout	[5] 6.2.12.10	The following GSD value is used [6]:C1_Response_Timeout
11			
12	Reserved		
13			
14			
15			

6.5.2.2.2 Part 2: Parameter data

Part 2 of the BMCP are the slave parameter data.

Table 4 – ~~BMPC~~ BMCP Part2 – Parameter data

Byte	Name	Defined in	Notice
16	Prm_Data_Len	[5] 6.2.12.11	Length of the Parameter Data including this value
17			
18	Station Status	[5] 6.2.4.1	The following GSD values are used [6]: • Bit 0-2: reserved, • Bit 3: WD_On, • Bit 4: Freeze_Req, • Bit 5: Sync_Req, • Bit 6,7(Lock/Unlock Request): [5] Table 402
19	WatchDog1	[5] 6.2.4.2	These values (WD_Fact_1 and WD_Fact_2) depend on the baud rate. A master should set these values and slaves should handle new values
20	WatchDog2	[5] 6.2.4.3	
21	Min Tsdr	[5] 6.2.4.4	Default value is 11 bit times [1]
22	Ident_Number	[5] 6.2.3.5	The following GSD value is used [6]: Ident_Number

Byte	Name	Defined in	Notice
23			
24	Group_Ident	[5] 6.2.4.5	Indicates the group assignment of the slave in a bitwise coded form
25	DPV1_Status_1	[5] 6.2.4.7	The following GSD values are used [6]: • Bit 0,1: reserved, • Bit 2: WD_Base_1ms, • Bit 3 – 5: reserved, • Bit 6: Fail_Safe, • Bit 7 : DPV1_Enable, These bits report the slave capabilities to the master and are changed by the MS0 master following its capabilities
26	DPV1_Status_2	[5] 6.2.4.8	The following GSD values are used [6]: • Bit 0: Check_Cfg_Mode, • Bit 1: reserved, • Bit 2: Enable_Update_Alarm, • Bit 3: Enable_Status_Alarm, • Bit 4: Enable_Manufacturer_Specific_Alarm, • Bit 5: Enable_Diagnostic_Alarm, • Bit 6: Enable_Process_Alarm, • Bit 7: Enable_Pull_Plug_Alarm, These bits report the slave capabilities to the master and are changed by the MS0 master following its capabilities
27	DPV1_Status_3	[5] 6.2.4.9	• Bit 0-2: - , • Bit 3: Prm Structure, • Bit 4: IsoM_Req, • Bit 5-6: reserved, • Bit 7: Prm_Cmd
28	User_Prm_Data		From Byte 28 of the BMCP, a DTM has to insert additional user parameter data. The information is given in the GSD file via value of User_Prm_Data
...			
NOTE 1 It is possible that a module does not have user parameter data. In this case, no parameter string is inserted into the BMCP for this module. NOTE 2 Some slaves have some fixed modules besides the BIM. Even if these modules do not appear in the configuration dialog or are not represented as Module DTMs, they have to be considered in the BCMP if they have parameters. NOTE 3 The three DPV1 status bytes are defined by the MS0 master.			

6.5.2.2.3 Part 3: Configuration data

The configuration data are provided as part 3:

Table 5 – BMCP Part3 – Configuration data

Byte	Name	Source	Notice
16+Prm_Data_Len	Cfg_Data_Len	[5] 6.2.12.13	Length including this value
16+Prm_Data_Len + 1			
16+Prm_Data_Len +2	Cfg_Data		Configuration data (if available)
...			

From byte position (16+Prm_Data_Len +2), the configuration strings for the BIM and the modules are provided in ascending order. The information is given in the GSD file via value module.

NOTE 1 Some slaves have some fixed modules besides the BIM. Even if these modules do not appear in the configuration dialog or are not represented as Module DTMs, they have to be considered in the BCMP if they have parameters.

NOTE 2 Empty slots have to be considered in the configuration data. Refer to the GSD file as to which configuration string has to be used.

6.5.2.2.4 Part 4: Address table and slave user parameters

In Part 4 of the BMCP, the address table and the slave user parameter section is provided.

Table 6 – Part 4: Address table and slave user parameters

Byte	Name	Source	Hints
16 + Prm_Data_Len + Cfg_Data_Len	Add_Tab_Len	[5] 6.2.12.15	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len	Slave_User_Data_Len	[6] 6.2.12.21	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 1			
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + 2	Slave_User_Data		Slave user parameter (if available)
...			

After Add_Tab_Len the address table data is inserted into the BMCP. If the master does not support this table the Add_Tab_Len is set to 2 and no data is inserted.

After Slave_UserDataLen the slave user data is inserted into the BMCP. If the master does not support this data the Slave_User_Data_Len is set to 2 and no data is inserted.

Table 7 – Part 4: Extended Prm data

Byte	Name	Source	Hints
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len	Ext_Prm_Data_Len	[5] 6.2.12.23	Length including this value
16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len + 1			

After Ext_Prm_Data_Len, the extended prm data is inserted into the BMCP. If the master does not support this data the Ext_Prm_Data_Len is set to 2 and no data is inserted.

6.5.2.2.5 Complete slave bus parameter set

All values are provided from the Slave DTM. It is responsibility of the Slave DTM to be compatible with the Slave GSD. The Master DTM could change some of these initial values sent by Slave DTM if they depend on the capabilities of the master.

Example

Within the GSD file, it is stated that the device supports the Freeze Mode by the keyword "Freeze_Mode_supp". The master sets the value "Freeze_Mode_Req" within the Slave Bus Parameter Set because only the master knows whether it supports this mode.

The following table explains which component is the source of the parameter values ("Parameter provided by"). Some of the values can be changed by the system or by user interaction. For those values it is indicated which component is allowed to change the values ("Configuration by".) If possible, the default values for the parameters are defined ("Default Value").

