Specification for PROFIBUS

Device Description and Device Integration

Volume 1: GSD

Version 5.1
July 2008

Order No: 2.122

Document Identification: TC4-05-0002b
File name: GSD-Spec_2122_V51_Jul08

Prepared by the PROFIBUS Working Group 2 “GSD Specification” in the Technical Committee 4 “System Integration”.

In this specification the following key words (in **bold** text) will be used:

may: indicates flexibility of choice with no implied preference.

should: indicates flexibility of choice with a strongly preferred implementation.

shall: indicates a mandatory requirement. Designers **shall** implement such mandatory requirements to ensure interoperability and to claim conformance with this specification.

Publisher:
PROFIBUS Nutzerorganisation e.V.
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 96 58 590
Fax: +49 (0) 721 / 96 58 589
E-mail: info@profibus.com
Web site: www.profibus.com

Contents

1 General Informations

1.1 Protocol of modifications

Date	Version	Author	Description
17.05.2004	5.02	A. Macher	Requirements for Slave Redundancy from TC 4 WG 4 worked in: Definition of keyword "Slave_Redundancy_supp" extended.
22.03.2005	5.04	A. Macher	PROFIsafe requirements worked in: New keyword "F_IO_StructureDescCRC" added.
21.07.2008	5.1	H. Oppmann	PROFIsafe requirements worked in: New keywords "Max_iParameter_Size" and "F_IO_StructureDescVersion"

1.2 Abbreviations

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASE	Application Service Elements
ASIC	Application Specific Integrated Circuit
Cfg	Configuration Identifier
CRC	Cyclic Redundancy Check
DIB	Device Indipendent Bitmap (same as Windows Bitmap), a raster graphic file format
DP	Decentralized Peripherals
DxB	Data-eXchange Broadcast (lateral slave communication)
EN	European Norm
FMS	Fieldbus Message Specification
GSD	General Station Description
HART	Highway Addressable Remote Transducer
HMD	HART Master Device
HMI	Human Machine Interface (= MMI)
ID	Identification
IEC	International Electrotechnical Commission
I&M	Identification and Maintenance
I/O	Input / Output
ISO	International Standardization Organisation
IsoM	Isochronous Mode
LAS	List of Active Stations
LSB	Least Significant Bit
MBP	Manchester coded Bus Powered
MBP-IS	MBP intrinsic safety
MBP-LP	MBP low power
MMI	Man Machine Interface (= HMI)
NC	Numerical Control
PA	Process Automation
PCF	Polymer Clad Fibre
PNO	PROFIBUS Nutzerorganisation e.V.
Prm	Parameter Assignment
PROFIBUS	Process Field Bus
RC	Robot Control
RS485	Radio Sector 485 standard, also known as EIA-485
TIA	Telecommunications Industry Association
TTL	Transistor-Transistor-Logic

2 Management Summery – Scope of this Document

Configuration tools currently available for PROFIBUS devices, which comply with IEC 61784-1:2003 CP3/1 and CP3/2 use a specially formatted ASCII file, referred to as General Station Description (GSD) file, which provides information about a device for example:

information needed to identify the connected device,
description of device data that can be accessed via the network (e.g. configurable parameters),
description of the communication capabilities supported by the device (e.g. transmission rate),
additional vendor specific information.

GSD objects, syntax and semantic are specified in clauses 3, 4, and Annex A.
The GSD allows a configuration tool to automate the device configuration process. The GSD requirements provide an open, consistent and compatible approach for performing device configuration.
All devices with a communication interface according IEC 61784-1 CP3/1 and CP3/2 shall have a GSD file. The main intention of GSD is to provide device information on a PROFIBUS communication network. In this document the name PROFIBUS DP and the acronym DP is used for the protocol and services of devices, which are compliant with IEC 61784-1 CP3/1 and CP3/2.

PROFIBUS devices may have different behavior and performance characteristics. Features differ in regard to available functionality (i.e., number of I/O signals and diagnostic messages) or possible bus parameters such as baud rate and time monitoring. These parameters may vary individually for each device type and vendor and are usually documented in the technical manual. In order to achieve a simple Plug and Play configuration for PROFIBUS devices, electronic device data sheets (GSD files) are defined to describe the communication features of the devices. These are named General Station Description (GSD) files, which allow easy configuration of PROFIBUS networks with devices from different manufacturers.

NOTE a synonym for GSD is "Communication Feature List", see IEC 61784-1:2003 Table 111.

GSD is a human readable ASCII text file. Clause 4 specifies keywords as mandatory or optional with the corresponding data type and their border values to support the configuration of PROFIBUS devices.
The GSD files characterize the features and performance capabilities of PROFIBUS devices.

Each vendor of a DP-Slave or a DP-Master (class 1) shall offer the characteristic features of the device as a device data sheet and a GSD file to the user. Using this information

enables the user to check all data in the configuration phase of a PROFIBUS system and errors can be avoided as early as possible. Based on the defined file format in clause 3, 4, and Annex A, it is possible to realize vendor independent configuration tools for PROFIBUS systems. The configuration tool uses the GSD files for testing the data. These were entered regarding the observance of limits and validity related to the performance of the individual device.

The distinction of the GSD files is achieved by the vendor- and device-identifiers.

In the case of a device that supports the PROFIBUS DP protocol and another protocol (e.g. PROFIBUS FMS), the other specific device data base information shall be located at the beginning of the GSD file.

NOTE ISO 15745-3 only describes the GSD for PROFIBUS DP.

The manufacturer of a device is responsible for the functionality and the quality of its GSD file. The device certification procedure is requesting either a standard GSD file based on a PROFIBUS profile or a device specific GSD file.
GSD fulfill the requirements of a communication network profile.
GSD file format is specified in 3. The GSD objects, syntax and semantic are specified in clause 4 and Annex A. The evolution of releases is described in Annex B.

List of affected patents

There is no affected patent known by the members of the Working Group. The list is empty. No patent search, neither external nor internal, has been done by the members of the Working Group up to now. PROFIBUS International does not guarantee the completeness of this list.

Requirement for certification tests

The General Station Description (GSD) file shall be checked according to the functionality of the device and the actual specification of the file. This check is precondition for doing the projecting of the PROFIBUS master and thus the interoperability testing.
A certification test has to ensure that a GSD file of Version 5.1 follows all “shall” rules that are specified in this document. This test can be processed with the check function of the PROFIBUS GSD Editor which is downloadable on the PNO webserver or by manually inspecting the GSD file.

3 Syntax and format of the GSD files

The GSD file shall be an ASCII file and it may be created with every applicable ASCII text editor. The DP-specific part shall begin with the identifier "#PROFIBUS-DP". The device data base shall be specified as parameter of a keyword. At the evaluation of the keywords the kind of letters, capital or small, are irrelevant.

NOTE The data medium, which the vendor of the DP-device uses for the delivery of the GSD file, is not defined here.

The file format shall be line oriented. Each line shall contain statements for exactly one parameter. If a semicolon is detected during the interpretation of the line, it is assumed that the rest of the line is a comment. The maximum number of characters per line shall be fixed to 80. If it is not possible to describe the information in one line, then it is allowed to use continuation lines. A "\" at the end of a line indicates that the following line is a continuation line. It is distinguished between number-parameters and text-parameters. No special end-identifier is defined. But it is to be ensured that the file ends after a complete line. Parameters, which are not used for a DP-Master or a DP-Slave, shall be omitted.

NOTE PROFIBUS-Master and PROFIBUS-Slave means devices, which are compliant with IEC 61784-1:2003 CP 3/1 or 3/2, see 7.2.2.1.2ff.

A GSD file should be created and provided to the user in the respective language. At least a default version (GS**D**) in English language is to be created. The language dependent files may only differ in the parameters of the type Visible-String and the Slave_Family. The language dependent device description data files differ regarding the last letter of the extension (*.gs?).

Default:	?=d
English:	?=e
French:	?=f
German:	?=g
Italian:	?=i
Portuguese:	?=p
Spanish:	?=s

General specifications

This section in the GSD file shall contain information on vendor and device names, hardware and software release states, supported baud rates, possible time intervals for monitoring times and the signal assignment on the bus connector.

Master-related specifications

This section in the GSD file shall contain all master-related parameters, such as: the maximum number of slaves that can be connected, or upload and download options. This section does not exist for slave devices.

Slave-related specifications

This section in the GSD file shall contain all slave-related specifications, such as the number and type of I/O channels, specification of diagnostic texts and information on the available modules with modular devices. In the individual sections, the parameters are separated by keywords. A distinction is made between mandatory parameters (i.e., Vendor_Name) and optional parameters (i.e., Sync_Mode_supp). The definition of parameter groups allows selection of options. In addition, bit map files with the symbols of the devices can be integrated. The format of the GSD is designed for flexibility. It contains both lists (such as the baud rates supported by the device) as well as space to describe the modules available in a modular device. Plain text can also be assigned to the diagnostic messages. This section does not exist for master devices.

4 Semantic and coding of the keywords

4.1 Conventions

The type ID specified for the keywords shall refer to the parameters with the same name. In the case of the parameters, a differentiation shall be made between:

- Mandatory (M): absolutely required
- Optional (O): possible in addition
- Default (D): Optional with default = 0 if not present
- Grouped (G): At least one keyword of the group is required

Expansions of the released GSD specifications (for example, new keywords) are provided in this document with a version ID (GSD_Revision) that indicates the version where the expansion was added. Keywords without version ID belong to the original version.

The keywords are classified in:

General specifications, see 4.2
Master-related specifications, see 4.3
Slave-related specifications, see 4.4

4.2 General specifications

4.2.1 General DP keywords

GSD_Revision: (M starting with GSD_Revision 1)
Version ID of the GSD file format.
Type: Unsigned8

Vendor_Name: (M)
Manufacturer's Name.
Type: Visible-String (32)

Model_Name: (M)
Manufacturer's designation (Controller Type) of device.
Type: Visible-String (32)

Revision: (M)
Revision version of the device.
Type: Visible-String (32)

Revision_Number: (O starting with GSD_Revision 1)
Version ID of the device. The value of the Revision_Number has to agree with the value of the Revision_Number in the slave-specific diagnosis.
Type: Unsigned8 (1 - 63)

Ident_Number: (M)
Device type of the device.
The Ident_Number is assigned by the PROFIBUS Nutzerorganisation e.V. (PNO) to each device type. Manufacturers of devices have to apply for the Ident_Number at the PNO.
Type: Unsigned16

Protocol_Ident: (M)
Protocol ID of the device.
Type: Unsigned8
0: PROFIBUS DP,
16 - 255: Manufacturer-specific

Station_Type: (M)
DP device type.
Type: Unsigned8
0: DP Slave,
1: DP Master (Class 1)

FMS_supp: (D)
This device is an FMS/DP mixed device.
Type: Boolean (1: True)

Hardware_Release: (M)
Hardware release of the device.
Type: Visible-String (32)

Software_Release (M)
Software release of the device.
Type: Visible-String (32)

9.6_supp: (G)
The device supports the baudrate 9.6 kbit/s.
Type: Boolean (1: True)

19.2_supp: (G)
The device supports the baudrate 19.2 kbit/s.
Type: Boolean (1: True)

31.25_supp: (G starting with GSD_Revision 2)
The device supports the baudrate 31.25 kbit/s.
Type: Boolean (1: True)

45.45_supp: (G starting with GSD_Revision 2)
The device supports the baudrate 45.45 kbit/s.
Type: Boolean (1: True)

93.75_supp: (G)
The device supports the baudrate 93.75 kbit/s.
Type: Boolean (1: True)

187.5_supp: (G)
The device supports the baudrate 187.5 kbit/s.
Type: Boolean (1: True)

500_supp: (G)
The device supports the baudrate 500 kbit/s.
Type: Boolean (1: True)

1.5M_supp: (G)
The device supports the baudrate 1.5 Mbit/s.
Type: Boolean (1: True)

3M_supp: (G starting with GSD_Revision 1)
The device supports the baudrate 3 Mbit/s.
Type: Boolean (1: True)

6M_supp: (G starting with GSD_Revision 1)
The device supports the baudrate 6 Mbit/s.
Type: Boolean (1: True)

12M_supp: (G starting with GSD_Revision 1)
The device supports the baudrate 12 Mbit/s.
Type: Boolean (1: True)

NOTE In order to secure the optimized performance of the publisher / subscriber functionality it is necessary to set the MaxTsdr_xx values according to the actual values of the device.

