PUBLICLY
AVAILABLE
SPECIFICATION

# IEC PAS 62453-3

Pre-Standard

First edition
2006-05



## Field Device Tool (FDT) interface specification –

## Part 3:
## PROFIBUS communication

Reference number
IEC/PAS 62453-3:2006(E)

## Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

## Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

## Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

PUBLICLY
AVAILABLE
SPECIFICATION

# IEC PAS 62453-3

Pre-Standard

First edition
2006-05



## Field Device Tool (FDT) interface specification –

## Part 3:
## PROFIBUS communication



International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE X

*For price, see current catalogue*

# CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

# Field Device Tool (FDT) interface specification –

# Part 3: PROFIBUS communication

## 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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
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.

A PAS is a technical specification not fulfilling the requirements for a standard but made available to the public.

IEC-PAS 62453-3 has been processed by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

| The text of this PAS is based on the following document: | This PAS was approved for publication by the P-members of the committee concerned as indicated in the following document |
|---|---|
| **Draft PAS** | **Report on voting** |
| 65C/398A/NP | 65C/411/RVN |

Following publication of this PAS, which is a pre-standard, the technical committee or subcommittee concerned will transform it into an International Standard.

This PAS shall remain valid for an initial maximum period of three years starting from 2006-05. The validity may be extended for a single three-year period, following which it shall be revised to become another type of normative document or shall be withdrawn.

IEC 62453 consists of the following parts under the general title *Field Device Tool (FDT) interface specification:*

Part 1: Concepts and detailed description

Part 2: INTERBUS communication

Part 3: PROFIBUS communication

Part 4: HART communication

Part 5: FOUNDATION FIELDBUS communication

# INTRODUCTION

This PAS is an interface specification for developers of FDT 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 shall inter-operate seamlessly.

With the integration of fieldbusses into control systems, there are a few other tasks which must be performed. This applies to fieldbusses in general. Although there are fieldbus- and device-specific tools, there is no unified way to integrate those 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 kind of fieldbusses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

# Field Device Tool (FDT) interface specification –

# Part 3: PROFIBUS communication

## 1 Scope

This part of IEC 62435 provides information for integrating the PROFIBUS protocol into the FDT interface specification (IEC 62453-1). This PAS neither contains the FDT specification nor modifies it.

## 2 Normative references

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

IEC 61158:2003 (all parts), *Digital data communications for measurement and control – Fieldbus for use in industrial control systems*

IEC 61158-2:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 2: Physical layer specification and service definition*

IEC 61158-5:2003, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 5: Application layer service definition*

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

## 3 General

### 3.1 PROFIBUS schema

PROFIBUS schemas are required to define the structure and semantics of the protocol-specific data transferred via XML documents on the FDT interfaces.

The schemas are based on definitions given in the PROFIBUS-Specification. Furthermore, they contain additional information about the device that is needed by systems to configure PROFIBUS links and to establish communication between the PROFIBUS master device and the PROFIBUS slave devices.

#### 3.1.1 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 PROFIBUS 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 IDtmInformation. On method GetInformation(), 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 IDtmParameter::GetParameters(). The structure of the busMasterConfigurationPart is defined according to the PROFIBUS-DP-Slave-Bus-Parameter- Set (see Bibliography and also IEC 61158 series).
  The master device uses this information to set up communication with the slave device.

### 3.1.2 Channels

In case of PROFIBUS, an FDT channel is a representative for a single date or a process value that can be accessed from a Frame Application via the master device. The XML document available at IFdtChannel 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 includes id and dp address) it is highly recommended that the XML document provides DPV1 address information. 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.

### 3.1.3 Parameterization

There are two options to write parameters set from the DTM's GUI to the PROFIBUS 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 the setup of bus communication. (This is PROFIBUS-specific; for details, see IEC 61158 series.)When changing User Parameters on runtime, the DTM must use a DP-V0 connection and the appropriate DP-V0 commands for parameter exchange as described in the XML schemas.
- 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 the setup of communication, DP-V1 services are not sent automatically. The Frame Application or a DTM must 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 IEC 61158 series.

## 4 Provided data

### 4.1 Interface IDtmParameter

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

### 4.2 SingleDataAccess interfaces

Via the interfaces IDtmSingleDeviceDataAccess and IDtmSingleInstanceDataAccess at least all parameters of the Physical Block and the status and Out value of the Function Blocks must be exposed.

## 5 Protocol specific usage of XML attributes

Table 1 explains how attributes in documents of the SingleDataAccess interfaces are used with PROFIBUS protocols.

**Table 1 – SingleDataAccessSchema attributes**

| Attribute | Description for use in Profibus |
|---|---|
| address | Profibus Slave Address:<br><br>The attribute 'address' (defined in FDTDataTypesSchema.xml) follow the different device models that are defined for PROFIBUS devices. FDT currently supports following models:<br><br>PROFIBUS DP / DPV1,<br><br>PROFIBUS PA,<br><br>PROFIdrive (greater or equal profile version 3)<br><br>**PROFIBUS DP / DPV1**<br><br>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.<br><br>The address attribute is APIxxSLOTyyINDEXzz<br><br>xx API<br><br>yy Slot<br><br>zz Index<br><br>xx, yy, zz are based on decimal format without leading '0'<br><br>**PROFIBUS PA**<br><br>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.<br><br>The address attribute is APIxxSLOTyyINDEXzz<br><br>xx API<br><br>yy Slot<br><br>zz Index<br><br>xx, yy, zz are based on decimal format without leading '0'<br><br>**PROFIdrive**<br><br>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.<br><br>A parameter may contain simple data or composed data (e.g. arrays).<br><br>The data of the device are accessible via a parameter channel (normaly slot 0 index 47).<br><br>The address attribute is APIxxSLOTyyINDEXzz.DOdo-id.pnu |

| Attribute | Description for use in Profibus |
|---|---|
| | xx API<br>yy Slot<br>zz Index<br>do-id Drive Object ID<br>pnu ParameterNumber<br>xx, yy, zz, do-id, pnu are based on decimal format without leading '0' |
| busCategory | See Clause 6 |
| deviceTypeId | The attribute "fdt:DtmDeviceType/@deviceTypeId" must contain the IDENT_NUMBER of the supported physical device. The IDENT_NUMBER must be entered in decimal format, however, the value should be displayed as hex to the user |
| deviceTypeInformation | A PROFIBUS device has to provide its GSD information as human readable string at this attribute<br><br>NOTE The GSD information is accessible via<br>IDtmParameter::GetParameters()<br>IDtmInformation::GetInformation() |
| deviceTypeInformationPath | Path to the file containing the information which is provided via the attribute 'deviceTypeInformation'.<br>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 IDtm::SetLanguage().<br>For PROFIBUS devices it is mandatory to provide this attribute.<br><br>Examples:<br>English: 'C:\MyFolder\ABCD.GSE'<br>German: 'C:\MyFolder\ABCD.GSG' |
| manufacturerId | Enter manufacturer according to Profile specification, for example in Profibus PA: Physical Block Index 10 : DEVICE_MAN_ID |
| semanticId<br>applicationDomain | The SemanticIDs for PROFIBUS follow the different device models, that are defined for PROFIBUS devices. FDT currently supports the following models:<br>PROFIBUS DP,<br>PROFIBUS PA,<br>PROFIdrive.<br><br>**PROFIBUS PA**<br>The applicationDomain is: FDT_PROFIBUS_PA<br>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.<br>The semanticId for PROFIBUS profile-related parameter follows the following rules:<br>The semanticId must be built based on the names defined in the profiles<br>Structured parameters must be combined with a '.'<br>Spaces within the profile definition must be exchanged with an underscore.<br>Blocks must be counted according to the Object Dictionary |

| Attribute | Description for use in Profibus |
|---|---|
| | The block number must be part of the semanticID<br><br>The semanticId is:<br>BlockType.BlockIndex.NameOfParameter.AttributeOfParameter<br>*Example:*<br>AnalogInputFB.3.OUT.Unit<br><br>**PROFIdrive**<br>The applicationDomain is: FDT_PROFIBUS_PROFIDRIVE<br>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.<br>A parameter may contain simple data or composed data (e.g. arrays).<br>The data of the device are accessible via a parameter channel (normally slot 0 index 47).<br>The semanticId is: DOdo-id.PNUpnu<br>do-id Drive Object ID<br>pnu ParameterNumber<br>do-id, pnu are based on decimal format without leading '0'<br>*Example:*<br>DO3.PNU64<br><br>**PROFIBUS DPV1**<br>The applicationDomain is: FDT_PROFIBUS_DPV1<br>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.<br>The semanticID for devices not based on a profile is directly based on the PROFIBUS address information:<br>The semanticId is: API*xx*.SLOT*yy*.INDEX*zz*<br>xx API<br>yy Slot<br>zz Index<br>xx, yy, zz are based on decimal format without leading '0' |
| subDeviceType | Enter manufacturer-specific value here |



## 6 Bus category

Profibus protocol is identified by the following unique identifier in busCategory attributes within XML BusCategory elements:

| BusCategory Element | Description |
|---|---|
| 036D1499-387B-11D4-86E1-00E0987270B9 | Support of Profibus DP V0 protocol |
| 036D1497-387B-11D4-86E1-00E0987270B9 | Support of Profibus DP V1 protocol |

Profibus uses the following unique identifier in physicalLayer attributes within XML PhysicalLayer elements:

| PhysicalLayer Element | Description |
|---|---|
| 036D1590-387B-11D4-86E1-00E0987270B9 | IEC 61158-2 (Profibus PA) |
| 036D1591-387B-11D4-86E1-00E0987270B9 | RS485 |
| 036D1592-387B-11D4-86E1-00E0987270B9 | Fiber |
| 036D1593-387B-11D4-86E1-00E0987270B9 | Ethernet |

## 7 Communication schema

Used at: IFdtCommunication::ConnectRequest()

IFdtCommunicationEvents2::OnConnectResponse2()

IFdtCommunication::DisconnectRequest()

IFdtCommunicationEvents::OnDisconnectResponse ()

IFdtCommunication::TransactionRequest()

IFdtCommunicationEvents::OnTransactionResponse()

### 7.1 DPV0 communication

The XML document contains the address information and the communication data.