Table 8 – Complete BMCP

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
0	Slave_Para_Len	Length of the BMCP including this value	Calculated by Slave DTM	-	-
1					
2	SI_Flag	Bit coded. The meaning of the bits:	-	-	-
		Bit 0: reserved	-	-	0
		Bit 1: Extra_Alarm_SAP	Slave DTM shows with this bit that the feature is supported	Master DTM	Extra_Alarm_SAP_sup p within GSD
		Bit 2: DPV1_Data_Types	Slave DTM shows what data types are used by the slave DTM	Master DTM	DPV1_Data_Types within Slave GSD
		Bit 3: DPV1_Supported	Slave DTM shows with this bit that the feature is supported	Master DTM	DPV1_Slave within GSD
		Bit 4: Publisher_enable	Slave DTM shows with this bit that the feature is supported	Master DTM	Publisher_sup within GSD
		Bit 5: Fail_Safe	Slave DTM can show by this flag that the slave needs fail safe (Fail_Safe or Fail_Safe_Required of the GSD)	Master DTM	Value within Slave GSD
3	Slave Type	Bit 6: New_Prm	If the Slave DTM set this value, it requests that the MS0 master should set the changed slave parameter. (see [5] 6.2.12.2)	-	0
		Bit 7: Active	If the Slave DTM has finished the module configuration set it to 1.	-	0
		0 = DP-Slave	Slave DTM shows that it is a DP-Slave	-	0
		Max_Diag_Data_Len	Slave DTM provides this information by Max_Diag_Data_Len in the GSD	-	Value within Slave GSD
5	Max_Alarm_Len	Length of the alarm structure see [5] table 396	Slave DTM, Conditions: One of GSD keywords:- Diagnostic_Alarm_sup - Process_Alarm_sup - Pull_Plug_Alarm_sup - Status_Alarm_sup - Update_Alarm_sup - Manufacturer_Specific_Alarm_sup OR - Alarm_Type_Mode_sup	-	Is calculated on the base of the different GSD values.
6	Max_Channel_Data_Len		Slave DTM: This field defines how much data can be transferred between slave and master. In this case the maximum of these values must be calculated and set by the slave DTM. Rule for calculation: Max_Data_Len C1_Max_Data_Len plus 4 Bytes (Function Num, Slot_Number, Length)	-	Value within Slave GSD
7	Diag_Update_Delay		Slave DTM	-	Slave GSD [6] Diag_Update_Delay

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
8	Alarm_Mode		Slave DTM	-	Slave GSD [6] Alarm_Sequence_Mod e_Count
9	Add_Sl_Flag	Bit coded. Following the meaning of the bits			
		Bit 0: NA_To_Abort		Master DTM	0
		Bit 1: Ignore_Aclr		Master DTM	0
		Bit 2 to Bit 7: reserved			0
10	C1_Timeout	The following GSD value is used [6]: C1_Response_Timeout	Slave DTM		Slave GSD [6] C1_Response_Timeout
11					
12	Reserved				0
13	Reserved				0
14	Reserved				0
15	Reserved				0
16	Prm_Data_Len	Length of the Parameter Data including this value	Calculated (or maybe fixed) by Slave DTM		
17					
	PRM_DATA				
18	Prm_Data (Station Status)	Bit 0-2: reserved		0	
		Bit 3: WD_On	MS0 Master defines that watchdog is used or not. If watchdog is enable the master has also to set WD_Fact_1 and WD_Fact_2	Master DTM	0
		Bit 4: Freeze_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	Freeze_Mode_supp within GSD
		Bit 5: Sync_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	(Sync_Mode_supp within GSD)
		Bit 6,7: Lock/Unlock Request [5] Table 402	Master DTM	0	
19	Prm_Data (WatchDog1) WD_Fact_1	These values depend on the baud rate. A master should set these values and slaves should handle new values. Consider within Usr_Prm_Data (DPV1_Status_1, Bit 2 WD_Base_1ms)		Master DTM	1

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
20	Prm_Data (WatchDog2) WD_Fact_2	See Byte 19, Watchdog_Time = 10 ms * WD_Fact_1 * WD_Fact_2 OR Watchdog_Time = 1 ms * WD_Fact_1 * WD_Fact_2		Master DTM	1
21	Prm_Data (Min Tsdr)	Default value is 11 bit times [1].		Master DTM	11
22	Prm_Data (Ident_Number)		SlavedTM	-	Slave GSD [6] Ident_Number
24	Prm_Data (Group_Ident)	This means the slave is not assigned to a group for global control. It is set by the master		Master DTM	0
25	Prm_Data (DPV1_Status_1)	Bit 0,1: reserved			0
		Bit 2: WD_Base_1ms (See Bytes 19 and 20)	Slave DTM shows with this bit that the feature is supported	Master DTM In some rare case there is a need to set the value to 1	WD_Base_1ms_supp within GSD
		Bit 3 – 5: reserved			0
		Bit 6: Fail_Safe	Slave DTM shows with this bit that the feature is supported	Master DTM	(Fail_safe within GSD)
		Bit 7: DPV1 Enable	Slave DTM	Master DTM	Slave GSD (If the GSD minimum include one of these setting : C1_Read_Write_supp = 1 or Diagnostic_Alarm_supp = 1 or Process_Alarm_supp = 1 or Pull_Plug_Alarm_supp = 1 or Status_Alarm_supp = 1 or Update_Alarm_supp = 1 or Manufacturer_Specific_Alarm_supp = 1 . The slave supports DPV1 and then the DPV1_Enable should be 1

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
26	Prm_Data (DPV1_Status_2)	Bit 0 : Check_Cfg_Mode			
	Slave DTM shows with this bit that the feature is supported.	Master DTM	Check_Cfg_Mode within GSD		
		Bit 1 : reserved	-	-	0
		Bit 2 : Enable_Update_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Update_Alarm_supp within GSD
		Bit 3 : Enable_Status_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Status_Alarm_supp within GSD
		Bit 4: Enable_Manufacturer_Specific_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Manufacturer_Specific_Alarm_supp within GSD
		Bit 5: Enable_Diagnostic_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Diagnostic_Alarm_supp within GSD
27	Prm_Data (DPV1_Status_3)	Bit 6: Enable_Process_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Process_Alarm_supp within GSD
		Bit 7: Enable_Pull_Alarm	Slave DTM shows with this bit that the feature is supported	Master DTM	Pull_Plug_Alarm_supp within Slave GSD
		Bit 0-2: Alarm_Mode	Slave DTM	Master DTM	Data within Slave GSD
		Bit 3: Prm_Structure.	Slave DTM shows with this bit that the feature is supported		Prm_Block_Structure_supp within GSD
		Bit 4: IsoM_Req	Slave DTM shows with this bit that the feature is supported	Master DTM	Isosynchron_Mode_supp within GSD
		Bit 5-6: reserved	-	-	0
		Bit 7: Prm_Cmd	Slave DTM shows with this bit that the feature is supported	Master DTM	PrmCmd_supp within GSD
28 to ...	Prm_Data(Usr_Prm_Data)		Calculated by Slave DTM	Slave DTM	Data within GSD
	Cfg_Data_Len	Length including this value	Calculated by Slave DTM	-	Data within GSD
	X1+1				
	X1+2 to ...	Cfg_Data	Slave DTM, Depending on Module Configuration	Slave DTM	Data within Slave GSD (Module, EndModule)
	X2	Length including this value	Calculated by the Communication DTM		
X2 + 1					

Byte	Name	Note	Parameter provided by	Configuration by	Default Value
X2 + 2 to ...	Add_Tab_Data	Address assignment Table (only for DPV0 Masters)	Calculated by the Communication DTM		
X3	Slave_User_Data_Len	Length including this value	Calculated by the Communication DTM		
X3 + 1					
X3 + 2 to ...	Slave_User_Data	Slave_User_Data	Calculated by the Communication DTM		Data within GSD
X4 + 1	Ext_Prm_Data_Len	Length including this value	Calculated by Slave DTM	-	
X4 + 2					
X4	Ext_Prm_Data	Extended Parameter Data	Slave DTM	-	Data within GSD
NOTE X1 = 16+Prm_Data_Len X2 = 16 + Prm_Data_Len + Cfg_Data_Len = 16 + x1 + Cfg_Data_Len X3 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len = X2 + Add_Tab_Len X4 = 16 + Prm_Data_Len + Cfg_Data_Len + Add_Tab_Len + Slave_User_Data_Len = X3 + Slave_User_Data_Len					

6.5.2.3 Modification of the BMCP

6.5.2.3.1 Propagation of changes

The BMCP includes parameter and configuration data. The slave DTM or the FDT Frame Application may modify the BMCP.