MaxTsdr_9.6: (G)
This is the time a responder needs as a maximum at a baudrate of 9.6 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_19.2: (G)
This is the time a responder needs as a maximum at a baudrate of 19.2 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_31.25: (G starting with GSD_Revision 2)
This is the time a responder needs as a maximum at a baudrate of 31.25 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_45.45: (G starting with GSD_Revision 2)
This is the time a responder needs as a maximum at a baudrate of 45.45 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_93.75: (G)
This is the time a responder needs as a maximum at a baudrate of 93.75 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_187.5: (G)
This is the time a responder needs as a maximum at a baudrate of 187.5 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_500: (G)
This is the time a responder needs as a maximum at a baudrate of 500 kbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_1.5M: (G)
This is the time a responder needs as a maximum at a baudrate of 1.5 Mbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_3M: (G starting with GSD_Revision 1)
This is the time a responder needs as a maximum at a baudrate of 3 Mbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_6M: (G starting with GSD_Revision 1)
This is the time a responder needs as a maximum at a baudrate of 6 Mbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

MaxTsdr_12M: (G starting with GSD_Revision 1)
This is the time a responder needs as a maximum at a baudrate of 12 Mbit/s to respond to a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned16
Time base: Bit Time

Redundancy: (D)
This value specifies whether a device supports redundant transmission engineering.
Type: Boolean
0: No, 1: Redundancy is supported.

Repeater_Ctrl_Sig: (D)
Here, the level of the bus connector signal CNTR-P is specified.
Type: Unsigned8
0: Not connected, 1: RS485, 2: TTL

24V_Pins: (D)
Here, the meaning of the bus connector signal M24V and P24V is specified.
Type: Unsigned8
0: Not connected, 1: Input, 2: Output

Implementation_Type: (O starting with GSD_Revision 1)
Here, a description is provided which standard implementation is used in the DP slave, for example, Standard Software, Controller or ASIC (Application Specific Integrated Circuit) solution. The manufacturer of the standard solution provides the name; the specification of that name shall be obeyed.
Type: Visible-String (32)

Bitmap_Device: (O starting with GSD_Revision 1)
Here, the file name of the bit map file (see NOTE) is specified that contains the symbolic representation of the device in standard cases.
Type: Visible-String (8)

Bitmap_Diag: (O starting with GSD_Revision 1)
Here, the file name of the bit map file (see NOTE) is specified that contains the symbolic representation of the device for diagnostic cases.
Type: Visible-String (8)

Bitmap_SF: (O starting with GSD_Revision 1)
Here, the file name of the bit map file (see NOTE) is specified that contains the symbolic representation of the device in special operating modes. The meaning is manufacturer-specific.
Type: Visible-String (8)

NOTE The file shall be in Windows Bitmap format, and have 70*40 pixels (width*height) in 16 colors.
The file name shall be given without path and extension. An extension of ".bmp" (Bitmap) is assumed. For backward compatibility, ".dib" (Device Indipendent Bitmap) is also allowed.

4.2.2 Additional keywords for different physical interfaces

Physical_Interface: (O starting with GSD_Revision 3)
This value specifies the execution of the Physical Layers of PROFIBUS. With this parameter it is possible to have devices with more than one physical interface or interfaces different from RS485. If this keyword is not used, then RS485 standard copper is the only supported physical interface. Between the keywords Physical_Interface and End_Physical_Interface, the Transmission_Delays and the Reaction_Delay of a slave device are specified, for the physical interface used in the device. The Transmission_Delay defines the delay time for the signal which is to be transmitted through the device. The Reaction_Delay defines the delay of signals processed by the device.

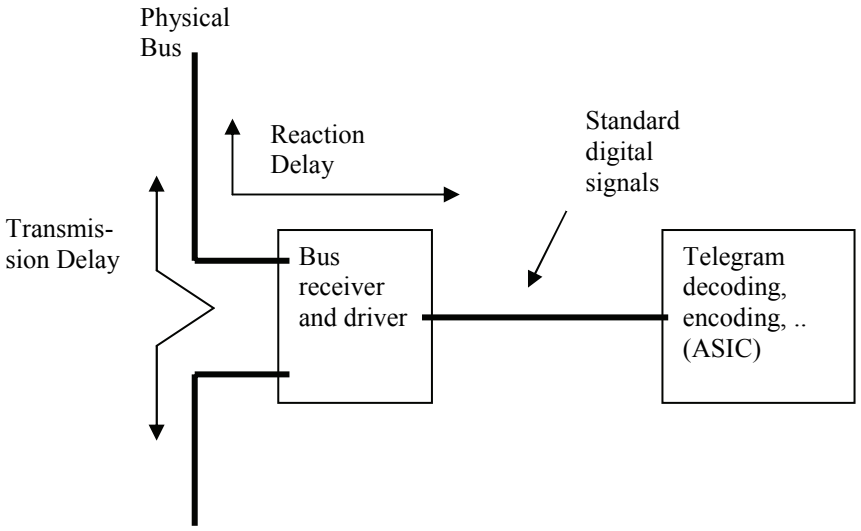


Figure 1 — Physical_Interface example

EXAMPLE The Transmission_Delay with RS485 is 0, the Reaction_Delay is also 0, because the delay in the driver is lower than 1 bit time, see Figure 1.

Especially with optical interfaces these parameters are necessary for the bus timing calculation.

Both the Transmission_Delay and the Reaction_Delay has to be defined for each supported baudrate. Otherwise the baudrate is not valid for this physical layer.

Coding of the interfaces:

Type: Unsigned8

0: RS485 (ANSI TIA/EIA RS-485-A); optional RS485-intrinsic safety version (see [2])
1: Manchester coded and bus powered (MBP); optional intrinsic safety (MBP-IS) and lower power (MBP-LP)
2: Plastic fibre
3: Glass multi mode fibre or Glass single mode fibre
4: Polymer Clad Fibre (PCF)
5-127: Reserved
128-255: Manufacturer specific

Parameters Used:

Transmission_Delay_9.6: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_19.2: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_31.25: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_45.45: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_93.75: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_187.5: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_500: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_1.5M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_3M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_6M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_12M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Reaction_Delay_9.6: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_19.2: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_31.25: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_45.45: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_93.75: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_187.5: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_500: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_1.5M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_3M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_6M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_12M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

4.3 Master-related specifications

4.3.1 DP Master (Class 1) related keywords

Master_Freeze_Mode_supp: (D starting with GSD_Revision 3)
The device supports the Freeze mode.
Type: Boolean (1: True)

Master_Sync_Mode_supp: (D starting with GSD_Revision 3)
The device supports the Sync mode.
Type: Boolean (1: True)

Master_Fail_Safe_supp: (D starting with GSD_Revision 3)
The device supports the Fail Safe.
Type: Boolean (1: True)

Download_supp: (D)
The device supports the functions Download, Start_seq and End_seq.
Type: Boolean (1: True)

Upload_supp: (D)
The device supports the functions Upload, Start_seq and End_seq.
Type: Boolean (1: True)

Act_Para_Brct_supp: (D)
The device supports the function Act_Para_Brct.
Type: Boolean (1: True)

Act_Param_supp: (D)
The device supports the function Act_Param.
Type: Boolean (1: True)

Max_MPS_Length: (M)
Maximum memory size (in bytes) that a device makes available for storing the master parameter set.
Type: Unsigned32

Max_Lsdu_MS: (M)
Here, the maximum L_sdu length for all master-slave communication relations is specified.
Type: Unsigned8

Max_Lsdu_MM: (M)
Here, the maximum L_sdu length for the master-master communication relations is specified.
Type: Unsigned8

Min_Poll_Timeout: (M)
This value indicates how long a DP master (Class 1) needs as a maximum for processing a master-master function.
Type: Unsigned16
Time base: 10 ms

Trdy_9.6: (G)
This value indicates how fast a DP master (Class 1), at a baudrate of 9.6 kbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_19.2: (G)
This value indicates how fast a DP master (Class 1), at a baudrate of 19.2 kbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_31.25: (G starting with GSD_Revision 2)
This value indicates how fast a DP master (Class 1), at a baudrate of 31.25 kbit/s is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_45.45: (G starting with GSD_Revision 2)
This value indicates how fast a DP master (Class 1), at a baudrate of 45.45 kbit/s is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_93.75: (G)
This value indicates how fast a DP master (Class 1), at a baudrate of 93.75 kbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_187.5: (G)
This value indicates how fast a DP master (Class 1), at a baudrate of 187.5 kbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_500: (G)
This value indicates how fast a DP master (Class 1), at a baudrate of 500 kbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_1.5M: (G)
This value indicates how fast a DP master (Class 1), at a baudrate of 1.5 Mbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_3M: (G starting with GSD_Revision 1)
This value indicates how fast a DP master (Class 1), at a baudrate of 3 Mbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_6M: (G starting with GSD_Revision 1)
This value indicates how fast a DP master (Class 1), at a baudrate of 6 Mbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Trdy_12M: (G starting with GSD_Revision 1)
This value indicates how fast a DP master (Class 1), at a baudrate of 12 Mbit/s, is ready to receive again after sending a request message (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tqui_9.6: (G)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E at a baudrate of 9.6 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_19.2: (G)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 19.2 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_31.25: (G starting with GSD_Revision 2)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 31.25 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_45.45: (G starting with GSD_Revision 2)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 45.45 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_93.75: (G)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 93.75 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_187.5: (G)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 187.5 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_500: (G)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 500 kbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_1.5M: (G)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 1.5 Mbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_3M: (G starting with GSD_Revision 1)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 3 Mbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_6M: (G starting with GSD_Revision 1)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 6 Mbit/s.
Type: Unsigned8
Time base: Bit Time

Tqui_12M: (G starting with GSD_Revision 1)
This value specifies the modulator fading time (T_{QUI}), (refer to IEC 61158-4:2003 Annex E) at a baudrate of 12 Mbit/s.
Type: Unsigned8
Time base: Bit Time

Tset_9.6: (G)
This value specifies the trigger time, at the baudrate of 9.6 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_19.2: (G)
This value specifies the trigger time, at the baudrate of 19.2 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_31.25: (G starting with GSD_Revision 2)
This value specifies the trigger time, at the baudrate of 31.25 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_45.45: (G starting with GSD_Revision 2)
This value specifies the trigger time, at the baudrate of 45.45 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_93.75: (G)
This value specifies the trigger time, at the baudrate of 93.75 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_187.5: (G)
This value specifies the trigger time, at the baudrate of 187.5 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_500: (G)
This value specifies the trigger time, at the baudrate of 500 kbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_1.5M: (G)
This value specifies the trigger time, at the baudrate of 1.5 Mbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_3M: (G starting with GSD_Revision 1)
This value specifies the trigger time, at the baudrate of 3 Mbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_6M: (G starting with GSD_Revision 1)
This value specifies the trigger time, at the baudrate of 6 Mbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

Tset_12M: (G starting with GSD_Revision 1)
This value specifies the trigger time, at the baudrate of 12 Mbit/s, in reference to Layer2 (setup time) from the arrival of an event until the corresponding response (refer to IEC 61158-4:2003 Annex E).
Type: Unsigned8
Time base: Bit Time

LAS_Len: (M)
This value indicates how many entries the device in question can manage in the list of active stations (LAS).
Type: Unsigned8

Tsdi_9.6: (G)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 9.6 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_19.2: (G)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 19.2 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_31.25: (G starting with GSD_Revision 2)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 31.25 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_45.45: (G starting with GSD_Revision 2)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 45.45 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_93.75: (G)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) of the initiator at a baudrate of 93.75 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_187.5: (G)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 187.5 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_500: (G)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 500 kbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_1.5M: (G)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 1.5 Mbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_3M: (G starting with GSD_Revision 1)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 3 Mbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_6M: (G starting with GSD_Revision 1)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 6 Mbit/s.
Type: Unsigned16
Time base: Bit Time

Tsdi_12M: (G starting with GSD_Revision 1)
This value specifies the station delay time (Tsdi) of the initiator (refer to IEC 61158-4:2003 Annex E) at a baudrate of 12 Mbit/s.
Type: Unsigned16
Time base: Bit Time