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, it returns the TransactionRequest() with a result="false". (See Tables 2 and 3.)

**Table 2 – DPV0CommunicationSchema attributes**

| Attribute | Description |
|---|---|
| busAddress | Address information according to the IEC 61158 series (see also DTMParameterSchema, attribute busAddress) |
| connectStatus | Describes the connection status established by the communication component.<br>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 whether the device is available or not.<br>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.<br>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 |

| Attribute | Description |
|---|---|
| errorCode | Status information according to the IEC 61158 series.<br>For description of error code see:<br>DIN 19245 Part 3, PROFIBUS (P. 40ff., 83ff., 39) |
| communicationReference | 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 | Delay time in ms between two communication calls |
| sequenceTime | Period of time in ms for the whole sequence |
| schemaVersion | Defines the version of the schema |

**Table 3 – DPV0CommunicationSchema elements**

| Tag | Description | Master Class |
|---|---|---|
| Abort | Describes the abort | - |
| ConnectRequest | Describes the communication request to establish a connection to a Profibus master device.<br>Depending on the network infrastructure used it is possible that this service is not mapped to a field bus request, but is used to manage the software.<br>It is also possible that this service is used as a trigger to set the state of a device. | - |
| ConnectResponse | Describes the communication response to the connect request and provides the information on how the following WriteUserParameter commands will be sent to the device (see connectStatus).<br>The resulting connection depends on the communication device used and on the configuration of the bus master. The DeviceDTM must expect that the resulting connection cannot be used to access all services.<br>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. It should be noted that 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.<br>For an overview of availability of services depending on the type of connection, see the following tables | - |
| DisconnectRequest | Describes the communication request to release a connection to a Profibus master device | - |
| DisconnectResponse | Describes the communication response | - |
| FDT | Root tag | - |
| ReadUserParameterRequest | Describes the communication request according to the Profibus DPV0 specification.<br>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.<br>This is based on a local service or not available | 1 |

| Tag | Description | Master Class |
|---|---|---|
| ReadUserParameterResponse | 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 | – |
| WriteUserParameterRequest | Describes the communication request according to the Profibus DPV0 specification. The user parameter will be send according to the established connection (see connectStatus).<br>For Master Cl. 1 this is based on the service “DDLM_Set_Prm”.<br>For Master Cl. 2 this is based on the service “DDLM_Set_Prm” (an optional service) | 1, 2 |
| WriteUserParameterResponse | Describes the communication response according to the Profibus DPV0 specification | - |
| ReadOutputDataRequest | Describes the communication request according to the Profibus DPV0 specification<br>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.<br>For Master Cl. 1 this is a proprietary service (or un-available).<br>For Master Cl. 2 refer to [1], section 8.2.2.3.6. “Read Output” (an optional service) | 1, 2 |
| ReadOutputDataResponse | Describes the communication response according to the Profibus DPV0 specification | – |
| SequenceBegin | Describes the sequence begin | – |
| SequenceEnd | Describes the sequence end | – |
| SequenceStart | Describes the sequence start | – |
| WriteOutputDataRequest | Describes the communication request according to the Profibus DPV0 specification.<br>The output data will be sent according to the established connection.<br>It is necessary to specify which data will be written by providing an fdt:ChannelReference.<br>If the underlying FDTChannel is provided as Master Cl. 2, writing Output Data will be not possible.<br>Refer to IEC 61158-5, 8.2.2.3.5: “Set Output”<br>This functionality depends on the type of master used. In terms of PLC this functionality is named “forcing” of values | 1 |
| WriteOutputDataResponse | Describes the communication response according to the Profibus DPV0 specification | – |
| ReadInputDataRequest | Describes the communication request according to the Profibus DPV0 specification.<br>Depending on whether the underlying FDTChannel is an 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.<br>For Master Cl.1 refer to IEC 61158-5, 8.2.2.3.3: “Get Input”.<br>For Master Cl.2 refer to IEC 61158-5, 8.2.2.3.2: “Read Input” | 1, 2 |
| ReadInputDataResponse | Describes the communication response according to the Profibus DPV0 specification | – |
| ReadDiagnosisDataRequest | Describes the communication request according to the Profibus DPV0 specification.<br>For Master Cl.1 refer to IEC 61158-5, 8.2.3.3.2: “Get slave diag”.<br>For Master Cl.2 refer to IEC 61158-5, 8.2.3.3.3: “Read slave diag” | 1, 2 |

| Tag | Description | Master Class |
|---|---|---|
| ReadDiagnosisDataResponse | Describes the communication response according to the Profibus DPV0 specification | – |

Depending on the bus master type and on the returned connectStatus the following services are available, see Table 4.

**Table 4 – Availability of services depending on master and connect status**

| For Master Class1 (C1) | | | |
|---|---|---|---|
| | connectStatus | | |
| Slave DTM Service Request | masterConnectedOnly | DeviceAtLifeList | DeviceInDataExchange |
| Connect | ✓ | ✓ | ✓ |
| ReadUserParameter | ✓ | ✓ | ✓ |
| WriteUserParameter | ✓ | ✓ | ✓ |
| ReadOutputData | | | ✓ |
| WriteOutputData | | | O |
| ReadInputData | | | ✓ |
| ReadDiagnosisData | | ✓ | ✓ |
| | | | |
| **For Master Class2 (C2)** | | | |
| | connectStatus | | |
| Slave DTM Action | masterConnectedOnly | DeviceAtLifeList | DeviceInDataExchange |
| Connect | ✓ | ✓ | ✓ |
| ReadUserParameter | ✓ | ✓ | ✓ |
| WriteUserParameter | ✓ | ✓ | ✓ |
| ReadOutputData | | | ✓ |
| WriteOutputData | | | |
| ReadInputData | | | ✓ |
| ReadDiagnosisData | | ✓ | ✓ |
| NOTE<br>✓ means that the service is available;<br>O means that the service is optional and may be available, depending on the capabilities of the underlying master device. | | | |



```
<Schema name="FDTProfibusDPV0CommunicationSchema" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes" xmlns:fdt="x-schema:FDTDataTypesSchema.xml">
    <!--Definition of Attributes-->
    <AttributeType name="schemaVersion" dt:type="number" default="1.21"/>
    <AttributeType name="busAddress" dt:type="ui1"/>
    <AttributeType name="errorCode" dt:type="bin.hex"/>
    <AttributeType name="communicationReference" dt:type="uuid"/>
    <AttributeType name="connectStatus" dt:type="enumeration" dt:values="masterConnectedOnly deviceAtLifeList
deviceInDataExchange "/>
    <AttributeType name="sequenceTime" dt:type="ui4"/>
    <AttributeType name="delayTime" dt:type="ui4"/>
    <!--Definition of Elements-->
    <ElementType name="ConnectRequest" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="busAddress" required="yes"/>
        <attribute type="fdt:systemTag" required="no"/>
    </ElementType>
    <ElementType name="ConnectResponse" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
```

```
        <attribute type="busAddress" required="yes"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="connectStatus" required="yes"/>
    </ElementType>
    <ElementType name="DisconnectRequest" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="busAddress" required="yes"/>
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="DisconnectResponse" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="busAddress" required="yes"/>
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="ReadUserParameterRequest" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="ReadUserParameterResponse" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="errorCode" required="yes"/>
        <element type="fdt:CommunicationData" minOccurs="0" maxOccurs="1"/>
    </ElementType>
    <ElementType name="WriteUserParameterRequest" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <element type="fdt:CommunicationData" minOccurs="1" maxOccurs="1"/>
    </ElementType>
    <ElementType name="WriteUserParameterResponse" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="errorCode" required="yes"/>
    </ElementType>
    <ElementType name="ReadOutputDataRequest" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <element type="fdt:ChannelReference" minOccurs="0" maxOccurs="1"/>
    </ElementType>
    <ElementType name="ReadOutputDataResponse" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="errorCode" required="yes"/>
        <element type="fdt:CommunicationData" minOccurs="0" maxOccurs="1"/>
    </ElementType>
    <ElementType name="WriteOutputDataRequest" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <element type="fdt:ChannelReference" minOccurs="1" maxOccurs="1"/>
        <element type="fdt:CommunicationData" minOccurs="1" maxOccurs="1"/>
    </ElementType>
    <ElementType name="WriteOutputDataResponse" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="errorCode" required="yes"/>
    </ElementType>
    <ElementType name="ReadInputDataRequest" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <element type="fdt:ChannelReference" minOccurs="0" maxOccurs="1"/>
    </ElementType>
    <ElementType name="ReadInputDataResponse" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="errorCode" required="yes"/>
        <element type="fdt:CommunicationData" minOccurs="0" maxOccurs="1"/>
    </ElementType>
    <ElementType name="ReadDiagnosisDataRequest" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="ReadDiagnosisDataResponse" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="communicationReference" required="yes"/>
        <attribute type="errorCode" required="yes"/>
        <element type="fdt:CommunicationData" minOccurs="0" maxOccurs="1"/>
    </ElementType>
    <ElementType name="SequenceBegin" content="empty" model="closed">
```

```
        <attribute type="sequenceTime" required="no"/>
        <attribute type="delayTime" required="no"/>
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="SequenceEnd" content="empty" model="closed">
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="SequenceStart" content="empty" model="closed">
        <attribute type="communicationReference" required="yes"/>
    </ElementType>
    <ElementType name="Abort" content="empty" model="closed">
<attribute type="communicationReference" required="no"/>
</ElementType>
    <ElementType name="FDT" content="eltOnly" model="closed">
        <attribute type="schemaVersion" required="no"/>
        <attribute type="fdt:nodeId" required="no"/>
        <group order="one">
            <element type="ConnectRequest"/>
            <element type="ConnectResponse"/>
            <element type="DisconnectRequest"/>
            <element type="DisconnectResponse"/>
            <element type="ReadUserParameterRequest"/>
            <element type="ReadUserParameterResponse"/>
            <element type="WriteUserParameterRequest"/>
            <element type="WriteUserParameterResponse"/>
            <element type="ReadOutputDataRequest"/>
            <element type="ReadOutputDataResponse"/>
            <element type="WriteOutputDataRequest"/>
            <element type="WriteOutputDataResponse"/>
            <element type="ReadInputDataRequest"/>
            <element type="ReadInputDataResponse"/>
            <element type="ReadDiagnosisDataRequest"/>
            <element type="ReadDiagnosisDataResponse"/>
            <element type="SequenceBegin"/>
            <element type="SequenceEnd"/>
            <element type="SequenceStart"/>
            <element type="Abort"/>
            <element type="fdt:CommunicationError"/>
        </group>
    </ElementType>
</Schema>
```