The system has to ensure that the Communication DTM representing the PROFIBUS master is aware of these changes. This is achieved by sending the event `InstanceDataChanged` from the Slave DTM to report the change of the BMCP. All changes should be reported as soon as possible, but not before the changes are persistent. The Frame Application informs the Parent DTM via `OnChildInstanceDataChanged`, that parameters of a child have been changed. Then the Communication DTM can get the new BCMP of the Slave DTM.

6.5.2.3.2 When it is allowed to change the BMCP

According to the FDT specification is allowed to change the parameters of a DTM starting from “configured” state (see IEC 62453-2).

The BCMP can be changed multiple times, but only if the DTM is in offline mode and if the data set can be locked.

If a Slave DTM wants to change parameters in online mode it must use DPV0 or DPV1 transaction requests. If there is no way of changing the parameters by transaction requests, the DTM must disable configuration and parameterization in the online state.

6.5.2.3.3 Parameter data

If the user changes parameters of the BIM or one module (for example via user interface) and this affects the BMCP; the DTM has to update the BMCP. In addition to this, it must request a save and inform the Frame Application via `InstanceDataChanged`. The Frame Application must distribute the information to all relevant components.

6.5.2.3.4 Configuration data

Configuration data will change every time if the user adds/removes modules or changes the module type, etc.

In the case of the modular DTM, the BIM will be informed when the user adds or removes modules via the service `ChildAdded` and service `ChildRemoved`. Changes of the parameters in module will be reported by service `OnChildInstanceDataChanged`.

The monolithic DTM or the BIM DTM update their BMCP, request a save and inform the Frame Application via `InstanceDataChanged`.

NOTE 1 Changes that affect the BMCP often take effect on the internal topology and the Process Channels. This information must be updated by the DTMs too.

NOTE 2 The BMCP may be changed by the Slave DTM and by the Communication DTM. The Communication DTM must not change the configuration data and the user parameters.

6.5.2.4 Special cases related to the BMCP

6.5.2.4.1 DPV1 support

In the GSD file there are two flags regarding DPV1. At first, the “DPV1_Slave” value: This means that the slave has the possibility to work as a DPV1 slave. If this value exists and the value is “1” then the respective bit within the `SI_Flag` of Slave Bus Parameter Set must be set. For older systems, there should also be the possibility to work as a DPV0 slave.

Only the Communication DTM knows that its master device is able to provide acyclic services.

After building the Slave Bus Parameter Set, the Communication DTM will receive the slave's initial BMCP. If, within the SI_Flag, the respective bit is set, the Master DTM must set the highest bit in the first byte of Extended DPV1 Status. Now the slave works as a DPV1 slave.

6.5.2.4.2 Extended DPV1 status

All the bits in the 3 bytes Extended DPV1 Status are set by the Communication DTM for the master. A Slave DTM must accept these settings and adapt its functionality if necessary.

7 Protocol specific usage of general data types

The following table (Table 9) explains how data types, defined in IEC 62453–2 within the namespace 'fdt', are used with CP 3/1 AND CP 3/2 protocols.

Table 9 – Protocol specific usage of general data types

Data type	Description for use in IEC 61784 CPF 3
fdt:address	Profibus Parameter Address: The attribute 'address' follows the different device models that are defined for PROFIBUS devices. FDT currently supports the following models: PROFIBUS DP / DPV1, PROFIBUS PA, PROFIdrive (greater or equal profile version 3) PROFIBUS DP / DPV1 The device model is based on devices that are composed of slots, whereas slots do not have to represent physical objects. The data that is contained in the slots, are addressable via Indexes. This data may be variables or composed blocks of data. The address attribute is APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0' PROFIBUS PA The device is represented by a device management structure and a number of blocks that provide different functionality (physical block, function block, transducer block). The blocks are mapped to slot addresses, but this mapping may vary depending on the device type. The address attribute is APIxxSLOTyyINDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0' PROFIdrive According to the PROFIdrive profile, a device (drive unit) may be composed by a number (1-many) of drive objects (DOs). The DOs may have different type. Each DO is uniquely identifiable and manages its own parameters. Each parameter can be uniquely identified by its number (PNU). Each DO has its own number space. A parameter may contain simple data or composed data (e.g. arrays). The data of the device are accessible via a parameter channel (normally slot 0 index 47). The address attribute is APIxxSLOTyyINDEXzz.DODO-id.pnu xx API yy Slot zz Index

Data type	Description for use in IEC 61784 CPF 3
	do-id Drive Object ID pnu ParameterNumber xx, yy, zz, do-id, pnu are based on decimal format without leading '0'
fdt:protocolId	See Clause 4
fdt:deviceTypeId	The element "fdt:deviceTypeId" shall contain the IDENT_NUMBER of the supported physical device. The IDENT_NUMBER shall be entered in decimal format; however, the value should be displayed as hex to the user
fdt:deviceTypeInformation	A PROFIBUS device has to provide its GSD information as human readable string at this attribute. NOTE The GSD information is accessible via the services NetworkManagementInfoRead GetTypeInformation
fdt:deviceTypeInformationPath	Path to the file containing the information which is provided via the attribute 'deviceTypeInformation'. In case of PROFIBUS the attribute contains the full path to the GSD file including the file name. The file name depends on the current locale according to the usage of service SetLanguage. For PROFIBUS devices it is mandatory to provide this attribute. Examples English: 'C:\MyFolder\ABCD.GSE' German: 'C:\MyFolder\ABCD.GSG'
fdt:manufacturerId	Manufacturer according to Profile specification. For example in Profibus PA : Physical Block Index 10 : DEVICE_MAN_ID
fdt:semanticId fdt:applicationDomain	The SemanticIDs for PROFIBUS follow the different device models that are defined for PROFIBUS devices. FDT currently supports the following models: PROFIBUS DP, PROFIBUS PA, PROFIdrive. PROFIBUS PA The applicationDomain is: FDT_PROFIBUS_PA The device is represented by a device management structure and a number of blocks that provide different functionality (physical block, function block, transducer block). The blocks are mapped to slot addresses, but this mapping may vary depending on the device type. Since the device model is based on blocks, the SemanticIds also are based on the block model. Within each block, the data is identifiable by names of parameters. The semanticId for PROFIBUS profile related parameter follows the following rules: - the semanticId shall be built based on the names defined in the profiles - structured parameters shall be combined with a '.'; - spaces within the profile definition shall be exchanged with an underscore; - blocks shall be counted according to the Object Dictionary; - the block number shall be part of the semanticID. The semanticId is: BlockType.BlockIndex.NameOfParameter.AttributeOfParameter Example AnalogInputFB.3.OUT.Unit PROFIdrive The applicationDomain is: FDT_PROFIBUS_PROFIDRIVE According to the PROFIdrive profile, a device (drive unit) may be composed by a number (1-many) of drive objects (DOs). The DOs may have different types. Each DO is uniquely identifiable and manages its own parameters. Each parameter can be uniquely identified by its number (PNU). Each DO has its own number space. A parameter may contain simple data or composed data (e.g. arrays). The data of the device are accessible via a parameter channel (normally slot