Max_Slaves_supp: (M)
This value indicates how many DP slave stations a DP master (Class 1) can handle.
Type: Unsigned8

Max_Master_Input_Len: (O starting with GSD_Revision 1)
Here, the maximum length of the input data per DP slave is specified that the DP master supports.
Type: Unsigned8

Max_Master_Output_Len: (O starting with GSD_Revision 1)
Here, the maximum length of the output data per DP slave is specified that the DP master supports.
Type: Unsigned8

Max_Master_Data_Len: (O starting with GSD_Revision 1)
Here, the sum of the lengths of the output and input data per DP slave is specified that the DP master supports. If this keyword is not provided, the maximum length will be the sum of the input and output data.
Type: Unsigned16

4.3.2 Additional master related keywords for DP extensions

DPV1_Master: (D starting with GSD_Revision 3)
The DP master supports DP-V1 extensions of the DP protocol.
Type: Boolean (1: True)

DPV1_Conformance_Class: (M if DPV1_Master, starting with GSD_Revision 3)
This value specifies the Conformance Class of the DP-Master (Class1). The following Conformance Classes are specified for DP-Master (Class 1):
Type: Unsigned8

1:	Conformance Class A
2:	Conformance Class B
0,3 - 255:	reserved

C1_Master_Read_Write_supp: (D starting with GSD_Revision 3)
The DP-Master (Class 1) supports the Read and Write services on the C1-communication relationship.
Type: Boolean (1: True)

Master_DPV1_Alarm_supp: (D starting with GSD_Revision 3)
The DP-Master (Class 1) supports alarms.
Type: Boolean (1: True)

Master_Diagnostic_Alarm_supp: (G if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The device supports Diagnostic_Alarm. A diagnostic alarm signals an event within a slot, for instance overtemperature, short circuit, etc..
Type: Boolean (1: True)

Master_Process_Alarm_supp: (G if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The device supports Process_Alarm. A process alarm signals the occurrence of an event in the connected process, for instance upper limit value exceeded.
Type: Boolean (1: True)

Master_Pull_Plug_Alarm_supp: (G if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The device supports Pull_Alarm. A pull alarm signals the withdrawal of a module at a slot.
Type: Boolean (1: True)

Master_Status_Alarm_supp: (G if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The device supports Status_Alarm. A status alarm signals a change in the state of a module, for instance run, stop or ready.
Type: Boolean (1: True)

Master_Update_Alarm_supp: (G if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The device supports Update_Alarm. An update alarm signals the change of a parameter in a slot e.g. by a local operation or remote access.
Type: Boolean (1: True)

Master_Manufacturer_Specific_Alarm_supp: (G if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The device supports Manufacturer_Specific _Alarm. A manufacturer specific alarm signals an event defined by the manufacturer.
Type: Boolean (1: True)

Master_Extra_Alarm_SAP_supp: (D if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
In addition to SAP 51 it is possible to handle the MSAL_Alarm_Ack via SAP 50 if the Bit Sl_Flag.Extra_Alarm_SAP in the corresponding slave parameter set is set. In this case there may be a higher performance because SAP 50 is used exclusively for the MSAL_Alarm_Ack service and the service can not be delayed by a running MSAC1_Write or MSAC1_Read service.
Type: Boolean (1: True)

Master_Alarm_Sequence_Mode: (M if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The DP master supports the Alarm Sequence Mode with the specified number of alarms for alarm handling.
The Sequence Mode is an option of parallel alarm handling.
Several alarms (2 – 32) of the same or different type can be active at one time (fixed by the DDLM_Set_Prm service).
Type: Unsigned8
0: Sequence_Mode not supported
1: 2 alarms in total
2: 4 alarms in total
3: 8 alarms in total
4: 12 alarms in total
5: 16 alarms in total
6: 24 alarms in total
7: 32 alarms in total

Master_Alarm_Type_Mode_supp: (M if Master_DPV1_Alarm_supp, starting with GSD_Revision 3)
The DP master supports the Alarm Type Mode.
The Type Mode is mandatory if the DP-Master supports parallel alarm handling.
One alarm of each type can be active at one time (fixed by the DDLM_Set_Prm service).
Type: Boolean (shall always be set to 1: True)

4.3.3 Additional master related keywords for DP-V2

Isochron_Mode_Synchronised (D starting with GSD_Revision 4):
This parameter indicates whether a master device has the capability to run in the Isochron_Mode and which model it does support. Therefore, the following 4 values are allowed:
Type: Unsigned8
0: Master device does not support the Isochron_Mode
1: Master device supports only the buffer synchronized Isochron_Mode (refer to IEC 61158-5:2003, 8.2.2.4.3.2)
2: Master device supports only the enhanced synchronized Isochron_Mode (refer to IEC 61158-5:2003, 8.2.2.4.3.3)
3: Master device supports both, the buffer synchronized and the enhanced synchronized Isochron_Mode.
NOTE For further information about the functionality of the Isochron_Mode see [5].

DXB_Master_supp: (D starting with GSD_Revision 4)
The DP-Master supports the service of Data Exchange with Broadcast.
Type: Boolean (1: True)

X_Master_Prm_SAP_supp: (D starting with GSD_Revision 4)
Indicates, if the X_Prm_SAP of the slave can be addressed by the master. Shall only be true, if DPV1_Master = 1 and if the master supports structured parameterization data.
Type: Boolean (1: True)

4.4 Slave-related specifications

4.4.1 Basic DP-Slave related keywords

Freeze_Mode_supp: (D)
The DP device supports the Freeze mode. DP slaves that support the Freeze mode have to guarantee that in the next data cycle after the Freeze control command, the values of the inputs that were frozen last are transferred to the bus.
Type: Boolean (1: True)

Sync_Mode_supp: (D)
The DP device supports the Sync mode.
Type: Boolean (1: True)

Auto_Baud_supp: (D)
The DP device supports automatic baudrate recognition.
Type: Boolean (1: True)

Set_Slave_Add_supp: (D)
The DP device supports the function Set_Slave_Add.
Type: Boolean (1: True)

User_Prm_Data_Len: (D)
Here, the length of User_Prm_Data is specified. The amount of data of User_Prm_Data has to agree with this parameter.
Type: Unsigned8

User_Prm_Data: (O)
Manufacturer-specific field. Specifies the default value for User_Prm_Data. If this parameter is used, its length has to agree with the User_Prm_Data_Len.
Type: Octet-String

Min_Slave_Intervall: (M)
This time specifies the minimum interval between two slave list cycles for the DP device.
Type: Unsigned16
Time base: 100 µs

Modular_Station: (D)
Here it is specified whether the DP device is a modular station.
It´s strongly recommended to model slaves in the following way:
A compact device has only one module with all configuration identifiers. A modular device has only one configuration identifier in each module definition. When a slave accepts only one configuration identifier selected from a number of possible configurations, then the slave should be a modular station with Max_Module =1.

Type: Boolean

0: compact device

1: modular device

Max_Module: (M if Modular_Station)
Here, the maximum number of modules of a modular station is specified.

Type: Unsigned8

Max_Input_Len: (M if Modular_Station)
Here, the maximum length of the input data of a modular station is specified in bytes.

Type: Unsigned8

Max_Output_Len: (M if Modular_Station)
Here, the maximum length of the output data of a modular station is specified in bytes.

Type: Unsigned8

Max_Data_Len: (O only if Modular_Station)
Here, the largest sum of the lengths of the output and input data of a modular station is specified in bytes. Max_Data_Len shall be in minimum the highest value of Max_Input_Len and Max_Output_Len, in maximum the sum of both. If this keyword is not provided, the maximum length is the sum of all input and output data.

Type: Unsigned16

EXAMPLE 1
Max_Input_Len = 24
Max_Output_Len = 30
Max_Data_Len = 30 (minimum)
EXAMPLE 2
Max_Input_Len = 120
Max_Output_Len = 120
Max_Data_Len = 200
EXAMPLE 3
Max_Input_Len = 240
Max_Output_Len = 240
Max_Data_Len = 480 (maximum)

(X_)Unit_Diag_Bit: (O, X_ starting with GSD_Revision 4)
In order to display manufacturer-specific status- and error messages of a DP slave centrally, it is possible to assign to a bit a text (Diag_Text) in the device-related diagnostic field if the bit value equals 1.
Parameters used:

Bit

Type: Unsigned16 (0 - 495, X_ 24 - 495)
Meaning: Bit position in device-related diagnostic field (LSB in first byte is Bit 0).

Diag_Text:

Type: Visible-String (32)

(X_)Unit_Diag_Bit_Help: (O starting with GSD_Revision 5)
Here additional information about the manufacturer-specific status- and error messages is defined. The configuration tool can offer this information to the user additional to the Diag_Text of the (X_)Unit_Diag_Bit corresponding bit position.
Parameters used:

Bit

Type: Unsigned16 (0 - 495, X_ 24 - 495)
Meaning: Bit position in device-related diagnostic field (LSB in first byte is Bit 0).

Help_Text:

Type: Visible-String (256)

(X_)Unit_Diag_Not_Bit: (O starting with GSD_Revision 4)
In order to display manufacturer-specific status- and error messages of a DP slave centrally, it is possible to assign to a bit a text (Diag_Text) in the device-related diagnostic field if the bit value equals 0.
Parameters used:

Bit

Type: Unsigned16 (0 - 495, X_ 24 - 495)
Meaning: Bit position in device-related diagnostic field (LSB in first byte is Bit 0).

Diag_Text:

Type: Visible-String (32)

(X_)Unit_Diag_Not_Bit_Help: (O starting with GSD_Revision 5)
Here additional information about the manufacturer-specific status- and error messages is defined. The configuration tool can offer this information to the user additional to the Diag_Text of the (X_)Unit_Diag_Not_Bit corresponding bit position.
Parameters used:

Bit

Type: Unsigned16 (0 - 495, X_ 24 - 495)
Meaning: Bit position in device-related diagnostic field (LSB in first byte is Bit 0).

Help_Text:

Type: Visible-String (256)

(X_)Unit_Diag_Area: (O, X_ starting with GSD_Revision 4)
Between the keywords (X_)Unit_Diag_Area and (X_)Unit_Diag_Area_End, the assignment of values in a bit field in the device-related diagnostic field to texts (Diag_Text) is specified.
Parameters used:

First_Bit:
Type: Unsigned16
Meaning: First bit position of the bit field (LSB in the first byte is Bit 0)

Last_Bit:
Type: Unsigned16 (0<=First_Bit<=Last_Bit<=495,
X_ 24<=First_Bit<=Last_Bit<=495)
Meaning: Last bit position of the bit field. The bit field may be 16 bits wide maximum.

(X_)Value:, (X_ starting with GSD_Revision 4)
Type: Unsigned16
Meaning: Value in the bit field

Diag_Text:
Type: Visible-String (32)

(X_)Value_Help: (O starting with GSD_Revision 5)
Type: Unsigned16
Meaning: Value in the bit field

Help_Text:
Type: Visible-String (256)

UnitDiagType: (O starting with GSD_Revision 4)
Between the keywords UnitDiagType and EndUnitDiagType, different structures within the Unit-Diag can be described. This is meaningful especially for DP-V1 slaves. Only the keywords starting with "X_" are allowed. The counting starts with octet 2, the first bit of the type (see also, Figure 2). The first bit to be defined is Bit24, the first bit of the Diagnosis_User_Data in octet 5 (see also Figure 3 and Figure 4. Description of Diagnosis_User_Data see IEC 61158-6:2003 Table 396, in row Device_Related_Diagnosis).

Octet	Name	7	6	5	4	3	2	1	0
1	Header_ Octet								
2	Type	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
3	Slot	*15*	*14*	*13*	*12*	*11*	*10*	*9*	*8*
4	Specifier	*23*	*22*	*21*	*20*	*19*	*18*	*17*	*16*
5	Diagnosis_User_ Data (0..59 Byte)	31	30	29	28	27	26	25	24
5		39	38	37	36	35	34	33	32
6						..	42	41	40
:						:			

Figure 2 — Counting of UnitDiagType

Parameters used:

Diag_Type_Number:

Type: Unsigned8

Meaning: Defines, if an alarm block (0 – 127) or a status block (128 – 255) is described.