## 7.2 DPV1 communication

The XML document contains the address information and the communication data, see Table 5.

**Table 5 – DPV1CommunicationSchema attributes and elements**

| Attribute | Description |
|---|---|
| api | Address information according to the IEC 61158 series.<br>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 | Address information according to the IEC 61158 series (see also DTMParameterSchema, attribute busAddress) |
| errorCode | Status information according to the IEC 61158 series.<br>For description of error code see:<br>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).<br>For description of abort information see:<br>DIN 19245 Part 3, PROFIBUS, (P. 54ff., 11) |
| index | Address information according to the IEC 61158 series |
| communicationReference | 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 | Minimum delay time in ms between two communication calls |
| MaxLenDataUnit | Optional attribute, to describe the amount of data, which can be transferred via the established connection<br>If this attribute is not available, no special length restriction is announced<br>Each communication component within the chain of interfaces concerning nested communication could introduce this attribute<br>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<br>If a communication component has no restriction, it should hand over the given value<br>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<br>If the data length is not applicable, the DTM should send an error message via OnErrorMessage() |
| networkMACAddress | Network or MAC address as described in the IEC 61158 series.<br>If there is a networkMACAddress, (i.e. networkMACAddress <> ""), the Profibus-DPV1-Master-DTM must set the corresponding “address type” in the Initiate request telegram “1”, else “0”.<br>Example:<br>Device with address 20 (14hex) addressed via a linking device:<br>networkMACAddress=“00000000000014” |
| scl | Access level as described in the IEC 61158 series.<br>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 | Period of time in ms for the whole sequence |
| slot | Address information according to the IEC 61158 series |
| schemaVersion | Defines the version of the schema |

| Attribute | Description |
|---|---|
| systemTag | System Tag of a DTM. It is strongly recommended to provide the attribute in the Request document |
| | |
| **Element** | **Description** |
| Abort | Describes the abort |
| ConnectRequest | Describes the communication request |
| ConnectResponse | Describes the communication response |
| destNetworkAddress | Describes the extended address of the destination |
| DisconnectRequest | Describes the communication request |
| DisconnectResponse | Describes the communication response |
| FDT | Root tag |
| NetworkAddress | Describes the extended address format |
| ReadRequest | Describes the communication request |
| ReadResponse | Describes the communication response |
| redundantAddresses | Describes the redundant addresses |
| SequenceBegin | Describes the sequence begin |
| SequenceEnd | Describes the sequence end |
| SequenceStart | Describes the sequence start |
| srcNetworkAddress | Describes the extended address of the source |
| WriteRequest | Describes the communication request |
| WriteResponse | Describes the communication response |

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 preferred address; the addresses in the redundantAddresses element should be used in this order if an alternative address must be used to connect to the redundant slave.



```
<Schema name="FDTProfibusDPV1CommunicationSchema" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes" xmlns:fdt="x-schema:FDTDataTypesSchema.xml" xmlns:fdtparam="x-
schema:DTMParameterSchema.xml">
    <!--Definition of Attributes-->
    <AttributeType name="schemaVersion" dt:type="number" default="1.21"/>
    <AttributeType name="api" dt:type="ui1"/>
    <AttributeType name="busAddress" dt:type="ui1"/>
    <AttributeType name="errorCode" dt:type="bin.hex"/>
    <AttributeType name="index" dt:type="ui1"/>
    <AttributeType name="communicationReference" dt:type="uuid"/>
    <AttributeType name="slot" dt:type="ui1"/>
    <AttributeType name="sequenceTime" dt:type="ui4"/>
    <AttributeType name="delayTime" dt:type="ui4"/>
    <AttributeType name="maxLenDataUnit" dt:type="ui1"/>
    <AttributeType name="scl" dt:type="ui1"/>
    <AttributeType name="networkMACAddress" dt:type="bin.hex"/>
    <!--Definition of Elements-->
    <ElementType name="redundantAddresses" content="eltOnly" model="closed">
        <element type="fdtparam:SlaveAddress" maxOccurs="*" minOccurs="1"/>
    </ElementType>

    <ElementType name="NetworkAddress" content="empty" model="closed">
        <attribute type ="api" required="yes"/>
        <attribute type ="scl" required="yes"/>
        <attribute type ="networkMACAddress" required="yes"/>
    </ElementType>
    <ElementType name="srcNetworkAddress" content="eltOnly" model="closed">
        <element type="NetworkAddress" minOccurs="1" maxOccurs="1"/>
    </ElementType>
    <ElementType name="destNetworkAddress" content="eltOnly" model="closed">
        <element type="NetworkAddress" minOccurs="1" maxOccurs="1"/>
    </ElementType>
```

```
<ElementType name="ConnectRequest" content="eltOnly" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="api" required="yes"/>
    <attribute type="busAddress" required="yes"/>
    <attribute type="fdt:systemTag" required="no"/>
        <element type="redundantAddresses" maxOccurs="1" minOccurs="0"/>
    <element type="srcNetworkAddress" minOccurs="0" maxOccurs="1"/>
    <element type="destNetworkAddress" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="ConnectResponse" content=" eltOnly" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="api" required="yes"/>
    <attribute type="busAddress" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
    <attribute type="errorCode" required="yes"/>
    <attribute type="maxLenDataUnit" required="no"/>
        <element type="redundantAddresses" maxOccurs="1" minOccurs="0"/>
    <element type="srcNetworkAddress" minOccurs="0" maxOccurs="1"/>
    <element type="destNetworkAddress" minOccurs="0" maxOccurs="1"/>
</ElementType>
<ElementType name="DisconnectRequest" content="empty" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="api" required="yes"/>
    <attribute type="busAddress" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
</ElementType>
<ElementType name="DisconnectResponse" content="empty" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="api" required="yes"/>
    <attribute type="busAddress" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
    <attribute type="errorCode" required="yes"/>
</ElementType>
<ElementType name="ReadRequest" content="empty" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="slot" required="yes"/>
    <attribute type="index" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
</ElementType>
<ElementType name="ReadResponse" content="eltOnly" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="slot" required="yes"/>
    <attribute type="index" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
    <attribute type="errorCode" required="yes"/>
    <element type="fdt:CommunicationData" minOccurs="1" maxOccurs="1"/>
</ElementType>
<ElementType name="WriteRequest" content="eltOnly" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="slot" required="yes"/>
    <attribute type="index" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
    <element type="fdt:CommunicationData" minOccurs="1" maxOccurs="1"/>
</ElementType>
<ElementType name="WriteResponse" content="empty" model="closed">
    <attribute type="fdt:nodeId" required="no"/>
    <attribute type="slot" required="yes"/>
    <attribute type="index" required="yes"/>
    <attribute type="communicationReference" required="yes"/>
    <attribute type="errorCode" required="yes"/>
</ElementType>
<ElementType name="SequenceBegin" content="empty" model="closed">
    <attribute type="sequenceTime" required="no"/>
    <attribute type="delayTime" required="no"/>
    <attribute type="communicationReference" required="yes"/>
</ElementType>
<ElementType name="SequenceEnd" content="empty" model="closed">
    <attribute type="communicationReference" required="yes"/>
</ElementType>
<ElementType name="SequenceStart" content="empty" model="closed">
    <attribute type="communicationReference" required="yes"/>
</ElementType>
<ElementType name="Abort" content="empty" model="closed">
    <attribute type="communicationReference" required="no"/>
</ElementType>
<ElementType name="FDT" content="eltOnly" model="closed">
    <attribute type="schemaVersion" required="no"/>
    <attribute type="fdt:nodeId" required="no"/>
```

```
        <group order="one">
            <element type="ConnectRequest"/>
            <element type="ConnectResponse"/>
            <element type="DisconnectRequest"/>
            <element type="DisconnectResponse"/>
            <element type="ReadRequest"/>
            <element type="ReadResponse"/>
            <element type="WriteRequest"/>
            <element type="WriteResponse"/>
            <element type="SequenceBegin"/>
            <element type="SequenceEnd"/>
            <element type="SequenceStart"/>
            <element type="Abort"/>
            <element type="fdt:CommunicationError"/>
        </group>
    </ElementType>
</Schema>
```