Data type	Description for use in IEC 61784 CPF 3
	0 index 47). The semanticId is: DOdo-id.PNUpnu do-id Drive Object ID pnu ParameterNumber do-id, pnu are based on decimal format without leading '0' Example DO3.PNU64 PROFIBUS DPV1, The applicationDomain is: FDT_PROFIBUS_DPV1 The device model is based on devices that are composed of slots, whereas slots do not have to represent physical objects. The data that is contained in the slots are addressable via Indexes. This data may be variables or composed blocks of data. The semanticID for devices not based on a profile is directly based on the PROFIBUS address information: The semanticId is: APIxx.SLOTyy.INDEXzz xx API yy Slot zz Index xx, yy, zz are based on decimal format without leading '0'
fdt:subDeviceType	Manufacturer specific value

8 Protocol specific common data types

Not applicable.

9 Network management data types

9.1 General

The data types specified in this subclause are used at following services:

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

The data type net:DeviceAddress (defined in IEC 62453–2) is used for defining the network address of a device.

The protocol specific data types are based on definitions given in the IEC 61784 and IEC 61158 specifications. Furthermore, they contain additional information about the device that is needed by systems to configure CPF 3 links and to establish communication between the CPF 3 master device and the CPF 3 slave devices.

9.1.1 Configuration

The configuration of the device itself is done with the aid of the DTM's GUI. Downloading the configuration into the slave device is performed via the CPF 3 master device. To do that and in order to set up the bus communication, the master needs information from the DTM as there is:

- GSD file
The GSD information is type-specific information and not instance-specific. It is not stored with single slave instances or in a global accessible file. It is provided by the DTM at the

service `GetTypeInformation`. On this service, a DTM of a PROFIBUS device provides the GSD information within its XML document.

The master device can use the general type-specific information from the slave's GSD information like bus timing parameters, supported baud rates, etc.

- **CFG-String (Cfg_Data)**

The CFG-String provides the instance-specific information about the current configuration of the device. It defines the structure of the data frames that will be transmitted on the PROFIBUS. This structure depends on the modules that are actually configured.

The DTM provides the CFG-String within the attribute `busMasterConfigurationPart` that is part of the XML document available via service `InstanceDataRead`. The structure of the `busMasterConfigurationPart` is defined according to the PROFIBUS-DP-Slave-Bus-Parameter-Set (see IEC 62453-2 and also IEC 61158 series).

The master device uses this information to set up communication with the slave device.

9.1.2 Process Channel

In case of CPF 3 protocols, an FDT-Channel is a representative for a single data or a process value that can be accessed from a Frame Application via the master device. The information available at services for IO related information describes how to access a channel via a PROFIBUS DPV1 command or how to address a channel within a PROFIBUS DP frame for cyclic I/O. Besides all mandatory elements (which include `id` and `dpAddress`) it is highly recommended that the DPV1 address information is provided. This information (DPV1 Slot) is used by some frames to manage the PROFIBUS device module information.

In a DPV0 environment, depending on the situation, the underlying master device may have either Master Class 1 functionality or Master Class 2 functionality. A Class 1 master can write output data to a device and control data exchange, where a Class 2 master can only read the output data. Generally, it is assumed that parameterization as described here is performed as a master Class 2 station.

9.1.3 Parameterization

There are two options to write parameters set from the DTM's GUI to the CPF 3 slave device in the field:

- **User Parameters**

User Parameters are part of the PROFIBUS-DP-Slave-Bus-Parameter-Set. They contain manufacturer-specific data to characterize the DP-Slave. The DTM writes the User Parameters to the `busMasterConfigurationPart`. The User Parameters are stored with the master device during PROFIBUS master configuration and are automatically sent to the slave during set up of bus communication. (This is PROFIBUS-specific; for details, see IEC 61158 series.) When changing User Parameters on runtime, the DTM shall use a DP V0 connection and the appropriate DP V0 commands for parameter exchange as described in the data types.

- **Writing Parameters with DP-V1 services (MSAC2 services)**

The DTM may use DP-V1 transport services to send its parameters to the slave device. For that, it has to use a DP-V1 connection and the corresponding communication commands. During setup of communication, DP-V1 services are not sent automatically. The Frame Application or a DTM shall invoke a download of parameters via DP-V1.

For details on the different behavior of slaves depending on the kind of parameterization, refer to the IEC 61158 series.

DP-V1 connections and communication commands can also be used to execute commands at the slave. For details on the use of DP-V1, see also the IEC 61158 series.

9.2 Master-bus parameter set

The following parameter set represents the contents of the data type busMasterConfigurationPart within the device instance data for Profibus master device (Table 10). This attribute has to be set for each Profibus master device according to the IEC 61158 series. For further details, please refer to the sequence chart “Configuration of a Fieldbus Master” and the IEC 61158 series.

Table 10 – Bus parameter set for master device

Name	Type	Comment
Bus_Para_Len	UINT	Length of the bus parameter set inclusive, the length parameter (range 34 to 216-1)
FDL_Add	USINT	Mandatory part of the Bus Parameter Set according to PROFIBUS-DP Specification; not used in this context as station address is transferred in a separate variable; FDL-Add may differ from real station address used by EE and DTMs
Baud_rate	USINT	Code number for the baud rate
TSL	UINT	Slot time
min TSDR	UINT	Minimum station delay response
max TSDR	UINT	Maximum station delay response
TQUI	USINT	Quit-time
TSET	USINT	Setup-time
TTR	UDINT	Target rotation time
G	USINT	GAP update factor
HSA	USINT	Highest station address
max_retry_limit	USINT	Maximum retry limit
Bp_Flag	USINT	Flags for the user interface, e.g. error action flag
Min_Slave_Interval	UINT	Smallest allowed time period between two slave poll cycles
Poll_Timeout	UINT	Master-master timeout
Data_Control_Time	UINT	Guaranteed time period between two Data_transfer_list updates
Octet 1 (reserved)	ARRAY[6] of USINT	
...		
Octet 6 (reserved)		
Master_User_Data_Len	UINT	Length of Master_User_Data inclusive length parameter
Master_Class2_Name	ARRAY (32) of CHAR	Name of DP class 2 master the parameter set was created with
Master_User_Data	ARRAY of USINT	Manufacturer specific

9.3 Slave bus parameter set

The following parameter set represents the contents of the data type busMasterConfigurationPart within the instance data set for Profibus slave devices (Table 11). This attribute has to be set for each Profibus slave device according to the IEC 61158 series. For further details, please refer to the sequence chart “Configuration of a Fieldbus Master” and the IEC 61158 series.