EXAMPLE 4

```
UnitDiagType            = 161
X_Unit_Diag_Bit(40)     = "TDP_error"
X_Unit_Diag_Bit(41)     = "TDX_error"
X_Unit_Diag_Bit(42)     = "TSYNC_Prm_Fault"
X_Unit_Diag_Area        = 57-63
X_Value(1)              = "Error 1"
X_Value_Help(1)         = "Please correct ...."
X_Value(10)             = "Error 10"
X_Value_Help(10)        = "Please correct ...."
X_Unit_Diag_Area_End
EndUnitDiagType
```

Figure 3 illustrates the coding of a diagnosis type Alarm, which can be described by a UnitDiagType.

Octet	Name	7	6	5	4	3	2	1	0
1	Header_ Octet	0	0	Block_Length (4..63)					
2	Type	0	Alarm_Type						
3	Slot	Slot_Number (0..244)							
4	Specifier	Sequence_Number					Add_ Ack	Alarm_ Specifier	
5 – length		Diagnosis_User_Data (0..59 Byte)							

Figure 3 — coding of a diagnosis type alarm

The following Alarm types are defined:

0	reserved
1	Diagnostic_Alarm
2	Process_Alarm
3	Pull_Alarm
4	Plug_Alarm
5	Status_Alarm
6	Update_Alarm
7 – 31	reserved
32 – 126	manufacturer-specific
127	reserved

Figure 4 illustrates the coding of a diagnosis type status, which can be described by a UnitDiagType too.

Octet	Name	7	6	5	4	3	2	1	0
1	Header_ Octet	0	0	Block_Length (4..63)					
2	Type	1	Status_Type						
3	Slot	Slot_Number (0..244)							
4	Specifier	Reserved						Status_ Specifier	
5 – length		Diagnosis_User_Data (0..59 Byte)							

Figure 4 — coding of a diagnosis type status

The following Status types are defined

0	reserved
1	Status_Message
2	Module_Status
3	DXB_Link_Status
4 – 29	reserved
30	PrmCmdAck
31	Red_State
32 – 126	manufacturer-specific
127	reserved

Module: (M)
Between the keywords Module and EndModule, the IDs of a DP compact device or the IDs of all possible modules of a modular slave are specified, manufacturer-specific error types are specified in the channel-related diagnostic field, and the User_Prm_Data is described. If, in the case of modular slaves, empty slots are to be defined as empty module (ID/s 0x00), the empty module has to be defined. Otherwise, empty slots would not appear in the configuration data.
If the keyword Channel_Diag is used outside the keywords Module and EndModule, the same manufacturer-specific error type is specified in the channel-related diagnostic field for all modules. Channel_Diag definitions for a manufacturer specific error type inside a module will overwrite the definition for this error type defined for the device. Channel_Diag inside a module do not influence other modules.
If the keywords Ext_User_Prm_Data_Ref or Ext_User_Prm_Data_Const (X_Ext_User_Prm_Data_Ref or X_Ext_User_Prm_Data_Const) are used outside the keywords Module and EndModule, the associated User_Prm_Data area refers to the entire device, and the data in the parameter offset to the entire User_Prm_Data. This User_Prm_Data area has to be at the start of the User_Prm_Data.
The module-specific User_Prm_Data is directly attached to the device-specific User_Prm_Data in the sequence in which the associated modules were configured. If the keywords Ext_User_Prm_Data_Ref or Ext_User_Prm_Data_Const (X_Ext_User_Prm_Data_Ref or X_Ext_User_Prm_Data_Const / F_Ext_User_Prm_Data_Ref or F_Ext_User_Prm_Data_Const) are used within the keywords Module and EndModule, the data in the parameter offset refers only to the start of the User_Prm_Data area that is assigned to this module.

Parameters used:

Mod_Name:
Type: Visible-String (32)
Meaning: Module name of a module used in a modular DP station, or device name of a compact DP slave. This name shall be unique for a device (same Ident_Number).

Config:
Type: Octet-String (17)
Type: Octet-String (244) (O starting with GSD_Revision 1)
Meaning: Here, the ID or IDs of the module of a modular DP slave or of a compact DP device are specified.
Note that for PROFIsafe modules (see [4]) only a limited range of data types is allowed.

Module_Reference: (O starting with GSD_Revision 1,
M starting with GSD_Revision 3)
Type: Unsigned16
Meaning: Here, the reference of the module description is specified. This reference shall be unique for a device (same Ident_Number). This referencing is useful in order to make language-independent configuring possible in a language-dependent system, or to recognize modules.

Ext_Module_Prm_Data_Len: (O starting with GSD_Revision 1)
Type: Unsigned8
Meaning: Here, the length of the associated User_Prm_Data is defined.

X_Ext_Module_Prm_Data_Len: (O starting with GSD_Revision 4)
Type: Unsigned8 (1 – 244)
Meaning: Here, the length of the associated User_Prm_Data for the X_Prm_SAP is defined.

F_Ext_Module_Prm_Data_Len: (O starting with GSD_Revision 4)
Type: Unsigned8 (1 – 237)
Meaning: Here, the length of the ExtUserPrmData for the F- module is defined.

Data_Area: (O starting with GSD_Revision 5)
Between the keywords Data_Area_Beg and Data_Area_End, the input and output areas of the module can be specified. The description always begins with the first area and rise without gaps.

Area_Name: (M between a Data_Area, starting with GSD_Revision 5)
Type: Visible-String (32)
Meaning: Name of the area who is described.

Related_CFG_Identifier: (M between a Data_Area, starting with GSD_Revision 5)
Type: Unsigned8
Meaning: Index of the CFG ID byte, begins with 1, even if only one CFG-Identifier exists.

IO_Direction: (M between a Data_Area, starting with GSD_Revision 5)
Type: Boolean
0: Input
1: Output
Meaning: Direction of the described Data_Area, Input or Output.

Length: (M between a Data_Area, starting with GSD_Revision 5)
Type: Unsigned8 (1 – 244)
Meaning: Length of the Data_Area in bytes.

Consistency: (M between a Data_Area, starting with GSD_Revision 5)
Type: Unsigned8
0: Consistency only for the given Data_Types of the Data_Area
1: Consistency of the whole Data_Area
Meaning: Demanded is either the consistency of the given Data_Type or of the whole Data_Area. The CFG ID has to have the same level of the consistency or one level higher.

Publisher_allowed: (M between a Data_Area, starting with GSD_Revision 5)
Type: Boolean (1: True)
Meaning: Data_Area is valid for received Publisher data, i.e. when IO_Direction=1 (Output), and if Subscriber_supp=1. If DP_Master_allowed=0, Publisher_allowed shall be True.

DP_Master_allowed: (M between a Data_Area, starting with GSD_Revision 5)
Type: Boolean (1: True)
Meaning: Data_Area is valid for received Master data, i.e. when IO_Direction=1 (Output). If Publisher_allowed=0, DP_Master_allowed shall be True.
See also the notes given for EXAMPLE 6.

Data_Type: (M between a Data_Area, starting with GSD_Revision 5)
Type: Unsigned8
Meaning: Specifies the Data_Types. This value complies with the standard data type specification in IEC 61158-6. One or more data types are possible, i.e. U8 or Float (Idx. 5,8) at PA.

EXAMPLE 5 (Drive)

```
Module = "Standard telegram 3" 0xC3,0xC4,0xC8,0xFD,0x00,0x03
; First Data_Area
Data_Area_Beg
Area_Name               = "Control words, speed setpoint"
Related_CFG_Identifier  = 1
IO_Direction            = 1  ;Output
Length                  = 10
Consistency             = 1
Publisher_allowed       = 1
DP_Master_allowed       = 1
Data_Type               = 6  ;Unsigned16
Data_Type               = 4  ;Integer 32
Data_Type               = 6  ;Unsigned16
Data_Type               = 6  ;Unsigned16
Data_Area_End
; Second Data_Area
Data_Area_Beg
Area_Name               = "Status words, actual values"
Related_CFG_Identifier  = 1
IO_Direction            = 0  ;Input
Length                  = 18
Consistency             = 1
Publisher_allowed       = 0
DP_Master_allowed       = 0
Data_Type               = 6  ;Unsigned16
Data_Type               = 4  ;Integer 32
Data_Type               = 6  ;Unsigned16
Data_Type               = 6  ;Unsigned16
```

```
        Data_Type               = 4  ;Integer 32
        Data_Type               = 4  ;Integer 32
        Data_Area_End
        ; End Data_Area
        EndModule
```

EXAMPLE 6 (Drive)

```
        Module = "Slave-to-slave, PD-1" 0x81,0xC0,0xF9
        Info_Text = "Slave-to-slave, Receive, PD length 1 word"
        Data_Area_Beg
        Area_Name               = "Slave-to-slave"
        Related_Cfg_Identifier  = 1
        IO_Direction            = 1  ;Output
        Length                  = 2
        Consistency             = 1
        Publisher_allowed       = 1
        DP_Master_allowed       = 0
        Data_Type               = 6  ;Unsigned16
        Data_Area_End
        EndModule
```

NOTE When Publisher_allowed=1 and DP_Master_allowed=0 (or Publisher_allowed=1 and DP_Master_allowed=1 and the master is disabled in the configuration tool, i.e. the slave acts only in subscriber mode), then the configuration tool shall only take the manufacturer specific data of the config identifier. The master-slave data length (outputs) must be set to zero. In this example, the valid config identifier that the master must send would be "0x01, 0xF9" (or, alternatively, an empty slot using "0x00").

EXAMPLE 7 (PROFIBUS PA device)

```
        Module = "READBACK + POS_D, SP"     0xC6,0x84,0x86,0x08, /
                                            0x05,0x08,0x05,0x05,0x05
        ; First Data_Area
        Data_Area_Beg
        Area_Name               = "Outputs"
        Related_CFG_Identifier  = 1
        IO_Direction            = 1  ;Output
        Length                  = 5
        Consistency             = 1
        Publisher_allowed       = 1
        DP_Master_allowed       = 1
        Data_Type               = 8  ;Floating Point 32
        Data_Type               = 5  ;Unsigned8
        Data_Area_End
        ; Second Data_Area
        Data_Area_Beg
        Area_Name               = "Inputs"
```

```
Related_CFG_Identifier  = 1
IO_Direction            = 0  ;Input
Length                  = 7
Consistency             = 1
Data_Type               = 8  ;Floating Point 32
Data_Type               = 5  ;Unsigned8
Data_Type               = 5  ;Unsigned8
Data_Type               = 5  ;Unsigned8
Data_Area_End
; End Data_Area
EndModule
```

Channel_Diag: (O)
With the keyword Channel_Diag, the assignment of manufacturer-specific error types (Error_Type) in the channel-related diagnostic field to texts (Diag_Text) is specified.
Parameters Used:

Error_Type:
Type: Unsigned8 (16 <= Error_Type <= 31)

Diag_Text:
Type: Visible-String (32)

Channel_Diag_Help: (O starting with GSD_Revision 5)
Here additional information about channel-related diagnostic is defined. The configuration tool can offer this information to the user additional to the Diag_Text of the Channel_Diag corresponding error type.
Parameters used:

Error_Type:
Type: Unsigned8 (16 <= Error_Type <= 31)
Help_Text:
Type: Visible-String (256)

Fail_Safe: (D starting with GSD_Revision 1)
Here it is specified whether the DP slave accepts a data message without data instead of a data message with data = 0 in the CLEAR mode of the DP master (Class 1).
Type: Boolean (1: True)

Max_Diag_Data_Len: (M starting with GSD_Revision 1)
Here, the maximum length of the diagnostic information (Diag_Data) is specified.
Type: Unsigned8 (6 – 244)

Modul_Offset: (D starting with GSD_Revision 1)
Here, the slot number is specified that is to appear in the configuration tool as the first slot number at configuring (is used for improved representation).
Type: Unsigned8

Slave_Family: (M starting with GSD_Revision 1)
Here, the DP slave is assigned to a function class. The family name is structured hierarchically. In addition to the main family, subfamilies can be generated that are respectively added with "@". A maximum of three subfamilies can be defined.