Example:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusDPV1CommunicationSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <ReadResponse slot="1" index="1" communicationReference="13966300-d860-4d18-8bc3-f11f8c9d7597"
errorCode="0000000000">
        <fdt:CommunicationData byteArray="FF01"/>
    </ReadResponse>
</FDT>


<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusDPV1CommunicationSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <ReadResponse slot="1" index="1" communicationReference="6B29FC40-CA47-1067-B31D-00DD010662DA"
errorCode="0">
        <fdt:CommunicationData byteArray="FF01"/>
    </ReadResponse>
</FDT>
```

Example for redundancy:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusDPV1CommunicationSchema.xml" xmlns:fdt="x-schema:FDTDataTypesSchema.xml"
xmlns:fdtparam="x-schema:DTMParameterSchema.xml">
    <ConnectRequest api="4" busAddress="1">
        <redundantAddresses>
            <fdtparam:SlaveAddress slaveAddress="2"/>
            <fdtparam:SlaveAddress slaveAddress="3"/>
        </redundantAddresses>
    </ConnectRequest>
</FDT>
```



## 8 ChannelParameterSchema

The following Table 6 describes the mapping of DPV1 Data types to FDT data types.

**Table 6 – Mapping of DPV1 data types to FDT data types**

| PROFIBUS Data types | FDT datatype |
|---|---|
| Boolean | Binary (0 = FALSE, all other values are considered to be TRUE) |
| Integer8 | int |
| Integer16 | int |
| Integer32 | int |

| PROFIBUS Data types | FDT datatype |
|---|---|
| Unsigned8 | byte |
| Unsigned16 | unsigned |
| Unsigned32 | unsigned |
| Floating Point | float |
| Visible String | string |
| Octet String | hexString |
| Date | dateAndTime |
| Time of Day | time |
| Time Difference | duration |
| DS-32 | structured |
| 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 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 (DPAddress or DPV1Address) also contains information that is necessary to retrieve the information from the data structure.

Used at: IFdtChannel::GetChannelParameters()

The ChannelParameter XML document describes 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 Table 7 provides a description of attributes and elements of the document.

**Table 7 – ChannelParameter attributes and elements**

| Attribute | Description |
|---|---|
| api | Address information according to the IEC 61158 series for channels accessible via Profibus DPV1 |
| bitLength | Additional data type information especially for fieldbus specific data types like 12 bit integer |
| bitPosition | Address information according to the IEC 61158 series for channels accessible via Profibus DP |
| frameApplicationTag | Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces |
| gatewayBusCategory | Unique identifier for a supported bus type like Profibus or HART according to the FDT specific CATID |
| invalidBit | Bit position of the invalid status channel accessible via Profibus DP |
| index | Address information according to the IEC 61158 series for channels accessible via Profibus DPV1 |
| logic | Additional data type information: positive 0=FALSE 1=TRUE |
| number | Address information for diagnosis according to the IEC 61158 series for channels accessible via Profibus DP. |

| Attribute | Description |
|---|---|
| | Read DP – Slave Diagnostic Information |
| protectedByChannelAssignment | TRUE if the channel is set to read only by the Frame Application. Usually set to TRUE if a channel assignment exists |
| schemaVersion | Defines the version of the schema |
| simulationBit | Bit position of the simulation status channel accessible via Profibus DP |
| substituteValueBit | Bit position of the substitute status channel accessible via Profibus DP |
| statusChannel | TRUE if the channel is for status information only |
| slotNumber | Address information according to the IEC 61158 series for channels accessible via Profibus DPV1 |

| Elements | Description |
|---|---|
| DpAddress | Address information according to the IEC 61158 series for channels accessible via Profibus DP |
| DpV1Address | 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 |
| FDT | Root tag |
| FDTChannel | Description of the channel |
| FDTChannelType | Description of the channel component in case of channels with gateway functionality |
| StatusInformation | Description of additional status information for channels accessible via Profibus DP |

```
<Schema name="FDTProfibusChannelParameterSchema" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes" xmlns:fdt="x-schema:FDTDataTypesSchema.xml" xmlns:appId="x-
schema:FDTApplicationIdSchema.xml">
    <!--Definition of Attributes-->
    <AttributeType name="schemaVersion" dt:type="number" default="1.21"/>
    <AttributeType name="api" dt:type="ui1"/>
    <AttributeType name="bitLength" dt:type="ui4"/>
    <AttributeType name="bitPosition" dt:type="ui4"/>
    <AttributeType name="frameApplicationTag" dt:type="string"/>
    <AttributeType name="gatewayBusCategory" dt:type="uuid"/>
    <AttributeType name="invalidBit" dt:type="ui4"/>
    <AttributeType name="index" dt:type="ui1"/>
    <AttributeType name="logic" dt:type="enumeration" dt:values="positive negative"/>
    <AttributeType name="number" dt:type="ui4"/>
    <AttributeType name="protectedByChannelAssignment" dt:type="boolean"/>
    <AttributeType name="simulationBit" dt:type="ui4"/>
    <AttributeType name="statusChannel" dt:type="boolean"/>
    <AttributeType name="substituteValueBit" dt:type="ui4"/>
    <AttributeType name="slotNumber" dt:type="ui1"/>
    <!--Definition of Elements-->
    <ElementType name="DpAddress" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="bitPosition" required="yes"/>
        <attribute type="bitLength" required="yes"/>
    </ElementType>
    <ElementType name="DpV1Address" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="api" required="yes"/>
        <attribute type="slotNumber" required="yes"/>
        <attribute type="index" required="yes"/>
        <attribute type="bitPosition" required="no"/>
        <attribute type="bitLength" required="no"/>
    </ElementType>
    <ElementType name="StatusInformation" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="logic" required="yes"/>
        <attribute type="invalidBit" required="no"/>
        <attribute type="simulationBit" required="no"/>
```

```
        <attribute type="substituteValueBit" required="no"/>
    </ElementType>
    <ElementType name="FDTChannel" content="eltOnly" model="closed" order="seq">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="fdt:tag" required="yes"/>
        <attribute type="fdt:id" required="yes"/>
        <attribute type="fdt:descriptor" required="no"/>
        <attribute type="protectedByChannelAssignment" required="yes"/>
        <attribute type="number" required="yes"/>
        <attribute type="fdt:dataType" required="yes"/>
        <attribute type="fdt:signalType" required="yes"/>
        <attribute type="frameApplicationTag" required="no"/>
        <attribute type="appId:applicationId" required="no"/>
        <element type="fdt:SemanticInformation" minOccurs="0" maxOccurs="*"/>
        <element type="fdt:BitEnumeratorEntries" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:EnumeratorEntries" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:Unit" minOccurs="0" maxOccurs="1"/>
        <element type="DpAddress" minOccurs="0" maxOccurs="1"/>
        <element type="DpV1Address" minOccurs="0" maxOccurs="1"/>
        <element type="StatusInformation" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:Alarms" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:Ranges" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:Deadband" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:SubstituteValue" minOccurs="0" maxOccurs="1"/>
        <element type="fdt:StructuredElements" minOccurs="0" maxOccurs="1"/>  <!--should be used if the data type is
structured-->
    </ElementType>
    <ElementType name="FDTChannelType" content="eltOnly" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <element type="fdt:VersionInformation" minOccurs="1" maxOccurs="1"/>
        <attribute type="gatewayBusCategory" required="no"/>
        <attribute type="statusChannel" required="no"/>
    </ElementType>
    <ElementType name="FDT" content="eltOnly" model="closed">
        <attribute type="schemaVersion" required="no"/>
        <attribute type="fdt:nodeId" required="no"/>
        <element type="FDTChannelType" minOccurs="1" maxOccurs="1"/>
        <element type="FDTChannel" minOccurs="1" maxOccurs="1"/>
    </ElementType>
</Schema>
```