Table 11 – Bus parameter set for slave device

Name	Type	Comment
Slave_Para_Len	UINT	Length of the slave parameter set inclusive the length parameter
Sl_Flag	USINT	Slave specific flags like New_Prm, Active, Fail_Safe, ...
Slave_Type	USINT	Manufacturer specific slave type denotation (0 by default for DP-Slaves)
Octet 1 (reserved)	ARRAY[12] of USINT	
...		
Octet 12 (reserved)		
Prm_Data_Len	UINT	Length of Prm_Data inclusive the length parameter (range 9 to 246)
Prm_Data	ARRAY of USINT	
Cfg_Data_Len	UINT	Length of Cfg_Data inclusive the length parameter (range 3 to 246)
Cfg_Data	ARRAY of USINT	
Add_Tab_Len	UINT	Length of Add_Tab inclusive the length parameter (range 2 to 216-31)
Add_Tab	ARRAY of USINT	Address assignment table
Slave_User_Data_Len	UINT	Length of Slave_User_Data inclusive the length parameter (range 2 to 216-31)
Slave_User_Data	ARRAY of USINT	Manufacturer specific data to characterize the DP-Slave for the master

9.4 Module and channel data

A slave's current module configuration including the belonging channels has to be available by FdtChannel objects. That is required as within the environment process variables have to be assigned to single channels and the slave configuration has to be displayed in the environment's system overview without using the DTM's user interface.

The addressing of channels within the PROFIBUS data frame is bit oriented and data type independent. A bit position and a bit length determine each channel. The data type of a parameter determines how to convert the bit field.

- Channels within PROFIBUS data frames (Figure 3):

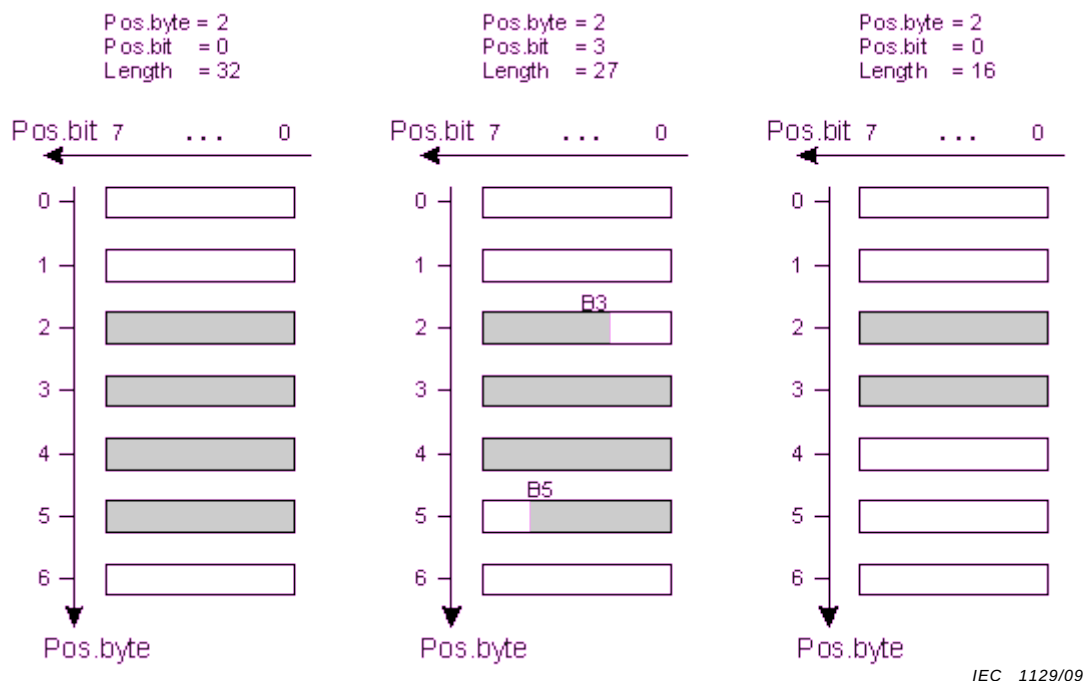


Figure 3 – Example for IO data within datagrams

- Representation in FDT Notation:

bitPosition="16"

bitPosition="19"

bitPosition="16"

bitLength="32"

bitLength="27"

bitLength="16"

The following example shows the expected structure description of a modular slave in a DTM parameter set using FDT-Parameters with the object types defined in the FDT-Specification.

- Representation of the data frame using GSD information:

BeginSlave;;

RIOLB8101.GSD;6;

BeginModules;;

1X03 FrequencyCount;0x51;

Empty ;0x00;

Empty ;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

Empty;0x00;

2XXX ValveBlock.;0x30;

EndModules;;

EndSlave;;

Description of the position of signal channels within the data frame ():

Table 12 – Signal channels within the data frame

# Frame description Remote I/O									
Vendor:		"RIO manufacturer"							
Devicetype :		"LB 8101"	Device ident:		0x8101		# hex		
Frameident:		"B1t1"							
Input data									
Channel	Channel type	Data	Data	Data	Invalid Bit	Invalid Bit	Subst. Value Bit	Subst. Value Bit	State-Channel
		Pos. byte	Pos. bit	Length	Pos. byte	Pos. bit	Pos. byte	Pos. bit	
Module: "EP01"									
"Digital Input (Counter) LB/FB 1X03 (4 Byte)"									
"Count_1_0"	READ INT	0	0	32	-/-	-/-	-/-	-/-	0
Module: "EP08"									
"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input and Output)"									
"DI_8_LFOU0"	READ BOOL	4	1	1	-/-	-/-	-/-	-/-	1
"DI_8_LF1"	READ BOOL	4	3	1	-/-	-/-	-/-	-/-	1
"DI_8_LF2"	READ BOOL	4	5	1	-/-	-/-	-/-	-/-	1
Output data									
Channel	Channel type	Data	Data	Data	Invalid Bit	Invalid Bit	Subst. Value Bit	Subst. Value Bit	State-Channel
		Pos. byte	Pos. bit	Length	Pos. byte	Pos. bit	Pos. byte	Pos. bit	
Module: "EP08"									
"Valve block LB/FB 2XXX / 1 DO / 2 DI (1 Byte Input u. Output)"									
"DO_8_0"	WRITE BOOL	0	0	1	4	1	-/-	-/-	0

9.5 GSD information

9.5.1 General

The GSD information is not stored with single slave instances or in a global accessible file. It is provided by the DTM via the service `GetTypeInformation`. A DTM of a PROFIBUS device provides the GSD information within the `fdt:DtmDeviceType` argument.