Type: Unsigned8

The following main families are specified:

0: General (can't be assigned to the categories below)
1: Drives
2: Switching devices
3: I/O
4: Valves
5: Controllers
6: HMI (MMI)
7: Encoders
8: NC/RC
9: Gateway
10: Programmable Logic Controllers
11: Ident systems
12: PROFIBUS PA Profile (independent of used Physical Layer)
13 – 255: reserved

EXAMPLE 8 Slave_Family=3@Digital@24V

Diag_Update_Delay: (D starting with GSD_Revision 3)
The parameter is used to count the number of DDLM_Slave_Diag.con while Diag_Data.Prm_Req is still set (for slaves with reduced performance). The value of the Diag_Update_Delay is related to the Min_Slave_Intervall of the Slave.

Type: Unsigned8

Delay = Diag_Upd_Delay * Min_Slave_Intervall

Fail_Safe_required: (D starting with GSD_Revision 3)
This keyword corresponds to the keyword "Fail_Safe" of GSD_Revision 1.
The information is mapped to the Bit "Fail_Safe" in the DPV1_Status_1 of the DDLM_Set_Prm service.
The combination Fail_Safe = 0 and Fail_Safe_required =1 for the device or any module is not possible.

Type: Boolean (1: True)

True: The device or a module requires the Fail_Safe mode for secure operation and is not optional.
False: The use of the Fail_Safe mode is optional.

Info_Text: (O starting with GSD_Revision 3)
Here additional information about the device or the module can be described. The configuration tool can offer this information to the user additional to the visible string of the Model_Name or Module.

Type: Visible-String (256)

Max_User_Prm_Data_Len: (O starting with GSD_Revision 1; M starting with GSD_Revision 5)
Here, the maximum length of the user parameterization data is specified. The definition of this keyword excludes the evaluation of User_Prm_Data_Len and User_Prm_Data.
Type: Unsigned8 (0 – 237)

Ext_User_Prm_Data_Ref: (O starting with GSD_Revision 1)
Here, a reference to a user parameterization data description is specified. The definition of this keyword excludes the evaluation of User_Prm_Data and User_Prm_Data_Len. If areas overlap when describing the parameterization data, the area defined last in the GSD file has priority.
Parameters used:
Reference_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of the User_Prm_Data is defined.

Reference_Number:
Type: Unsigned16
Meaning: This reference number has to be the same as the reference number that is defined in the User_Prm_Data description.

Ext_User_Prm_Data_Const: (O starting with GSD_Revision 1)
Here, a constant part of the user parameterization data is specified. The definition of this keyword excludes the evaluation of User_Prm_Data and User_Prm_Data_Len. If areas overlap when describing the parameterization data, the area defined last in the GSD file has priority.
Parameters used:
Const_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of parameterization data is defined.

Const_Prm_Data:
Type: Octet-String
Meaning: Here, the constants or default selections within the parameterization data are defined.

ExtUserPrmData: (O starting with GSD_Revision 1)
Between the keywords ExtUserPrmData and EndExtUserPrmData, a parameter of the user parameterization data is described. The definition of this keyword excludes the evaluation of User_Prm_Data.
Parameters used:
Reference_Number:
Type: Unsigned16
Meaning: Here, the reference of the parameterization data description is specified. This reference has to be unique.

Ext_User_Prm_Data_Name:
Type: Visible-String (32) or "[SlotNumber]"
Meaning: Clear text description of the parameters. Here, the slot number can be entered automatically.
[SlotNumber]: (O starting with GSD_Revision 5)
If the Visible-String of the Ext_User_Prm_Data_Name is "[SlotNumber]", the real slot number will be entered automatically by the configuration tool.

EXAMPLE 9
ExtUserPrmData = 17 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData

Data_Type_Name:
Type: Visible-String (32)
Meaning: Default value of the described parameter.

Default_Value:
Type: DataType (has to correspond to the Data_Type_Name)
Meaning: Default value of the described parameter.

Min_Value:
Type: Data_Type (has to correspond to the Data_Type_Name)
Meaning: Minimum value of the described parameter.

Max_Value:
Type: Data_Type (has to correspond to the Data_Type_Name)
Meaning: Maximum value of the described parameter.

Allowed_Values:
Type: Data_Type_Array (16) (has to correspond to the Data_Type_Name)
Meaning: Permitted values of the described parameter.

Prm_Text_Ref:
Type: Unsigned16
Meaning: This reference number has to be the same as the reference number that is defined in the PrmText description.

Changeable: (O starting with GSD_Revision 4)
Type: Boolean (1: True, default = 1 if not present)
Meaning: Indicates whether this user parameter shall be changeable in the user dialog.

Visible: (O starting with GSD_Revision 4)
Type: Boolean (1: True, default = 1 if not present)
Meaning: Indicates whether this user parameter shall be visible in the user dialog.

PrmText:
Between the keywords PrmText and EndPrmText, possible values of a parameter are described. Texts are also assigned to these values.
Parameters Used:
Reference_Number:
Type: Unsigned16
Meaning: Here, the reference of the PrmText description is specified. This reference must be unique.

Text_Item:
Parameter Used:
Prm_Data_Value:
Type: Data_Type (has to correspond to the Data_Type_Name in the parameter description).
Meaning: Here, the value of the parameter is specified that is to be described.
Text:
Type: Visible-String (32)
Meaning: Description of the parameter value.

Prm_Block_Structure_supp: (O starting with GSD_Revision 4)
Here, the slave indicates that the block structure of the extended parameterization is supported within the user parameterization data.
If Prm_Block_Structure_supp = 1, the parameterization data shall be structured. The bit Prm_Structure (DPV1_Status_3) will be set by the configuration tool.
If Prm_Block_Structure_supp = 0, the parameterization data shall not be structured, but can show the form of a Block-Structure. The bit Prm_Structure will not be set by the configuration tool.
- The Prm_Structure is necessary for following blocks:
  PrmCmd(Structure_Type=2), DXB-Linktable(3), IsoM-Parameter(4), DXB-Subscribertable(7), Time AR Parameter(8), Manufacturer specific blocks(32 .. $128_{Decimal}$)
- Following blocks shall not be (pre-)defined within the GSD file:
  PrmCmd(Structure_Type=2), DXB-Linktable(3), IsoM-Parameter(4), DXB-Subscribertable(7), Time AR Parameter(8).
  For these blocks the configuration tool will insert the correspondig Prm_Block automatically to the parameterization telegram regarding to the keywords and the tool settings after the fixed blocks. The first fixed block contains the 3 DPV1-Status-Bytes.
- The F_Parameter-Block(5) is a fixed block and shall be described by the slave related keywords for PROFIsafe Profile.

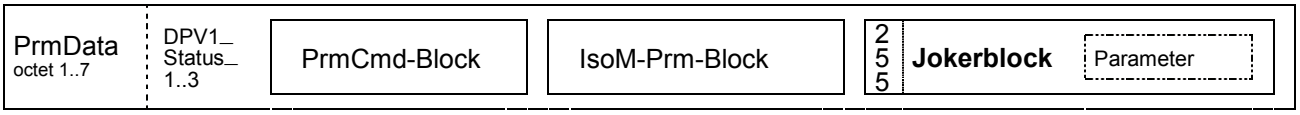
- The User_Prm_Data($129_{Decimal}$) and the Manufacturer specific blocks(32 .. $128_{Decimal}$) shall be described by the (X_)Ext_User_Prm_Dat_Ref or by the (X_)Ext_User_Prm_Dat_Const. These blocks shall be fixed defined within the GSD file. Fixed blocks will be inserted always at begin of the parameterization data.

Shall only be true, if DPV1_Slave = 1.
Type: Boolean (1: True)

Prm_Block_Structure_req: (O starting with GSD_Revision 4)
This parameter indicates whether the slave does require the master to support the Prm_Block_Structure.
Type: Boolean (1: True)
True: The device cannot be operated by a master that does not support the Prm_Block_Structure
False: The use of the Prm_Block_Structure is optional.

Jokerblock_supp: (O starting with GSD_Revision 5)
Indicates, if the DP-Slave supports a Jokerblock according to the block structure of the extended parameterization within the UserPrmData. Following rules have to be observed:
• the Jokerblock shall be used at the end of the parameterization telegram (after fix defined blocks as well as after blocks who will be inserted by the configuration tool);
• the parameter "length" of the Jokerblock is defined with "255";
• the Jokerblock shall not be used for PrmCmd, DXB-Linktable, IsoM_Parameter, DXB-Subscribertable, Time AR parameter, F_Parameter;
• the Jokerblock can be send to every slot;
• the Jokerblock can be used also at the X_Prm_SAP as last block of the extended parameterization telegram
Type: Boolean (1: True)

EXAMPLE 10 (Parameterization Telegram)



Jokerblock_Type: (M if Jokerblock_supp, starting with GSD_Revision 5)
Between the parameter Jokerblock_Type and End_Jokerblock_Type, each single block of the parameterization data of the Jokerblock will be described.
This parameter indicates what Structure_Type is described within the block.
Type: Unsigned8
0 .. 31: Reserved
32 .. 128: Manufacturer specific Data
129: User_Prm_Data
130 .. 255: Reserved

Jokerblock_Slot: (M if Jokerblock_supp, starting with GSD_Revision 5)
This parameter indicates the referenced Slot_Number.
Type: Unsigned8

Jokerblock_Location: (D starting with GSD_Revision 5)
This parameter indicates the location (SAP) where the Jokerblock should be inserted.
Type: Unsigned8
0: Prm-Telegram
1: Prm-Telegram or Ext-Prm-Telegram;
Only allowed, if X_Prm_SAP_supp = 1.
2: Ext-Prm-Telegram;
Only allowed, if X_Prm_SAP_supp = 1.
3 .. 255: Reserved

EXAMPLE 11

```
Jokerblock_supp          =  1
;
Jokerblock_Type          = 32
Jokerblock_Slot          =  5
Jokerblock_Location      =  1
End_Jokerblock_Type
;
Jokerblock_Type          = 33
Jokerblock_Slot          =  6
Jokerblock_Location      =  0
End_Jokerblock_Type
;
```

PrmCmd_supp: (O starting with GSD_Revision 5)
Indicates, if the DP-Slave supports PrmCmd.
Type: Boolean (1: True)

PrmCmd_req: (O starting with GSD_Revision 5)
Indicates, whether the slave does require the master to support PrmCmd.
Type: Boolean (1: True)

Slave_Max_Switch_Over_Time: (O starting with GSD_Revision 5)
Time needed within DP-Slave from PrmCmd receipt until the update of diagnosis with the calculated Red_State.
Type: Unsigned16
Time base: 10 ms

Slave_Redundancy_supp: (O starting with GSD_Revision 5)
Indicates, if the DP-Slave supports slave redundancy according [1].
Type: Unsigned8

0:	not supported
1:	Slave is not redundant but can be connected to a flying master.
2 .. 7:	Reserved
8:	Slave supports redundancy according [1].
9:	Slave supports redundancy according [1] or can be connected to a flying master. If connected to a flying master, the slave is used not redundant.
10 .. 255:	Reserved

If the value of "Slave_Redundancy_supp" is not equal to 0, the PrmCmd_supp keyword shall be set to "1" (true).

Ident_Maintenance_supp: (O starting with GSD_Revision 5)
The device or module supports I&M functions according [6].
Type: Boolean (1: True)

Time_Sync_supp: (O starting with GSD_Revision 5)
The device supports clock synchronization according to IEC 61784-1:2003, 7.2.3.2.5.10, that references to IEC 61158-5:2003, 8.2.9 Time ASE and from there to IEC 61158-3:2003, 14.4.5 and others.
Type: Boolean (1: True)

Max_iParameter_Size: (D starting with GSD_Revision 5)
Defines the maximum size of the i-Parameters that are required, in bytes.
Type: Unsigned32

4.4.2 Additional keywords for module assignment

**SlotDefinition: (O only if Modular_Station,
starting with GSD_Revision 3)**

Between the keywords SlotDefinition and EndSlotDefinition, the possibilities of using the modules within the slots is described.
The modules are referenced by the Module_Reference. The names of the slots are mandatory.
The default module will be integrated automatically in the configuration (-telegram). This module can be replaced with one of the permitted modules from the list.
The modules can be encountered using permitted values (8,9,13,...) or using a complete range (17-22).