Example:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-schema:FDTDataTypesSchema.xml">
    <FDTChannelType>
        <fdt:VersionInformation name="myname" vendor="myVendor" version="1.0" date="2000-08-05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="myTag" fdt:id="PV" protectedByChannelAssignment="0" number="123" fdt:dataType="float"
fdt:signalType="output">
        <DpAddress bitPosition="17" bitLength="32"/>
        <DpV1Address api="1" slotNumber="1" index="1" bitPosition="0" bitLength="32"/>
        <StatusInformation logic="positive"/>
        <fdt:Alarms>
            <fdt:Alarm alarmType="lowAlarm">
                <fdt:StaticValue staticValue="25"/>
            </fdt:Alarm>
            <fdt:Alarm alarmType="highAlarm">
                <fdt:StaticValue staticValue="100"/>
            </fdt:Alarm>
        </fdt:Alarms>
        <fdt:Ranges>
            <fdt:Range>
                <fdt:LowerRange>
                    <fdt:ChannelReference idref="PV_LOWER_RANGE_VALUE"/>
                </fdt:LowerRange>
                <fdt:UpperRange>
                    <fdt:ChannelReference idref="PV_UPPER_RANGE_VALUE"/>
                </fdt:UpperRange>
                <fdt:Unit>
                    <fdt:ChannelReference idref="PV_RANGE_VALUES_UNITS_CODE"/>
                </fdt:Unit>
            </fdt:Range>
        </fdt:Ranges>
    </FDTChannel>
</FDT>
```

## 9 Topology scan schema

Used at: IDtmEvents::OnScanResponse()

The XML document describes one entry in the list of scanned PROFIBUS-Devices.

| Tag | Description |
|---|---|
| ProfibusDevice | Specifies a Profibus device |

```
<Schema name="DTMProfibusDeviceSchema" xmlns="urn:schemas-microsoft-com:xml-data" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml" xmlns:dt="urn:schemas-microsoft-com:datatypes">
    <!--Definition of Attributes-->
    <AttributeType name="schemaVersion" dt:type="number" default="1.21"/>
    <AttributeType name="busAddress" dt:type="ui1"/>
    <!--Definition of Elements-->
    <ElementType name="ProfibusDevice" content="empty" model="closed">
        <attribute type="fdt:nodeId" required="no"/>
        <attribute type="schemaVersion" required="no"/>
        <attribute type="busAddress" required="yes"/>
        <attribute type="fdt:deviceTypeId" required="yes"/>
        <attribute type="fdt:subDeviceType" required="no"/>
    </ElementType>
</Schema>
```

## 10 Master-bus parameter set

The following parameter set represent the content of the attribute busMasterConfigurationPart within the DTMParameterSchema for Profibus master device, see Table 8. 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 8 – Bus parameter set for master device**

| Name | Type | Comment |
|---|---|---|
| Bus_Para_Len | Unsigned16 | Length of the bus parameter set inclusive the length parameter (range 34 to $2^{16}$-1) |
| FDL_Add | Unsigned8 | 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 | Unsigned8 | Code number for the baud rate |
| $T_{SL}$ | Unsigned16 | Slot Time |
| min $T_{SDR}$ | Unsigned16 | Min. station delay response |
| max $T_{SDR}$ | Unsigned16 | Max. station delay response |
| $T_{QUI}$ | Unsigned8 | Quit-time |
| $T_{SET}$ | Unsigned8 | Setup-time |
| $T_{TR}$ | Unsigned32 | Target rotation time |
| G | Unsigned8 | GAP Update Factor |
| HSA | Unsigned8 | Highest station address |
| max_retry_limit | Unsigned8 | Max. retry limit |
| Bp_Flag | Unsigned8 | Flags for the user interface, e.g. error action flag |

| Name | Type | Comment |
|---|---|---|
| Min_Slave_Interval | Unsigned16 | Smallest allowed time period between two slave poll cycles |
| Poll_Timeout | Unsigned16 | Master-master timeout |
| Data_Control_Time | Unsigned16 | Guaranteed time period between two Data_transfer_list updates |
| Octet 1 (reserved) | Octet-String | |
| .... | | |
| Octet 6 (reserved) | | |
| Master_User_Data_Len | Unsigned16 | Length of Master_User_Data inclusive lengthparameter |
| Master_Class2_Name | Visible-String (32) | Name of DP class 2 master the parameter set was created with |
| Master_User_Data | Octet-String | Manufacturer-specific |

## 11 Slave bus parameter set

The following parameter set represents the content of the attribute busMasterConfigurationPart within the DTMParameterSchema for Profibus slave devices, see Table 9. This attribute has to be set for each Profibus slave device according to the IEC 61158 series. For further details, refer to the sequence chart "Configuration of a Fieldbus Master" and the IEC 61158 series.

**Table 9 – Bus parameter set for slave device**

| Name | Type | Comment |
|---|---|---|
| Slave_Para_Len | Unsigned16 | Length of the slave parameter set inclusive the length parameter |
| Sl_Flag | Unsigned8 | Slave specific flags like New_Prm, Active, Fail_Safe, .... |
| Slave_Type | Unsigned8 | Manufacturer-specific slave type denotation (0 by default for DP-Slaves) |
| Octet 1 (reserved) | Octet-String | |
| .... | | |
| Octet 12 (reserved) | | |
| Prm_Data_Len | Unsigned16 | Length of Prm_Data inclusive the length parameter (range 9 to 246) |
| Prm_Data | Octet-String | |
| Cfg_Data_Len | Unsigned16 | Length of Cfg_Data inclusive the length parameter (range 3 to 246) |
| Cfg_Data | Octet-String | |
| Add_Tab_Len | Unsigned16 | Length of Add_Tab inclusive the length parameter (range 2 to $2^{16}$-31) |
| Add_Tab | Octet-String | Address assignment table |
| Slave_User_Data_Len | Unsigned16 | Length of Slave_User_Data inclusive the length parameter (range 2 to $2^{16}$-31) |
| Slave_User_Data | Octet-String | Manufacturer-specific data to characterize the DP-Slave for the master |

## 12 Module and channel data

A slave's current module configuration including the relevant 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 bitfield.

- Channels within PROFIBUS data frames, see Figure 1:

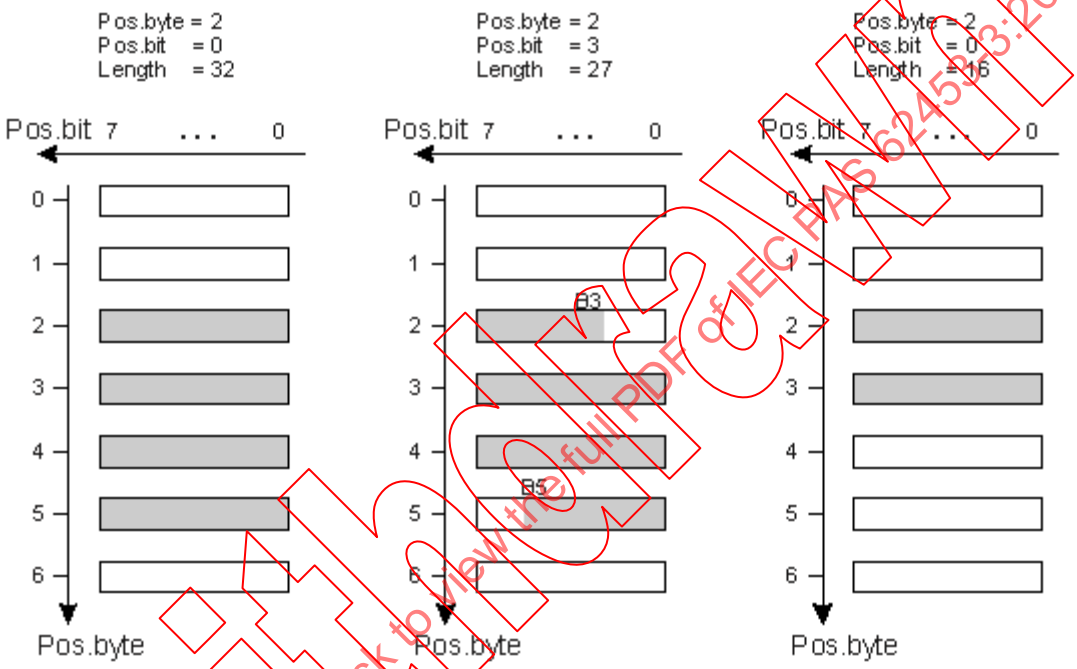


**Figure 1 – Example for IO data within telegramms**

- 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, see Table 10.

**Table 10 – Signal channels within the data frame**

| # Frame description Remote I/O | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Vendor: | "RIO manufacturer" | | | | | | | | | | | |
| Devicetype: | "LB 8101" | | Deviceid ent: | | | 0x8101 | # hex | | | | | |
| Frameident: | "B1t1" | | | | | | | | | | | |
| | | | | | | | | | | | | |
| # Input data | | | | | | | | | | | | |
| # Channel | # Channel type | # Data | # Data | # Data | # Invalid Bit | # Invalid Bit | # Subst. Value Bit | # Subst. Value Bit | # Sim. Bit | # Sim. Bit | # Status-Channel | |
| | | # PoS. byte | # PoS. bit | # Length | # PoS. byte | # Pos. bit | # 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 u. Output)“ | | | | | | | | | | |
| "DI_8_LFOUT0" | 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 | # Sim. Bit | # Sim. Bit | # Status-Channel | |
| | | # PoS. byte | # PoS. bit | # Length | # PoS. byte”# ”o“. b”t“# PoS. byte | | # PoS. byte | # Pos. bit | | | | |
| Modul: | "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 | |

- Corresponding XML documents:
- DTM Parameters:

```
<?xml version=" 1.0"?>
<FDT xmlns="x-sch ema:DTMParameterSchema.xml" xm lns:fdt="x-schema:FDTDataTypesSchema.xml"
 fdt:storageState="persistant"  fdt:dataSetState="default">
    <fdt:DtmDeviceType>
        <fdt:VersionInformation name="LB 8101" vendor="RIO Manufacturer" version="1.0" date="2000-08”05"/>
        <fdt:SupportedLanguages>
            <fdt:LanguageId languageid””1"/>
        </fdt:SupportedLanguages>
        <fdt:BusCategories>
            <fdt:BusCategory busCategory="036D1497-387B-11D4-86E1-00E098727”B9"/>
            <fdt:BusCategory busCategory="036D1499-387B-11D4-86E1-00E098727”B9"/>
        </fdt:BusCategories>
    </fdt:DtmDeviceType>
    <DtmDevice fdt:tag="00PGH10EC”01">
        <InternalTopology>
            <InternalChannel>
                <Module moduleid="1" slot="1">
                    <fdt:VersionInformation name="LB/FB 1”03" vendor="RIO Manufacturer" versio="10" date"2000-
08”05" descriptor"Digital Input (Counter 4 Byte"/>
                    <fdt:ChannelReferences>
                        <fdt:ChannelReference idref"Count_”_0"/>
                    </fdt:ChannelReferences>
                </Module>
            </InternalChannel>
            <InternalChannel>
                <Module moduleid="8" slot="8">
                    <fdt:VersionInformation name"1DO / 2DI" vendor="Vendor name" version="".0" date="2000-08”05"
descriptor="Valve Block LB/FB 2XXX 1Byte Input Output"/>
                    <fdt:ChannelReferences>
                        <fdt:ChannelReference idref="DI_8LFO”T0"/>
                        <fdt:ChannelReference idref="DI_”_1"/>
                        <fdt:ChannelReference idref="DI_8_”F1"/>
                        <fdt:ChannelReference idref="DI_”_2"/>
                        <fdt:ChannelReference idref="DI_8_”F2"/>
                        <fdt:ChannelReference idref="DO_”_0"/>
                    </fdt:ChannelReferences>
                </Module>
            </InternalChannel>
        </InternalTopology>
    </DtmDevice>
</FDT>
```

- FDT Channel “Count_1_0”:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType>
        <fdt:VersionInformation name="Digital Input (Counter 4 Byte)" vendor="RIO Manufacturer" version="1.0"
date="2000-08”05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="00PGH10EB001" fdt:id="Count_0_1" protectedByChannelAssignment="0" number="0"
fdt:dataType="int" fdt:signalType="input">
        <DpAddress bitPosition="0" bitLength=”32"/>
    </FDTChannel>
</FDT>
```

- FDT Channel “DI_8_LFOUT0”:

```
<?xml version=”1.0”?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType statusChannel="1">
        <fdt:VersionInformation name="Digital Input Valve Block" vendor="RIO Manufacturer" version="1.0" date="2000-
08-05"/>
    </FDTChannelType>
```

```
    <FDTChannel fdt:tag="00PGH10FB001" fdt:id="DI_8LFOUT0" protectedByChannelAssignment="0" number="10"
fdt:dataType="binary" fdt:signalType="input">
        <DpAddress bitPosition="33" bitLength="1"/>
    </FDTChannel>
</FDT>
```

- FDT Channel “DI_8_1”:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType>
        <fdt:VersionInformation name="Digital Input Valve Block" vendor="RIO Manufacturer" version="1.0" date="2000-
08-05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="00PGH10EB002" fdt:id="DI_8_1" protectedByChannelAssignment="0" number="1"
fdt:dataType="binary" fdt:signalType="input">
        <DpAddress bitPosition="34" bitLength="1"/>
        <StatusInformation logic="positive" invalidBit="35"/>
    </FDTChannel>
</FDT>
```

- FDT Channel “DI_8_LF1”:

```
<?xml version="1.0?">
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType statusChannel="1">
        <fdt:VersionInformation name="Digital Input Valve Block" vendor="RIO Manufacturer" version="1.0" date="2000-
08-05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="00PGH10FB002" fdt:id="DI_8_LF1" protectedByChannelAssignment="0" number="11"
fdt:dataType="binary" fdt:signalType="input">
        <DpAddress bitPosition="35" bitLength="1"/>
    </FDTChannel>
</FDT>
```

- FDT Channel “DI_8_2”:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType>
        <fdt:VersionInformation name="Digital Input Valve Block" vendor="RIO Manufacturer" version="1.0" date="2000-
08-05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="00PGH10EB003" fdt:id="DI_8_2" protectedByChannelAssignment="0" number="2"
fdt:dataType="binary" fdt:signalType="input">
        <DpAddress bitPosition="36" bitLength="1"/>
        <StatusInformation logic="positive" invalidBit="37"/>
    </FDTChannel>
</FDT>
```

- FDT Channel “DI_8_LF2”:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType statusChannel="1">
        <fdt:VersionInformation name="Digital Input Valve Block" vendor="RIO Manufacturer" version="1.0" date="2000-
08-05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="00PGH10FB003" fdt:id="DI_8_LF2" protectedByChannelAssignment="0" number="12"
fdt:dataType="binary" fdt:signalType="input">
        <DpAddress bitPosition="37" bitLength="1"/>
    </FDTChannel>
</FDT>
```

- FDT Channel “DO_8_0”:

```
<?xml version="1.0"?>
<FDT xmlns="x-schema:FDTProfibusChannelParameterSchema.xml" xmlns:fdt="x-
schema:FDTDataTypesSchema.xml">
    <FDTChannelType>
        <fdt:VersionInformation name="Digital Output Valve Block" vendor="RIO Manufacturer" version="1.0"
date="2000-08-05"/>
    </FDTChannelType>
    <FDTChannel fdt:tag="00PGH10EB004" fdt:id="DO_8_0" protectedByChannelAssignment="0" number="3"
fdt:dataType="binary" fdt:signalType="output">
        <DpAddress bitPosition="0" bitLength="1"/>
        <StatusInformation logic="positive" invalidBit="32"/>
    </FDTChannel>
</FDT>
```

# 13 ProfiSafe

## 13.1 Motivation

The purpose of the ProfiSafe-UseCase paper is to compile a collection of typical existing failsafe automation examples from the discrete and continuous manufacturing areas and to derive a common systematic approach. This proposed solution ought to be based on

- Profibus DPV1 and ProfiSafe communication principles,
- new possibilities of combined failsafe controllers (standard and failsafe programs within one PLC),
- and current activities of other Profibus working groups like FDT/DTM and Proxy FB.

Within the context of the ProfiSafe-UseCase paper this PAS covers the partial aspect "FDT programmers interface for individual device parameter verification".

## 13.2 General parameter handling

In order to communicate in a safe way, each ProfiSafe device (F-Slave) requires so-called F-Parameters for the adjustment of the operational mode of its ProfiSafe driver. Those parameters comprise watch dog time, safety integrity level, container size, etc. In case of simple F-slaves no further parameters are required. F-Parameters will therefore be defined within GSD Revision 4.0. See Figure 2.

In contrast, a complex F-slave requires additional individual and safety relevant parameters (individual device parameter) that, due to its size (>240 Bytes), often cannot be transported by the initial parameterization telegram (Prm-Telegram). As an amendment, ProfiSafe guidelines are suggesting an additional method via proxy function blocks in a safety PLC, thus providing additional functionality like program-controlled reparametrization, etc.

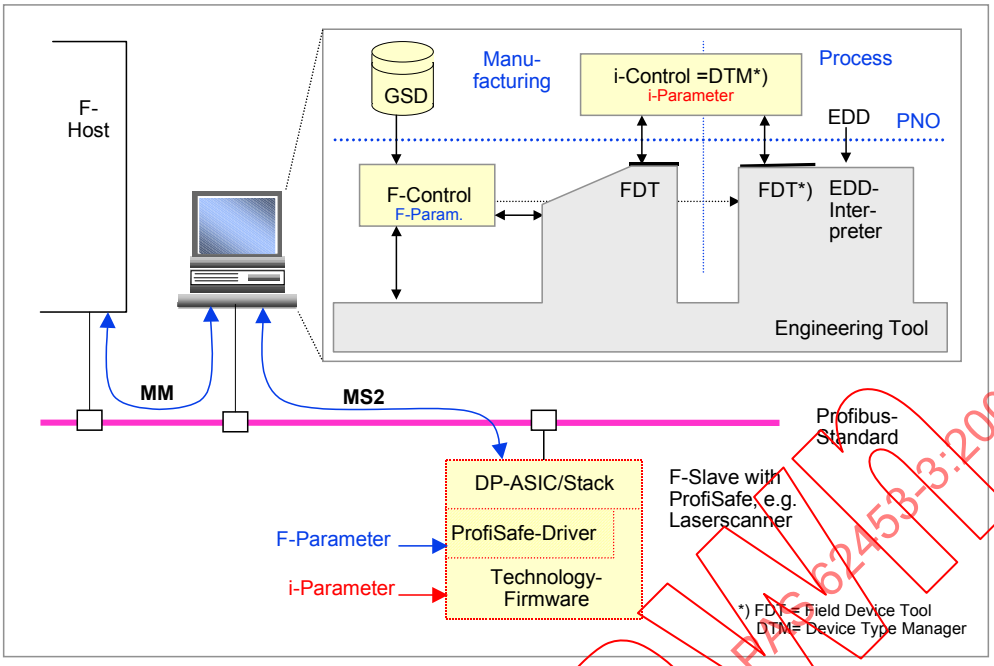


**Figure 2 – F-Parameter and individual device parameter**

Individual device parameters, being F-device specific by nature, ought to be handled by Device-Type-Manager programs (DTM) coming from the F-device manufacturers. The engineering tool operates as a Frame Application for such a DTM of a F-slave. It routes the communication requests of a DTM to its device (MS2) and provides data persistence. Otherwise it covers all traditional tasks like network configuration, parameterization, commissioning and diagnosis. It communicates with the PLC via the Master-Master-Protocol (MM). In this respect a PLC on the Profibus is a device of its own class (Master Class 1). The engineering tool itself is defined to be a "Master Class 2" device.

### 13.3 ProfiSafe individual device parameter

The structure of the ProfiSafe individual device parameter set is defined within the ProfiSafe guidelines. This set is wrapped up in XML-format for transportation across the FDT interface. See Figure 3.

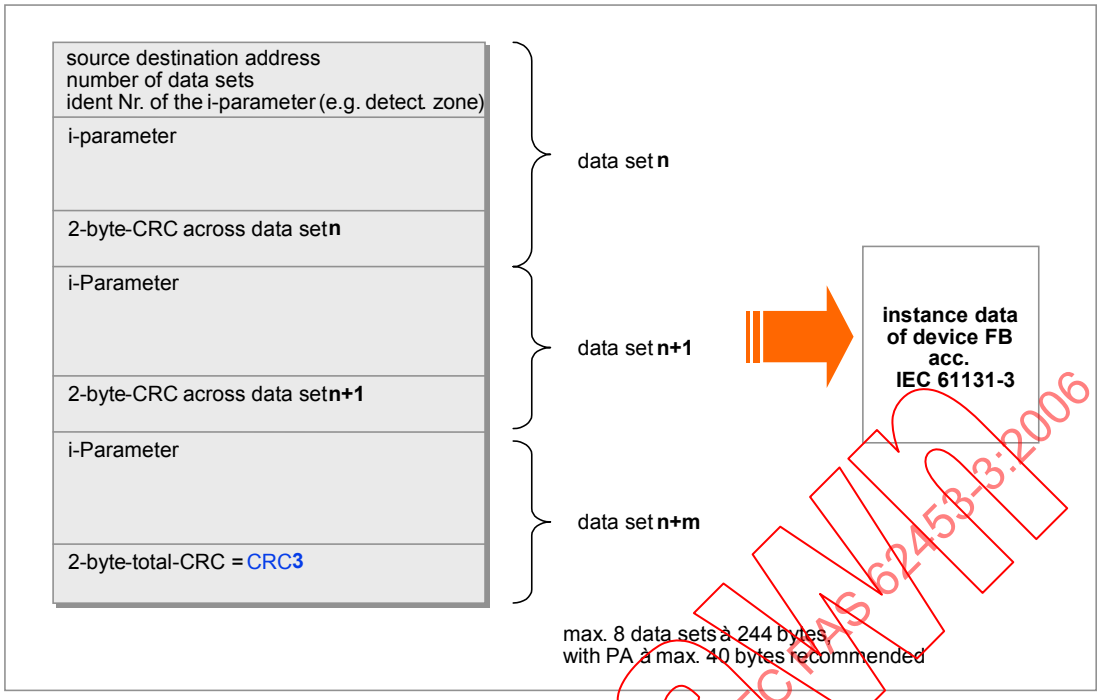


**Figure 3 – Data structure of ProfiSafe individual device parameters**

The data structures and the data formats of the inner individual device parameters are proprietary. The whole set is divided into parts that fit into a Profibus data set ready for transportation. Each data set comprises a 2-bytes CRC for data consistency check. The polynomial is fixed and defined within the ProfiSafe guidelines. The CRC of the last set checks the total block.

## 14 GSD information

The GSD information is not stored with single slave instances or in a global accessible file. It is provided by the DTM at IDtmInformation. On method GetInformation(), a DTM of a PROFIBUS device provides the GSD information within its XML document.

## 15 Profibus device identification

There are different Profibus specific identification elements.

A Profibus Scan may detect different device types: I&M devices, Profile devices or pure DP devices.

Following rule must be applied for Profibus communication channels:

- if I&M is available, create I&M identification,
- otherwise check Profile PA,
- otherwise create pure DP info = IDENT_NUMBER.

Profibus char array rules:

- in all strings based on char ranges defined in the fieldbus protocol specification, the leading spaces are left trimmed. The char array is to be filled with 0x20h (blank),
- in VisibleStrings, invisible characters provided by a device have to be replaced by '?'

### 15.1 FDTProfibusIdentSchema

The following Table 11 lists the identification relevant data, source and format for Profibus DP.

**Table 11 – FDTProfibusIdentSchema – attributes with Profibus DP specific mapping**

| Profibus-Attribute name (semantic name) | Semantic Element | Data request in physical device | Protocol specific name | Profibus data format | XML-FDT format (display ormat) |
|---|---|---|---|---|---|
| busProtocol | IdBusProtocol | For all DP devices: “protocol_DP” | Protocol | | Enum |
| slaveAddress | IdAddress | Bus address is provided as part of live list by a PROFIBUS master. Service: FMA1/2_LIVE_LIST | Slave Address | Unsigned8 | string (dec) |
| identNumber | IdTypeID | IDENT_NUMBER can be requested by: DP Service DDLM_SLAVE_DIAG<br>Allowed values are: Profile IDENT_NUMBER : 0x9700 ( 0x9700 to 0x9742) or<br>Manufacturer specific IDENT_NUMBER | IDENT_NUMBER | Unsigned16 | Bin.hex (hex) |

| Profibus-Attribute name (semantic name) | Semantic Element | Data request in physical device | Protocol specific name | Profibus data format | XML-FDT format (display ormat) |
|---|---|---|---|---|---|
| manufacturerSpecificExtension | | Can be used by DTM for a vendor-specific device identification information, e.g. by combining a number of device parameter values into one string value. This can be used to identify a specific device variant | | | String |

The following Table 12 shows the mapping for PROFIBUS I&M data

**Table 12 – FDTProfibusIdentSchema – attributes with Profibus I&M specific mapping**

| Profibus-Attribute name (semantic name) | Semantic Element | Data request in physical device | Protocol specific name | Profibus data format | XML-FDT format (display format) |
|---|---|---|---|---|---|
| identNumber, manufacturerSpecificExtension and slaveAddress as defined in DP table. | | | | | |
| busProtocol | IdBusProtocol | For all I&M devices : “protocol_IM” | Protocol | - | Enum |
| manufacturerId | IdManufacturer | I&M 0 Element 1<br>Mapped to PB.DEVICE_MAN_ID | MANUFACTURER_ID | Unsigned16 | ui2 |
| orderId | - | I&M 0 Element 2<br>Mapped to PB.DEVICE_ID | ORDER_ID | 20 Octets<br>Visible String | string [20] |
| serialNumber | IdSerialNumber | I&M 0 Element 3<br>unique serial number<br>Mapped to PB.DEVICE_SER_NUM | SERIAL_NUMBER | 16 Octets<br>VisibleString | string [16] |
| hardwareRevision | IdHardwareRevision | I&M 0 Element 4<br>HARDWARE_REVISION<br>NOTE:<br>If the device is a PA device,(PROFILE_ID is set to 0x9700), HARDWARE_REVISION must be read from PA_I&M0 [6] | HARDWARE_REVISION | 2 Octets - unsigned int 16<br>Converted to string to cover also :<br>PA_I&M0 : VisibleString 16 | String |

| Profibus-Attribute name (semantic name) | Semantic Element | Data request in physical device | Protocol specific name | Profibus data format | XML-FDT format (display format) |
|---|---|---|---|---|---|
| softwareRevision | IdSoftwareRevision | I&M 0 Element 5<br>SOFTWARE_REVISION<br>NOTE:<br>If the device is a PA device (PROFILE_ID is set to 0x9700), SOFTWARE_REVISION must be read from PA_I&M0<br>String format is used in FDT to harmonize version handling for all fieldbus protocols | SOFTWARE_REVISION | 4 Octets – 1 char + 3 unsigned 8 , e.g.: V1.2.3<br>PA_I&M0 : VisibleString 16 | String |
| profileID | - | I&M 0 Element 7 | PROFILE_ID | 2 Octets – unsigned 16 | ui2 |
| profileSpecificType | - | I&M 0 Element 8<br>MSB = BLOCK_OBJECT, LSB = PARENT_CLASS | PROFILE_SPECIFIC_ TYPE | 2 Octets – unsigned 16 | ui2 |
| imVersion | - | I&M 0 Element 9 | IM_VERSION | 2 Octets – unsigned 16; MSB major version xxx, LSB minor version yyy | Float<br>xxx.yyy |
| imSupported | - | I&M 0 Element 10 | IM_SUPPORTED | 2 Octets - | ui2 |
| tagFunction | IdTag | I&M 1 Element 1<br>TAG_FUNCTION | TAG_FUNCTION | 32 Octets – visible string | String [32] |
| tagLocation | - | I&M 1 Element 2<br>TAG_LOCATION | TAG_LOCATION | 22 Octets – visible string | String [22] |
| GenericSupport | IdDTMSupportLevel | Enumeration:<br>GenericDTM,<br>ProfileDTM,<br>BlockSpecificProfileDTM | DTM Support Level | - | Enum |

The following Table 13 lists the identification relevant data, source and format for Profibus PA.

**Table 13 – FDTProfibusIdentSchema – attributes with Profibus PA specific mapping**

| Profibus-Attribute name (semantic name) | Semantic Element | Data request in physical device | Protocol specific name | Profibus data format | XML-FDT format (display format) |
|---|---|---|---|---|---|
| identNumber, manufacturerSpecificExtension and slaveAddress as defined in DP table. | | | | | |
| busProtocol | IdBusProtocol | For all PA devices : “protocol_PA” | Protocol | - | enum |
| manufacturerId | IdManufacturer | Physical Block – Index 10 (Unsigned16 – 2 decimal) | DEVICE_MAN_ID | Unsigned16 | ui2 |
| device_id | - | Physical Block - Index 11 - VisibleString 16 | DEVICE_ID | VisibleString16 | string [16] |
| profile | - | Block structure of physical block – element 7 : OctetString - (Index 10 – size 2) | Profile | OctetString2 | ui2 |
| profileRevision | IdBusProtocolVersion | Block structure of physical block – element 8 Unsigned 16 - (Index 6) | Profile Revision | Unsigned16 | ui2 |
| softwareRevision | IdSoftwareRevision | Physical Block - Index 8 (VisibleString 16) | SOFTWARE_REVISION | VisibleString16 | string [16] |
| hardwareRevision | IdHardwareRevision | Physical Block - Index 9 (VisibleString 16) | HARDWARE_REVISION | VisibleString16 | string [16] |
| serialNumber | IdSerialNumber | Physical Block - Index 12 (VisibleString 16) | DEVICE_SER_NUM | VisibleString16 | string [16] |
| deviceTAG | IdTag | BO of PB - OctetString 32 | TAG_DESC | OctetString32 | String [16] |
| profileSpecificType | - | FDT attribute mapped to Block Object PARENT_CLASS sort of first TransducerBlock | BO PARENT_CLASS | Unsigned16 | ui2 |
| GenericSupport | IdDTMSupportLevel | Enumeration:<br>GenericDTM,<br>ProfileDTM,<br>BlockSpecificProfileDTM | DTM Support Level | | enum |

The following Table 14 gives a description of the attributes with protocol independent semantics.

**Table 14 – FDTProfibusIdentSchema – attributes with protocol independent semantics**

| Attribute | Description |
|---|---|
| schemaVersion | Identifies the schema version |
| idDTMSupportLevel | enumeration<br>genericSupport<br>profileSupport<br>blockspecificProfileSupport<br>specificSupport<br>identSupport |
| match | Used by device DTM to define a regular expression which must match to scanned physical define identification information |
| nomatch | Used by device DTM to define a regular expression which must not match to scanned physical define identification information.<br>Used by device DTM to indicate if identification information may not match |
| | |
| **Tag** | **Description** |
| RegExpr | Includes regular expression string – either for match or for nomatch |