The existing DCS uses the GSD information to obtain information about the possible configuration and parameters of a DP Slave.

Besides the information about modules and its parameters, a GSD file contains additional information about the slave, such as the supported baud rates.

This information is useful for a DCS system to configure an entire network according to the capabilities of different slaves.

9.5.2 GSD for gateway devices

9.5.2.1 Types of Profibus gateway devices

There are two types of gateway devices.

The visible gateway devices work as a PROFIBUS slave (with a PROFIBUS address) and to the underlying network they act as a master.

The invisible gateway devices just transform the PROFIBUS network to the underlying network.

For both types of devices there is a need for special GSDs.

9.5.2.2 Visible gateway devices

These visible gateway devices are shipped without a GSD file. Instead they have a proprietary software suite that configures and parameterizes it or they are shipped with a tool that creates a GSD for parameterization software. The GSD tool creates a GSD for the slave depending on the underlying network configuration or bus settings (for example baud rates).

In these cases the DTM should provide the functionality of the GSD tool. If the GSD is configuration-dependent, it could call service `NetworkManagementInfoRead` for each of its children, extract the GSD information and create configuration-dependent GSD information in the same way the tool does. After initialization of the DTM, it should deliver a BMCP according to the linking device itself. Every time a child is added or removed, this leads to a change in its own parameter document.

If the GSD depends on bus settings, a DTM's configuration or parameterization dialog could be used to change bus settings. Based on these settings, updated GSD information can be inserted in the information document. Here too the DTM has to request a save and call service `OnInstanceDataChanged`.

Note that the internal structure with its modules and channels has to be updated as well.

9.5.2.3 Transparent gateway devices

There are transparent linking devices on the market (PROFIBUS FMS/DP and PROFIBUS PA) performing a baudrate transformation. This requires a special handling of the slave specific GSD files. There are tools available which are able to adapt existing GSD files according to the higher baudrate (so called 'GSD Converter').

The GSD information must be delivered by the DTM for the device. In order to support this kind of linking devices, a slave DTM must expose the GSD file on hard drive using the attribute 'deviceTypeInformationPath' (refer to DtmDeviceType defined in IEC 62453-2).

A DTM developer should be aware, that the referenced GSD file might be converted in order to support specific system architecture.

It is expected that a Profibus DTM in the attribute 'deviceTypeInformation' is exposing exactly the GSD file which is referenced by the attribute 'deviceTypeInformationPath'.

10 Communication data types

10.1 General

The data types defined here are used at:

- Connect service;
- Disconnect service;
- Transaction service.

10.2 Error information provided by Communication Channel

In every Response data type of FDT Profibus specification an element 'errorCode' is provided. According to PROFIBUS, the error code is standardized to consist of 3 bytes, where each byte carries a meaning.

Since the error code is exchanged between different DTMs (e.g. Communication-DTM and Device-DTM) and since the receiver of the error code will try to understand the error code, the provider shall use the standard format:

- standard length 3 bytes;
- if the device provides error codes, these error codes are provided (and not local error codes from the Master).

If no error occurred, the element 'errorCode' shall be filled with zeros ("000000").

10.3 DPV0 communication

The supported services depend on the type of PROFIBUS master functionality that is provided by the communication infrastructure. Master Class 1 devices typically control the slaves and provide cyclic communication, Master Class 2 devices typically are used to configure the slaves and provide acyclic communication.

Not all defined services are supported if the Master is not in cyclic data exchange with the slaves. In such cases, the following behavior is expected:

If a Communication Channel receives a request that can not be supported, the Transaction service responds with a result="false".

The data types described in this clause are defined for the following namespace:

Namespace: dpv0

Table 13 – Simple DPV0 communication data types

Data type	Definition	Description
busAddress	USINT	Address information according to the IEC 61158 series (see also IEC 62453-2, attribute busAddress)
communicationReference	UUID	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during creation of the connection. The address information has to be used for all following communication calls
connectStatus	enumeration (masterConnectedOnly deviceAtLifeList deviceInDataExchange)	Describes the connection status established by the communication component. The status "masterConnectedOnly" means that the communication component has established a connection to the Profibus master device and will accept an online access to the user parameters, independent of whether the device is available or not. The status "deviceAtLifeList" means that the communication component has established a connection to the Profibus master device and has checked that the device is in the life list of the master stack. In this state, the master will accept an online access to the user parameters and will send the user parameter to the device, independent of whether the device is in data-exchange or not. The status "deviceInDataExchange" means that the communication component has established a connection to the Profibus master device and has checked that the device is in data-exchange. In this state, the master will accept an online access to the user parameters and will send the user parameter to the device, so that the new data will directly influence the process
delayTime	UDINT	Delay time in [ms] between two communication calls
errorCode	ARRAY OF USINT	Status information according to the IEC 61158 series. For description of error code see: DIN 19245 Part 3, PROFIBUS (P. 40ff., 83ff., 39)
schemaVersion	INT	Defines the version of the schema
sequenceTime	UDINT	Period of time in [ms] for the whole sequence