Slot: (O starting with GSD_Revision 3)
Meaning: This parameter specifies the modules that can be used in the specified slot

Slot_Number:
Type: Unsigned8
Meaning: Here the number of the slot within the device is specified. The number of the slot must be starting with 1 and arise without gaps. If the SlotDefinition is used, then it's highly recommended, that the Modul_Offset is also equal 1. Not every slot of a device must be described by this slot definition. Additional modules may appear behind the highest defined Slot_Number.

Slot_Name:
Type: Visible-String (32)
Meaning: Text description of the slot (This means the application function name).

Default_Value:
Type: Unsigned16
Meaning: Default value, Module_Reference of the module used in this slot.

Min_Value:
Type: Unsigned16
Meaning: Minimum value, lowest Module_Reference of the modules that can be used in this slot.

Max_Value:
Type: Unsigned16
Meaning: Maximum value, highest Module_Reference of the modules that can be used in this slot.

Allowed_Values:
Type: Data_Type_Array (256) of Unsigned16
Meaning: Permitted values, list of Module_Reference of the modules that can be used in this slot.

4.4.3 Slave related keywords for DP extensions

PROFIBUS extensions mean the features of DP-V1 (see IEC 61784-1:2003 A3.1) and list of options (see IEC 61784-1:2003 A3.1 and 7.2.3.2.5), compared to DP-V0.

Table 1 illustrates the dependence of GSD keywords regarding the PROFIBUS DP extensions. Some of the keywords become only valid when other keywords (main selectors for DP-V1 protocol functions) are set TRUE. The right column of the table shows the resulting features and behavior of the device described by the GSD definitions of the left two columns.
In this GSD description the acyclic channel between master class1 and slave has the name MS1 and between master class2 and slave has the name MS2.

NOTE The corresponding names in the previous documents are MSAC_C1 and MSAC_C2.

A configuration tool for the DP extensions has to handle the defined first three byte of the user parameter data itself.
These bytes can also be defined by the known mechanism of the GSD (Ext_User_Prm_Dat_Ref,...), but the configuration tool for the DP extensions overwrites than GSD definitions. At last these bytes can be defined by the keywords for DP extensions, the configuration tool for the DP extensions overwrites the definitons from the user parameter and ext user parameter.

Table 1 — GSD keywords

Main Condition	Additional Condition	Conclusion
DPV1_Slave=0		Device is conform to PROFIBUS DP-V0, see IEC 61784-1:2003 A3.1 Device can not be operated with the following DP extensions (no acyclic services MS1, no data type support, no DP-V1 specific parameterization, no DP-V1 diagnosis model)
DPV1_Slave=0	C1_Read_Write_supp = 1 or DPV1_Data_Types = 1 or Check_Cfg_Mode = 1	invalid combination
DPV1_Slave=1		Device is conform to PROFIBUS DP-V1 extensions, see IEC 61784-1:2003 A3.1 Device supports DP-V1 specific parameterization and DP-V1 diagnosis model. This is an assumption for acyclic services MS1, Data_Types and Check_Cfg_Mode which are supported as stated by the corresponding keywords.

Main Condition	Additional Condition	Conclusion
DPV1_Slave=1 and C1_Read_Write _supp =0	C1_Max_Data_Len > 0 or C1_Response_Time out > 0 or C1_Read_Write _required = 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	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp =1		Device is conform to PROFIBUS DP-V1 extensions, see IEC 61784-1:2003 A3.1 and supports MS1 connection. This is an assumption for defining features of the MS1 connection and for Alarm support which are stated by the corresponding keywords.
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Diagnostic _Alarm_supp = 0	Diagnostic_Alarm _required = 1	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Process_Alarm_supp = 0	Process_Alarm _required = 1	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Pull_Plug_Alarm_supp = 0	Pull_Plug_Alarm _required = 1	Invalid combination

Main Condition	Additional Condition	Conclusion
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Status_Alarm_supp = 0	Status_Alarm _required = 1	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Status_Alarm_supp = 0	Status_Alarm_required = 1	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Update_Alarm_supp = 0	Update_Alarm_required = 1	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp = 1 and Manufacturer_Specific_Alarm_supp = 0	Manufacturer_Specific_Alarm_required = 1	Invalid combination
DPV1_Slave=1 and C1_Read_Write _supp =1 and 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		Device is conform to PROFIBUS DP extensions and supports MSAC_C1 connection and Alarms. This is an assumption for defining features of the Alarms which are stated by the corresponding keywords.

Main Condition	Additional Condition	Conclusion
C2_Read_Write _supp =0	C2_Max_Data_Len > 0 or C2_Response_Timeout > 0 or C2_Read_Write _required =1 or C2_Max_Count_Channels > 0 or Max_Initiate_PDU _Length > 0	Invalid combination
C2_Read_Write _supp =1		Device supports MS2 connection. The support of DP-V1 specific parametrization and DP-V1 diagnosis model is strongly recommended for migration of the whole DP extensions. Features of the MS2 connection are stated by the corresponding keywords.
WD_Base_1ms _supp		This works independent from the other PROFIBUS DP extensions. The assumption is that User_Prm_Data_Len > 0 are supported.

DPV1_Slave (D starting with GSD_Revision 3)
True, if the device uses DP-V1 functionality. This keyword is an extension to "Station_Type" and indicates if the slave operates as a standard DP- or DP-Slave with extended functionality.

The support of the several DP-V1 functionalities is defined in the following function specific keywords.

Type: Boolean (1: True)

C1_Read_Write_supp (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module with extended functionality is supporting the Read and Write services on the C1-communication relationship.

Type: Boolean (1: True)

C2_Read_Write_supp (D starting with GSD_Revision 3)
The DP-Slave with extended functionality is supporting the Read and Write services on the C2-communication relationship.

Type: Boolean (1: True)

C1_Max_Data_Len: (M if C1_read_write_supp, starting with GSD_Revision 3)
The parameter specifies the maximum length of user data excluding Function_Num, Slot_Number, Index and Length, transferred on the MSAC_1 communication channel.
Type: Unsigned8 (0 .. 240)

C2_Max_Data_Len: (M if C2_read_write_supp, starting with GSD_Revision 3)
The parameter specifies the maximum length of user data excluding Function_Num, Slot_number, Index, Length, transferred on the MSAC_2 communication channel.
Type: Unsigned8 (0,48 .. 240)

C1_Response_Timeout: (M if C1_read_write_supp, starting with GSD_Revision 3)
The parameter C1_Response_Timeout represents the efficiency of a DP-Slave with extended functionality. Each DP-Slave with extended functionality has to ensure that the parameter C1_Response_Timeout reaches the smallest value that is possible. By means of this parameter the DP-Slave with extended functionality indicates the maximum time to process an acyclic service (read, write, alarm_ack) on the C1-communication relationship.
Type: Unsigned16 (1 .. 65535)
Time base: 10 ms

C2_Response_Timeout: (M if C2_read_write_supp, starting with GSD_Revision 3)
The parameter C2_Response_Timeout represents the efficiency of a DP-Slave with extended functionality. Each DP-Slave with extended functionality has to ensure that the parameter C2_Response_Timeout reaches the smallest value that is possible. By means of this parameter the DP-Slave with extended functionality indicates the maximum time to process an acylic service (read, write, Data_Transport) on the C2-communication relationship.
Type: Unsigned16 (1 .. 65535)
Time base: 10 ms

C1_Read_Write_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires C1_Read_Write services to be accessed.
Type: Boolean (1: True)

C2_Read_Write_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires C2_Read_Write services to be accessed.
Type: Boolean (1: True)

C2_Max_Count_Channels: (M if C2_read_write_supp, starting with GSD_Revision 3)
The parameter defines the maximal amount of active C2 channels of the DP-V1 Slave.
Type: Unsigned8 (0 .. 49)

Max_Initiate_PDU_Length: (M if C2_read_write_supp, starting with GSD_Revision 3)
The parameter specifies the maximum length of an Initiate Request PDU including the Function_Num to the Resource Manager.
Type: Unsigned8 (0, 52 .. 244)

Diagnostic_Alarm_supp (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module supports Diagnostic_Alarm. A diagnostic alarm signals an event within a slot, for instance over temperature, short circuit, etc..
Type: Boolean (1: True)

Process_Alarm_supp (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module supports Process_Alarm. A process alarm signals the occurrence of an event in the connected process, for instance upper limit value exceeded.
Type: Boolean (1: True)

Pull_Plug_Alarm_supp (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module supports Pull_Plug_Alarm. A pull alarm signals the withdrawal of a module at a slot.
Type: Boolean (1: True)

Status_Alarm_supp (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module supports Status_Alarm. A status alarm signals a change in the state of a module, for instance run, stop or ready.
Type: Boolean (1: True)

Update_Alarm_supp: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module supports Update_Alarm. An update alarm signals the change of a parameter in a slot e.g. by a local operation or remote access.
Type: Boolean (1: True)

Manufacturer_Specific_Alarm_supp: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module supports Manufacturer_Specific_Alarm. A manufacturer specific alarm signals an event defined by the manufacturer.
Type: Boolean (1: True)

Extra_Alarm_SAP_supp (D starting with GSD_Revision 3)
Additional to SAP 51 it is possible to handle the MSAL_Alarm_Ack via SAP 50 if the Bit Sl_Flag.Extra_Alarm_SAP in the corresponding Slave Parameter Set is set. In this case there may be a higher performance because SAP 50 is used exclusively for the MSAL_Alarm_Ack service and the service cannot be delayed by a running MSAC1_Write or MSAC1_Read service.
Type: Boolean (1: True)

Alarm_Sequence_Mode_Count: (D starting with GSD_Revision 3)
The DP-Slave supports the Alarm_Sequence_Mode for alarm handling when this parameter is not 0. If this parameter is set to 0 only the Type Mode is supported by the slave.
The Sequence Mode is an option of the parallel alarm handling.
Several alarms (2 – 32) of the same or different type can be active (unacknowledged) at one time (fixed by the DDLM_Set_Prm service) at the DP-V1 Slave.
Type: Unsigned8 (0, 2 .. 32)

Alarm_Type_Mode_supp: (D starting with GSD_Revision 3;
M if the DP-Slave supports alarms, starting with GSD_Revision 4)
The DP-Slave supports the Type Mode for alarm handling.
The Type Mode is mandatory if the DP-Slave supports alarms.
Only one alarm of a specific Alarm_Type can be active at one time (fixed by the DDLM_Set_Prm service).
Type: Boolean (shall always be set to 1: True)

Diagnostic_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Process_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Pull_Plug_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Status_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Update_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Manufacturer_Specific_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

DPV1_Data_Types: (O starting with GSD_Revision 3)
The DP-Slave uses the vendor specific data of the extended identifier format for all modules with extended identifier format for coding of data types.
Type: Boolean (1: True)

WD_Base_1ms_supp: (D starting with GSD_Revision 3)
The DP-Slave supports the time base of 1 millisecond for the watchdog.
Type: Boolean (1: True)

Check_Cfg_Mode: (D starting with GSD_Revision 3)
With this parameter the slave indicates the possibility of a different user specific way to check the Cfg-Data.
This mode is switched on by the "Check_Cfg_Mode" in the DPV1_Status_2 of the Prm data.
Type: Boolean (1: True)

4.4.4 Slave related keywords for Data Exchange with Broadcast

Publisher_supp: (D starting with GSD_Revision 3)
The DP-Slave supports the Publisher functionality of Data Exchange with Broadcast.
Type: Boolean (1: True)

Subscriber_supp: (D starting with GSD_Revision 4)
The DP-Slave supports the Subscriber functionality of Data Exchange with Broadcast. If Subscriber_supp = 1, DPV1_Slave shall be 1.
Type: Boolean (1: True)

NOTE In order to secure the optimized performance of the publisher / subscriber functionality it is necessary to set the MaxTsdr_xx values (see 4.2.1) according to the actual values of the device.