```
<Schema name="FDTProfibusIdentSchema" xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
    <!--Definition of Attributes-->
    <AttributeType name="schemaVersion" dt:type="number" default="1.21"/>
    <AttributeType name="busProtocol" dt:type="enumeration" dt:values="protocol_DP protocol_IM  protocol_PA"/>
    <!--                                                                -->
    <!--                DP                                              -->
    <!--                                                                -->
    <AttributeType name="slaveAddress" dt:type="ui1"/>
    <!-- data source : live list provided by Profibus master - display decimal format -->
    <AttributeType name="identNumber" dt:type="bin.hex"/>
    <!-- Profile IDENT_NUMBER : 0x9700 ( 0x9700 to 0x9742) or Manufacturer specific IDENT_NUMBER -->
    <!-- Display format : 0xabcd -->
    <!--                                                                -->
    <!--                IM                                              -->
    <!--                                                                -->
    <AttributeType name="manufacturer_id" dt:type="ui2"/>
    <!--   I&M 0 Element 1Mapped to  PB.DEVICE_MAN_ID -->
    <AttributeType name="order_id" dt:type="string" dt:maxLength="20"/>
    <!--  I&M 0 Element 2 Mapped to PB.DEVICE_ID -->
    <AttributeType name="serialNumber" dt:type="string" dt:maxLength="16"/>
    <!--  I&M 0 Element 3 unique serial number - Mapped to PB.DEVICE_SER_NUM -->
    <AttributeType name="hardwareRevision" dt:type="string"/>
    <!-- I&M 0 Element 4 HARDWARE_REVISION - 2 Octets transformed  to string (to be consistent to PA_I&M) -->
    <!-- if PROFILE_ID is 0x9700 = device is PA device, HARDWARE_REVISION must be read from PA_I&M0 element 2 -->
    <AttributeType name="softwareRevision" dt:type="string"/>
    <!-- I&M 0 Element 5 SOFTWARE_REVISION - 4 Octets : 4 Octets - 1 char + 3 unsigned 8 , e.g.: V1.2.3 -->
    <!-- if PROFILE_ID is 0x9700 = device is PA device, SOFTWARE_REVISION must be read from PA_I&M0 element 3 -->
    <AttributeType name="profileID" dt:type="ui2"/>
    <!--I&M 0 Element 7  PROFILE_ID -->
    <AttributeType name="profileSpecificType" dt:type="ui2"/>
    <!--  I&M 0 Element 8 PROFILE_SPECIFIC_TYPE -  MSB = BLOCK_OBJECT, LSB = PARENT_CLASS -->
    <AttributeType name="imVersion" dt:type="float"/>
    <!--  I&M 0 Element 9 IM_VERSION - 2 Octets - unsigned 16; MSB major version xxx, LSB minor version yyy - convert to float xxx.yyy -->
    <!-- Frame Application and DTMs should split the float to major version and minor version for display -->
    <AttributeType name="imSupported" dt:type="ui2"/>
    <!--  I&M 0 Element 10 IM_SUPPORTED -->
    <AttributeType name="tagFunction" dt:type="string" dt:maxLength="32"/>
    <!-- I&M1 element 1 TAG_FUNCTION - mapped to semantic of IdTag -->
    <AttributeType name="tagLocation" dt:type="string" dt:maxLength="22"/>
    <!--  I&M1 element 2 TAG_LOCATION -->
```