Table 14 – Structured DPV0 Communication data types

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
Abort	STRUCT			Describes the abort	-
	communicationReference	O	[0..1]		
ConnectRequest	STRUCT			Describes the communication request to establish a connection to a Profibus master device. Depending on the used network infrastructure it is possible that this service is not mapped to a field bus request, but is used to manage the software. It is also possible that this service is used as a trigger to set the state of a device	-
	busAddress	M	[1..1]		
	fdt:systemTag	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ConnectResponse	STRUCT			Describes the communication response to the connect request and provides the information how the following WriteUserParameter commands will be sent to the device (see connectStatus). The resulting connection depends on the used communication device and on the configuration of the bus master. The DeviceDTM shall expect that the resulting connection cannot be used to access all services. ConnectResponse may return true even if there is no communication with the device possible. For example if ConnectRequest returns true with connectionStatus = "masterConnectedOnly" this means that there is a connection to the master device, but the device is not at the life list and can therefore not be accessed. Depending on the connectStatus some services may not be available (e.g. in status "masterConnectedOnly" only services provided by the master are accessible (ReadUserParameter, WriteUserParameter, ...) whereas services provided by the device are not accessible. NOTE In this example the DTM is in state "online" but it cannot access the device. The provided services also depend on the type of master device. For an overview of availability of services depending on the type of connection, see the following tables.	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
	connectStatus	M	[1..1]		
DisconnectRequest	STRUCT			Describes the communication request to release a connection to a Profibus master device	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		
DisconnectResponse	STRUCT			Describes the communication response	-
	busAddress	M	[1..1]		
	communicationReference	M	[1..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ReadDiagnosis→Data→Request	STRUCT			Describes the communication request according to the Profibus DPV0 specification For Master Cl.1 refer to IEC 61158-5, 8.2.3.3.2 “Get slave diag” For Master Cl.2 refer to IEC 61158-5, 8.2.3.3.3 “Read slave diag”	1, 2
	communicationReference	M	[1..1]		
ReadDiagnosis→Data→Response	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadInput Data→Request	STRUCT			Describes the communication request according to the Profibus DPV0 specification. Depending on whether the underlying FDTChannel is a Master Cl. 1 or a Master Cl. 2 this service will be local or result in a read access to the slave device. It is possible to specify which data is read by providing an fdt:ChannelReference. For Master Cl.1 refer to IEC 61158-5, 8.2.2.3.3 “Get Input” For Master Cl.2 refer to IEC 61158-5, 8.2.2.3.2 “Read Input”	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadInput Data→Response	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
ReadOutputData~Request	STRUCT			Describes the communication request according to the Profibus DPV0 specification. Depending on whether the underlying FDTChannel is a Master Cl. 1 or a Master Cl. 2 this service will be local or result in a read access to the slave device. It is possible to specify which data will be read by providing an fdt:ChannelReference. For Master Cl. 1 this is a proprietary service (or unavailable). For Master Cl. 2 refer to [1], section 8.2.2.3.6. "Read Output" (an optional service)	1, 2
	communicationReference	M	[1..1]		
	fdt:ChannelReference	O	[0..1]		
ReadOutputData~Response	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		
ReadUserParameter~Request	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The request retrieves only the information that is available from the master device (local service). It may differ from the actual data of the device. The implementation on how to provide the data is proprietary. This is based on a local service or not available	1
	communicationReference	M	[1..1]		
ReadUserParameter~Response	STRUCT			Describes the communication response, read from the Profibus master device, according to the Profibus DPV0 specification. The returned data may reflect only the information that is available from the master device. It may differ from the actual data of the device	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		
	fdt:CommunicationData	O	[0..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
SequenceBegin	STRUCT			Describes the sequence begin	-
	sequenceTime	O	[0..1]		
	delayTime	O	[0..1]		
	communicationReference	M	[1..1]		
SequenceEnd	STRUCT			Describes the sequence end	-
	communicationReference	M	[1..1]		
SequenceStart	STRUCT			Describes the sequence start	-
	communicationReference	M	[1..1]		
WriteOutputData~Request	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The output data will be sent according to the established connection. It is necessary to specify which data will be written by providing an fdt:ChannelReference. If the underlying FDTChannel is provided as Master Cl. 2, writing Output Data will be not possible. Refer to IEC 61158-5, 8.2.2.3.5 "Set Output". This functionality depends on the used type of master. In terms of PLC this functionality is named "forcing" of values	1
	communicationReference	M	[1..1]		
	fdt:ChannelReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteOutputData~Response	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Data type	Definition			Description	Master Class
	Elementary data type	Usage	Multiplicity		
WriteUserParameterRequest	STRUCT			Describes the communication request according to the Profibus DPV0 specification. The user parameter will be sent according to the established connection (see connectStatus). For Master Cl. 1 this is based on the service "DDL_M_Set_Prm". For Master Cl. 2 this is based on the service "DDL_M_Set_Prm" (an optional service)	1, 2
	communicationReference	M	[1..1]		
	fdt:CommunicationData	M	[1..1]		
WriteUserParameterResponse	STRUCT			Describes the communication response according to the Profibus DPV0 specification	-
	communicationReference	M	[1..1]		
	errorCode	M	[1..1]		

Depending on the bus master type and on the returned connectStatus following services are available.

Table 15 – Availability of services for Master Class1 (C1)

Slave DTM Service Request	connectStatus		
	masterConnectedOnly	DeviceAtLifeList	DeviceInDataExchange
Connect	✓	✓	✓
ReadUserParameter	O	O	O
WriteUserParameter	✓	✓	✓
ReadOutputData			✓
WriteOutputData			O
ReadInputData			✓
ReadDiagnosisData		✓	✓
NOTE ✓ : the service is available, O : the service is optional and may be available, depending on the capabilities of the underlying master device.			

For Master Class2 (C2), not all connect states are available:

Table 16 – Availability of services for Master Class2 (C2)

Slave DTM Action	connectStatus	
	DeviceAtLifeList (no DPV1 connection to device)	DeviceInDataExchange (DPV1 connection to device)
Connect	✓	✓
ReadUserParameter		
WriteUserParameter	O	O
ReadOutputData		O

Slave DTM Action	connectStatus	
	DeviceAtLifeList (no DPV1 connection to device)	DeviceInDataExchange (DPV1 connection to device)
WriteOutputData		
ReadInputData		✓
ReadDiagnosisData	✓	✓
NOTE ✓ : the service is available, O : the service is optional and may be available, depending on the capabilities of the underlying master device.		

If the Master Class 2 communication component supports DPV1 and DPV0 and has established a DPV1 connection to the device, a call to service Connect for DPV0 shall return the status "DeviceInDataExchange" even if the device is not in status DataExchange.

If no DPV1 connection is established, the Master Class 2 communication component shall verify the availability of the device (at least by service LifeList).

10.4 DPV1 communication

The data types are defined to provide the address information and the communication data required for DPV1 communication (see Table 17 and Table 18).

The data types described in this subclause are defined for the following namespace:

Namespace: dpv1

Table 17 – Simple DPV1 communication data types

Data type	Definition	Description
api	USINT	Address information according to the IEC 61158 series. If the device needs special values for the DPV1-Initiate, the DeviceDTM is responsible for providing the values in the ConnectRequest. If the values are not provided, the communication component will use default values (0)
busAddress	USINT	Address information according to the IEC 61158 series (see also DTMPParameterSchema, attribute busAddress)
errorCode	UUID	Status information according to the IEC 61158 series. For description of error code see: PROFIBUS Nutzerorganisation e.V., PROFIBUS Guideline, Order-Nr. 2.0082, Technical Guideline, PROFIBUS – DP Extensions to EN 50170 (DPV1), Version 2.0, April 1998. (P.. 120-121); .For description of abort information see: DIN 19245 Part 3, PROFIBUS, (P..54ff., 11)
index	UDINT	Address information according to the IEC 61158 series
communicationReference	ARRAY OF USINT	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during the connect. The address information has to be used for all following communication calls
delayTime	USINT	Minimum delay time in [ms] between two communication calls