DXB_Max_Link_Count: (O starting with GSD_Revision 4)
The maximum number of supported links to different publishers. Has to be unequal 0, if Subscriber_supp = 1.
Type: Unsigned8 (0 – 125)

DXB_Max_Data_Length: (O starting with GSD_Revision 4)
The maximum data length (in one piece) for a supported link to one publisher. Has to be unequal 0, if Subscriber_supp = 1.
Type: Unsigned8 (1 – 244)

DXB_Subscribertable_Block_Location: (D starting with GSD_Revision 5)
This parameter indicates what type of SAP is supported by the DXB-Subscribertable.
Type: Unsigned8

0: Prm-Telegram
1: Prm-Telegram or Ext-Prm-Telegram;
Only allowed, if X_Prm_SAP_supp = 1.
2: Ext-Prm-Telegram;
Only allowed, if X_Prm_SAP_supp = 1.
3: No Subscribertable to load
4 .. 255: Reserved

EXAMPLE

```
; Slave related keywords for DXB - Start
Publisher_supp              =  1
Subscriber_supp             =  1
DXB_Max_Link_Count    = 10
DXB_Max_Data_Length  = 32
DXB_Subscribertable_Block_Location =  1
; Slave related keywords for DXB - End
```

4.4.5 Slave related keywords for Isochronous Mode

Isochron_Mode_supp: (D starting with GSD_Revision 4)
This parameter indicates if the slave supports the Isochron_Mode. If the parameter is set to FALSE, all other isochronous parameters are not significant.
Type: Boolean (1: True)

Isochron_Mode_required: (D starting with GSD_Revision 4)
This parameter indicates whether the slave does require the master to support Isochron_Mode. If the parameter is set to TRUE, the slave cannot be operated by a master that does not support Isochron_Mode.
Type: Boolean (1: True)

TBASE_DP: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
Time base of T_{DP}, the DP cycle time, TDP_MIN and TDP_MAX, in units of $^{1}/_{12}$ µs. The smallest possible value shall be declared. This parameter shall not be present if Isochron_Mode_supp=0.
Type: Unsigned32, allowed values are 375, 750, 1500, 3000, 6000, 12000 which correspond to 31.25, 62.5, 125, 250, 500, 1000 µs respectively.

NOTE A configuration tool will calculate the possible values for T_{DP}. T_{DP} shall be the multiple value of TBASE_DP of all devices at the bus.

TDP_MIN: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
Minimum of T_{DP}, the DP cycle time, based on TBASE_DP. The values of this parameter for higher time bases of T_{DP} shall be calculated out of this value.
Type: Unsigned16, with range from 1 – 2^{16}-1

TDP_MAX: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
The maximum DP cycle time supported by the DP device in isochronous mode, based on TBASE_DP. The values of this parameter for higher time bases of T_{DP} shall be calculated out of this value. TDP_MAX should not exceed the range of 32 ms. The behaviour of values of TDP above 32 ms is not specified.
Type: Unsigned16, with range from 1 – 2^{16}-1

T_PLL_W_MAX: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
The maximum value of the jitter which is acceptable at the device input (RS485 receiver) based on $^1/_{12}$ µs.
Type: Unsigned16, with range from 12 – 2^{16}-1

TBASE_IO: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
Time base of T_I and T_O, where T_I is the point in time when the input values are collected and T_O is the point in time when the output values are taken over. The allowed values for the time base are equal to the definition for TBASE_DP (see above). The smallest possible value shall be declared. This parameter shall not be present if Isochron_Mode_supp=0.
Type: Unsigned32

TI_MIN: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
The minimum time based on TBASE_IO that is necessary to get and update the input values of an individual DP Slave. The values of this parameter for higher time bases of T_I and T_O shall be calculated out of this value.
Type: Unsigned16, with range from 0 (special case), 1 – 2^{16}-1

NOTE The values TI_MIN = TO_MIN = 0 shall cause the master to set the values T_I = T_O = 0. With the values T_I = T_O = 0 the buffered synchronized isochronous mode of a PROFIdrive slave is adjusted according to IEC 61158-5:2003, 8.2.2.4.3.2.

TO_MIN: (M if Isochron_Mode_supp, starting with GSD_Revision 4)
The minimum time based on TBASE_IO that is necessary at the end of the cyclic part of the Isochron DP cycle (T_{DX}) to get and output the output values given in units of TBASE_IO of an individual DP Slave. The values of this parameter for higher time bases of T_I and T_O shall be calculated out of this value.
Type: Unsigned16

EXAMPLE

```
; Slave related keywords for Isochronous Mode - Start
Isochron_Mode_supp     = 1
Isochron_Mode_required = 0
TBASE_DP               = 1500  ; equal to 125 µs
TDP_MAX                =  256  ; 256 * 125 µs =  32 ms
```

```
TDP_MIN              =  16  ;  16 * 125 µs =   2 ms
TBASE_IO             = 1500  ; equal to 125 µs
TI_MIN               =   1  ;   1 * 125 µs = 125 µs
TO_MIN               =   1  ;   1 * 125 µs = 125 µs
T_PLL_W_MAX          =  12  ; equal 12*1/12 µs = 1 µs
; Slave related keywords for Isochronous Mode – End
```

This example means, the device supports Isochron_Mode and can be run by either master whether it supports Isochron_Mode or not. Further, the time base for both, the DP cycle time and the T_I/T_O values is 1500 which corresponds to 125 µs. Therefore the minimal DP cycle time necessary for 3 Mbit/s is 16*125 µs which equals 2 ms, for 6 Mbit/s is 8*125 µs which equals 1 ms, the maximum cycle time supported by the device is 256*125 ms which equals 32 ms, the T_I and T_O can be calculated with 125 µs each (T_O 125 ms greater than T_{DX}), the maximum value of the jitter is $12*^1/_{12}$ µs which equals 1 µs.

4.4.6 Slave related keywords for PROFIsafe Profile

A DP-Slave device that implements a behavior according to the PROFIsafe profile shall specify its capabilities and the user parameters with the following set of keywords.

F parameters intended to be invisible shall be omitted. Omitted F parameters have a fixed value of **"0"** and are not included in calculation of CRC0 (F_ParamDescCRC).

The following F parameters shall be entered:

- of the bit parameter(*) set: F_SIL and F_Par_Version
- of the byte parameter set: all

That means, only F_Block_ID, F_CRC_Length, F_Check_iPar and F_Check_SeqNr may be omitted.

*) Bit parameters are those that consist of less than one byte, i.e. one to seven bits.

NOTE Further information to PROFIsafe is provided in [4].

F_ParamDescCRC (O starting with GSD_Revision 4)
In order to read the PROFIsafe parameter description safely from the GSD file, 2 byte of CRC code are necessary. The CRC code has to be calculated according to the PROFIsafe guidelines and certified by a registered authority (e.g. TUEV). The value of this parameter will not be transferred to the slave device but is needed to avoid errors during the parameterization with the configuration tool.
Type: Unsigned16

F_IO_StructureDescVersion (O starting with GSD_Revision 5)
Controls the layout of the PROFIsafe IO structure.
Currently, it affects the valid range of the F_IO_StructureDescCRC. A value of 1 indicates a 16-bit CRC while a value of 2 indicates a 32-bit CRC. If this attribute is not present, a value of 1 is assumed.
Type: Unsigned8

F_IO_StructureDescCRC (O starting with GSD_Revision 5)
In order to read the PROFIsafe IO structure description (config date) safely from the GSD file, a CRC code is necessary. The CRC code has to be calculated according to the PROFIsafe guidelines and certified by a registered authority (e.g. TUEV). The value of this parameter will not be transferred to the slave device but is needed to avoid errors during the configuration of the IO-Structure with the configuration tool.
The valid range is dependent on the presence and value of the keyword F_IO_StructureDescVersion. If missing or 1, the range is limited to 0..65535. If 2, the range is that of Unsigned32.
Type: Unsigned32

F_Ext_User_Prm_Data_Ref: (O starting with GSD_Revision 4)
Here, a reference to a User_Prm_Data description is specified. The definition of this keyword excludes the evaluation of User_Prm_Data. If areas overlap when describing the ExtUserPrmData, the area defined last in the device description block has priority.
Parameters used:

Reference_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of the ExtUserPrmData is defined.

Reference_Number:
Type: Unsigned16
Meaning: This reference number has to be the same as the reference number that is defined in the ExtUserPrmData description.

F_Ext_User_Prm_Data_Const: (O starting with GSD_Revision 4)
Here, a constant part of the ExtUserPrmData is specified. The definition of this keyword excludes the evaluation of User_Prm_Data. If areas overlap when describing the ExtUserPrmData, the area defined last in the GSD file has priority.
Parameters used:

Const_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of User_Prm_Data is defined.

Const_Prm_Data:
Type: Octet-String
Meaning: Here, the constants or default selections within the ExtUserPrmData are defined.

4.4.7 Slave related keywords for extended parameterization

X_Prm_SAP_supp: (D starting with GSD_Revision 4)
Indicates, if the X_Prm_SAP is supported by the slave. Shall only be true, if DPV1_Slave = 1.
Type: Boolean (1: True)

X_Max_User_Prm_Data_Len: (M if X_Prm_SAP_supp, starting with GSD_Revision 4)
Here, the maximum length of the ExtUserPrmData is specified. The use of this keyword is only allowed if DPV1_Slave = 1 and if X_Prm_SAP_supp = 1.
Type: Unsigned8 (5 – 244)

X_Ext_Module_Prm_Data_Len: (O starting with GSD_Revision 4)
Here, the length of the associated ExtUserPrmData is defined.
The use of this keyword is only allowed if DPV1_Slave = 1 and if X_Prm_SAP_supp = 1.
Type: Unsigned8 (1 – 244)

X_Ext_User_Prm_Data_Ref: (O starting with GSD_Revision 4)
Here, a reference to ExtUserPrmData description is specified. If areas overlap when describing the ExtUserPrmData, the area defined last in the GSD file has priority.
Parameters used:

Reference_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of the ExtUserPrmData is defined.

Reference_Number:
Type: Unsigned16
Meaning: This reference number has to be the same as the reference number that is defined in the ExtUserPrmData description.

X_Ext_User_Prm_Data_Const: (O starting with GSD_Revision 4)
Here, a constant part of the ExtUserPrmData is specified. If areas overlap when describing the ExtUserPrmData, the area defined last in the GSD file has priority.
Parameters used:

Const_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of ExtUserPrmData is defined.

Const_Prm_Data:
Type: Octet-String
Meaning: Here, the constants or default selections within the ExtUserPrmData are defined.

X_Prm_Block_Structure_supp: (M if X_Prm_SAP_supp, starting with GSD_Revision 4)
Here, the slave indicates that the block structure of the extended parameterization is supported when using the X_Prm_Service.
Shall only be true, if DPV1_Slave = 1.
Shall be true, if X_Prm_SAP_supp = 1.
Type: Boolean (1: True)

4.4.8 Slave related keywords for subsystems

A PROFIBUS DP slave device which has gateway capability towards an underlying communication system, also called *subsystem*, can provide a directory which holds DP indexes of the internal buffers representing the addressable Process Data objects. The user needs the information where to find this directory in order to get access to the data buffers representing the underlying communication system. The device manufacturer may provide one directory in slot 0 (this makes sense for a compact slave) or one directory in each slot for a modular slave.
Both keywords are optional, but only one keyword shall be used at the same time. This is because a modular slave could also use slot 0 for this directory, which is then valid for all type of modules. In that case no module specific definition is required.

Subsys_Dir_Index: (O starting with GSD_Revision 4)
The device has capabilities of a gateway towards a subsystem. The index of the subsystem object directory is given by this value. This definition has to appear within the unit definition. In order to decode the directory, the kind of the subsystem shall be specified in brackets.
Type of Index: Unsigned8
Type of Subsystem: Unsigned8, the values standing for:
1: Gateway capability according to [3]
0, 2 .. 127: Reserved
128 .. 255: User specific

EXAMPLE 1
Subsys_Dir_Index (1) = 15
means, the device is a gateway with a subsystem master device according to [3] where the subsystems master device object directory can be found in slot 0 at index 15.

Subsys_Module_Dir_Index: (O starting with GSD_Revision 4)
The device has capabilities of a gateway towards a subsystem. The index of the subsystem object directory is module specific and is given by this value. The slot corresponds to the module.
This definition has to appear within the module definition In order to decode the directory, the kind of the subsystem shall be specified.
Type of Index: Unsigned8
Type of Sybsystem: Unsigned8, the values standing for:
1: Gateway capability according to [3]
0, 2 .. 127: Reserved

128 .. 255: User specific

EXAMPLE 2

Subsys_Module_Dir_Index (1) = 42

means, the device is a gateway with a subsystem master device according to [3]. The subsystem master device object directory of the module where this definition appears can be found in the corresponding slot at index 42.