Data type	Definition	Description
maxLenDataUnit	USINT	Optional attribute, to describe the amount of data, which can be transferred via the established connection. If this attribute is not available, no special length restriction is announced. Each communication component within the chain of interfaces concerning nested communication could introduce this attribute. Each communication component should change the contents of the attribute based on the following rule: The new value is the minimum of the current value and the restriction of its own implementation. If a communication component has no restriction, it should hand over the given value. If a communication component is able to reuse an established connection concerning a new connect request, it should take into account the data length determined for the existing connection. If the data length is not applicable, the DTM should send an error message via service ErrorMessage
networkMACAddress	ARRAY OF USINT	Network or MAC address as described in the IEC 61158 series. [5] If there is a networkMACAddress, (i.e. networkMACAddress <> ""), the Profibus-DPV1-Master-DTM shall set the corresponding "address type" in the Initiate request telegram "1", else "0". Example Device with address 20 (14hex) addressed via a linking device: networkMACAddress="00000000000014"
scl	INT	Access level as described in the IEC 61158 series. [5] If the device needs special values for the DPV1-Initiate, the DeviceDTM is responsible for providing the values in the ConnectRequest. If the values are not provided, the communication component will use default values (0)
sequenceTime	USINT	Period of time in [ms] for the whole sequence
slot	UDINT	Address information according to the IEC 61158 series

Table 18 – Structured DPV1 communication data types

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
Abort	STRUCT			Describes the abort
	communicationReference	O	[0..1]	
ConnectRequest	STRUCT			Describes the communication request
	api	M	[1..1]	
	busAddress	M	[1..1]	
	fdt:systemTag	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
ConnectResponse	STRUCT			Describes the communication response
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	maxLenDataUnit	O	[0..1]	
	RedundantAddresses	O	[0..1]	
	SrcNetworkAddress	O	[0..1]	
	DestNetworkAddress	O	[0..1]	
DestNetworkAddress	STRUCT			Describes the extended address of the destination [5]
	NetworkAddress	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response
	api	M	[1..1]	
	busAddress	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
NetworkAddress	STRUCT			Describes the extended address format
	api	M	[1..1]	
	scl	M	[1..1]	
	networkMACAddress	M	[1..1]	
ReadRequest	STRUCT			Describes the communication request
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
ReadResponse	STRUCT			Describes the communication response
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
RedundantAddresses	STRUCT			Describes the redundant addresses
	fdtparam:SlaveAddress	M	[1..*]	
SequenceBegin	STRUCT			Describes the sequence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	

Data type	Definition			Description
	Elementary data types	U s a g e	Multi- plicity	
SequenceEnd	STRUCT			Describes the sequence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Describes the sequence start
	communicationReference	M	[1..1]	
SrcNetworkAddress	STRUCT			Describes the extended address of the source
	NetworkAddress	M	[1..1]	
WriteRequest	STRUCT			Describes the communication request
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	M	[1..1]	
WriteResponse	STRUCT			Describes the communication response
	slot	M	[1..1]	
	index	M	[1..1]	
	communicationReference	M	[1..1]	
	errorCode	M	[1..1]	

Within a connect request, a DTM of a Profibus redundant slave can provide additional redundant slave addresses, set within its parameter document. The busAddress attribute is to be used as the preferred address; the addresses in the redundantAddresses element should be used in this order if an alternative address is used to connect to the redundant slave.

11 Channel parameter data types

The following table (Table 19) describes the mapping of DPV1 Data types to FDT data types.

Table 19 – Mapping of DPV1 data types to FDT data types

PROFIBUS data types	FDT data type
Boolean	BOOL
Integer8	SINT
Integer16	INT
Integer32	DINT
Unsigned8	USINT
Unsigned16	UINT
Unsigned32	UDINT
Floating Point	FLOAT
Visible String	STRING
Octet String	ARRAY of USINT
Date	DATE
Time of Day	TIME
Time Difference	duration
DS-32	structured

PROFIBUS data types	FDT data type
DS-33	structured
DS-34	structured
DS-35	structured
DS-36	structured
...	structured
DS-52	structured
..	structured

The format of all data types is defined in the IEC 61158 series.

If only a part of the retrieved data is process data (e.g. for DS-33 the data consist of the float-value and a byte status), the access information (DPAAddress or DPV1Address) also contains information that is necessary to retrieve the information from the data structure.

Used at ReadChannelInformation service and WriteChannelInformation service.

The Channel **P**parameter data types describe a how to access a channel via a Profibus DPV1 command or how to adress a channel within a Profibus DP frame for cyclic I/O. The following tables (Table 20 and Table 21) provide a description of the data types.

The data types described in this clause are defined for the following namespace:

Namespace: profichannel

Table 20 – Simple Channel Parameter data types

Data type	Definition	Description
api	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
bitLength	UDINT	Additional data type information especially for fieldbus specific data types like 12 bit integer
bitPosition	UDINT	Address information according to the IEC 61158 series for channels accessible via Profibus DP
frameApplicationTag	STRING	Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces
gatewayBusCategory	UUID	Unique identifier for a supported bus type like Profibus or HART according to the FDT specific CATID
index	USINT	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
invalidBit	UDINT	Bit position of the invalid status channel accessible via Profibus DP
logic	UDINT	Additional data type information: positive 0=FALSE 1=TRUE
number	BOOL	Address information for diagnosis according to the IEC 61158 series for channels accessible via Profibus DP. See [5], 9.3.1 Read DP – Slave Diagnostic Information
protectedByChannelAssignment	INT	TRUE if the channel is set to read only by the Frame Application. Usually set to TRUE if a channel assignment exists
schemaVersion	UDINT	Defines the version of the schema
simulationBit	USINT	Bit position of the simulation status channel accessible via Profibus DP
slotNumber	BOOL	Address information according to the IEC 61158 series for channels accessible via Profibus DPV1
statusChannel	UDINT	TRUE if the channel is for status information only
substituteValueBit	STRING	Bit position of the substitute status channel accessible via Profibus DP

Table 21 – Structured Channel Parameter data types

Data type	Definition			Description
	Elementary data type	U s a g e	Multi- plicity	
DpAddress	STRUCT			Address information according to the IEC 61158 series for channels accessible via Profibus DP
	bitPosition	M	[1..1]	
	bitLength	M	[1..1]	
DpV1Address	STRUCT			Address information according to the IEC 61158 series for channels accessible via Profibus DPV1. It is highly recommended to provide this address information to enable assignment of logical addresses to a profibus module
	api	M	[1..1]	
	slotNumber	M	[1..1]	
	index	M	[1..1]	
	bitPosition	O	[0..1]	
	bitLength	O	[0..1]	
FDTChannel	STRUCT			Description of the channel
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannel-Assignment	M	[1..1]	
	number	M	[1..1]	
	fdt:dataType	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	appld:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	DpAddress	O	[0..1]	
	DpV1Address	O	[0..1]	
	StatusInformation	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
	fdt:StructuredElements	O	[0..1]	