Annex A
(normative)

Formal description of GSD

Table A.2 specifies GSD in a formal way. All data in brackets are optional. The symbol "|" means the logical or-operation.
The number before every rule is a sequence number (S#) enabling the rules to be referenced.

Table A.2 — Formal Description of GSD format

S#	Formal description
259)	<Max_iParameter_Size> = [<WS>] Max_iParameter_Size [<WS>]=[<WS>]<Unsigned32><LineEnd>
258)	<F_IO_StructureDescVersion> = [<WS>] F_IO_StructureDescVersion [<WS>]=[<WS>]<Unsigned8><LineEnd>
257)	<F_IO_StructureDescCRC> = [<WS>] F_IO_StructureDescCRC [<WS>]=[<WS>]<Unsigned32><LineEnd>
256)	<Backslash> = \
255)	<Long-Line> = <Backslash><LineEnd>
254)	<WS> = <Space> | <Tab> | <Long-Line> | <WS><Space> | <WS><Tab> | <WS><Long-Line>
253)	<CRLF> = <Carriage Return><Line Feed> | <Carriage Return> | <Line Feed>
252)	<Num> = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
251)	<Namechar> = a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | _ | . | - | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | <Num>
250)	<Otherchar> = + | * | / | < | > | (|) | [|] | { | } | ! | $ | % | & | ? | ´ | ^ | | | = | # | ; | , | : | `
249)	<Baudrate> = 9.6 | 19.2 | 31.25 | 45.45 | 93.75 | 187.5 | 500 | 1.5M | 3M | 6M | 12M

248)	<Stringchar> = <Namechar> | <Otherchar>
247)	<Char> = <Stringchar> | "
246)	<Com> = ; | <Com><Char> | <Com><WS>
245)	<ComLn> = <Com><CRLF>
244)	<LineStart> = [<WS>] | [<WS>]<LineEnd><LineStart> { empty line }
243)	<LineEnd> = <CRLF> | <Com><CRLF> | <WS><LineEnd> | <LineEnd><ComLn> | <LineEnd><CRLF>
242)	<Boolean> = 0 | 1
241)	<Decimal> = <Num> | <Decimal><Num>
240)	<Hexchar> = <Num> | A | B | C | D | E | F | a | b | c | d | e | f
239)	<Hexadecimal> = 0x<Hexchar> | <Hexadecimal><Hexchar>
238)	<Number> = <Decimal> | <Hexadecimal>
237)	<Octet> = <Number> { 0 <= <Octet> <= 255 }
236)	<Unsigned8> = <Octet>
235)	<Unsigned16> = <Number> {0<=<Unsigned16><=65535}
234)	<Unsigned32> = <Number> { 0 <= <Unsigned32> <= 4294967295 }
233)	<Signed8> = [-] <Number> { -128 <= <Signed8> <= 127 }
232)	<Signed16> = [-] <Number> { -32768 <= <Signed16> <= 32767 }
231)	<Signed32> = [-] <Number> { -2147483648 <= <Signed32> <= 2147483647 }
230)	<Octet-String> = [<WS>]<Octet> | <Octet-String>[<WS>],[<WS>]<Octet>
229)	<String> = <Stringchar> | <Space> | <String><Stringchar> | <String><Space>
228)	<Visible-String> = "<String>"
227)	<Keyword> = <Namechar> | <Keyword><Namechar>
226)	<Any-String> = <Char> | <WS> | <Any-String><Char> | <Any-String><WS>

225)	<Any-Line> = <CRLF> | <Any-String><CRLF>
224)	<Any-Text> = <Any-Line> | <Any-Text><Any-Line>
223)	<User-Definition> = <Keyword>[<WS>] [<Otherchar><Any-Line>]
222)	<GSD_Revision> = <Unsigned8>
221)	<Vendor_Name> = <Visible-String> { Length <= 32 }
220)	<Model_Name> = <Visible-String> { Length <= 32 }
219)	<Revision> = <Visible-String> { Length <= 32 }
218)	<Revision_Number> = <Unsigned8>
217)	<Ident_Number> = <Unsigned16>
216)	<Protocol_Ident> = <Unsigned8>
215)	<Station_Type> = <Unsigned8>
214)	<FMS_supp> = <Boolean>
213)	<Hardware_Release> = <Visible-String> { Length <= 32 }
212)	<Software_Release> = <Visible-String> { Length <= 32 }
211)	<Baudrate_supp> = <Boolean>
210)	<MaxTsdr> = <Unsigned16>
209)	<Redundancy> = <Boolean>
208)	<Repeater_Ctrl_Sig> = <Unsigned8>
207)	<24V_Pins> = <Unsigned8>
206)	<Implementation_Type> = <Visible-String> { Length <= 32}
205)	<Bitmap_Device> = <Visible-String> { Length <= 8}
204)	<Bitmap_Diag> = <Visible-String> { Length <= 8}
203)	<Bitmap_SF> = <Visible-String> { Length <= 8}
202)	<Transmission_Delay> = <Unsigned16>
201)	<Reaction_Delay> = <Unsigned16>
200)	<Master_Freeze_Mode_supp> = <Boolean>
199)	<Master_Sync_Mode_supp> = <Boolean>
198)	<Master_Fail_Safe_supp> = <Boolean>
197)	<Download_supp> = <Boolean>
196)	<Upload_supp> = <Boolean>
195)	<Act_Para_Brct_supp> = <Boolean>
194)	<Act_Param_supp> = <Boolean>
193)	<Max_MPS_Length> = <Unsigned32>
192)	<Max_Lsdu_MM> = <Unsigned8>
191)	<Max_Lsdu_MS> = <Unsigned8>
190)	<Min_Poll_Timeout> = <Unsigned16>
189)	<Trdy> = <Unsigned8>
188)	<Tqui> = <Unsigned8>

187)	<Tset> = <Unsigned8>
186)	<Tsdi> = <Unsigned16>
185)	<LAS_Len> = <Unsigned8>
184)	<Max_Slaves_supp> = <Unsigned8>
183)	<Max_Master_Input_Len> = <Unsigned8>
182)	<Max_Master_Output_Len> = <Unsigned8>
181)	<Max_Master_Data_Len> = <Unsigned16>
180)	<Isochron_Mode_Synchronised> = <Unsigned8>
179)	<DXB_Master_supp> = <Boolean>
178)	<X_Master_Prm_SAP_supp> = <Boolean>
177)	<DPV1_Master> = <Boolean>
176)	<DPV1_Conformance_Class> = <Unsigned8>
175)	<C1_Master_Read_Write_supp> = <Boolean>
174)	<Master_DPV1_Alarm_supp> = <Boolean>
173)	<Master_Diagnostic_Alarm_supp> = <Boolean>
172)	<Master_Process_Alarm_supp> = <Boolean>
171)	<Master_Pull__Plug_Alarm_supp> = <Boolean>
170)	<Master_Status_Alarm_supp> = <Boolean>
169)	<Master_Update_Alarm_supp> = <Boolean>
168)	<Master_Manufacturer_Specific_Alarm_supp> = <Boolean>
167)	<Master_Extra_Alarm_SAP_supp> = <Boolean>
166)	<Master_Alarm_Sequence_Mode> = <Unsigned8>
165)	<Master_Alarm_Type_Mode_supp> = <Boolean>
164)	<Freeze_Mode_supp> = <Boolean>
163)	<Sync_Mode_supp> = <Boolean>
162)	<Set_Slave_Add_supp> = <Boolean>
161)	<Auto_Baud_supp> = <Boolean>
160)	<User_Prm_Data_Len> = <Unsigned8>
159)	<User_Prm_Data> = <Octet-String>
158)	<Min_Slave_Intervall> = <Unsigned16>
157)	<Modular_Station> = <Boolean>
156)	<Max_Module> = <Unsigned8>
155)	<Max_Input_Len> = <Unsigned8>
154)	<Max_Output_Len> = <Unsigned8>
153)	<Max_Data_Len> = <Unsigned16>
152)	<Modul_Offset> = <Unsigned8>
151)	<Bit> = <Unsigned16>
150)	<Diag_Text> = <Visible-String> { Length <= 32}
149)	<Help_Text> = <Visible-String> { Length <= 256}
148)	<First_Bit> = <Bit>
147)	<Last_Bit> = <Bit>

146)	<Value> = <Unsigned16>
145)	<Mod_Name> = <Visible-String> { Length <= 32}
144)	<Config> = <Octet-String>
143)	<Error_Type> = <Unsigned8> { 16 <= <Error_Type> <= 31 }
142)	<Subfamily_Name> = <String> { Length <= 32}
141)	<Family_Name> = <Unsigned8> | <Unsigned8>@<Subfamily_Name> | <Unsigned8>@<Subfamily_Name> @<Subfamily_Name> | <Unsigned8>@<Subfamily_Name> @<Subfamily_Name>@<Subfamily_Name>
140)	<Info_Text> = Info_Text[<WS>]=[<WS>]<Visible-String>{Length<=256}
139)	<Prm_Block_Structure_req> = <Boolean>
138)	<Prm_Block_Structure_supp> = <Boolean>
137)	<Jokerblock_supp> = <Boolean>
136)	<Jokerblock_Type> = <Unsigned8>
135)	<Jokerblock_Slot> = <Unsigned8>
134)	<Jokerblock_Location> = <Unsigned8>
133)	<Jokerblock-Item> = Jokerblock_Slot[<WS>]=[<WS>]<Jokerblock_Slot> | Jokerblock_Location[<WS>]= [<WS>]<Jokerblock_Location>
132)	<Jokerblock-List> = <Jokerblock-Item> | <Jokerblock-List><Jokerblock-Item>
131)	<Jokerblock-Def> = Jokerblock_Type[<WS>]= [<WS>]<Jokerblock_Type><LineEnd> <Jokerblock-List> End_Jokerblock_Type
130)	<Fail_Safe> = <Boolean>
129)	<Fail_Safe_required> = <Boolean>
128)	<Max_Diag_Data_Len> = <Unsigned8>
127)	<Diag_Update_Delay> = <Unsigned8>
126)	<PrmCmd_supp> = <Boolean>
125)	<Slave_Max_Switch_Over_Time> = <Unsigned16>
124)	<Slave_Redundancy_supp> = <Unsigned8>
123)	<Ident_Maintenance_supp> = <Boolean>
122)	<Time_Sync_supp> = <Boolean>
121)	<DPV1_Slave> = <Boolean>

120)	<C1_Read_Write_supp> = <Boolean>
119)	<C2_Read_Write_supp> = <Boolean>
118)	<C1_Max_Data_Len> = <Unsigned8>
117)	<C2_Max_Data_Len> = <Unsigned8>
116)	<C1_Response_Timeout> = <Unsigned16>
115)	<C2_Response_Timeout> = <Unsigned16>
114)	<C1_Read_Write_required> = <Boolean>
113)	<C2_Read_Write_required> = <Boolean>
112)	<C2_Max_Count_Channels> = <Unsigned8>
111)	<Max_Initiate_PDU_Length> = <Unsigned8>
110)	<Diagnostic_Alarm_supp> = <Boolean>
109)	<Process_Alarm_supp> = <Boolean>
108)	<Pull_Plug_Alarm_supp> = <Boolean>
107)	<Status_Alarm_supp> = <Boolean>
106)	<Update_Alarm_supp> = <Boolean>
105)	<Manufacturer_Specific_Alarm_supp> = <Boolean>
104)	<Extra_Alarm_SAP_supp> = <Boolean>
103)	<Alarm_Sequence_Mode_Count> = <Unsigned8>
102)	<Alarm_Type_Mode_supp> = <Boolean>
101)	<Alarm_required> = <Boolean>
100)	<DPV1_Data_Types> = <Boolean>
99)	<WD_Base_1ms_supp> = <Boolean>
98)	<Check_Cfg_Mode> = <Boolean>
97)	<Max_User_Prm_Data_Len> = <Unsigned8>
96)	<Reference_Number> = <Unsigned16>
95)	<Reference_Offset> = <Unsigned8>
94)	<Const_Offset> = <Unsigned8>
93)	<Const_Prm_Data> = <Octet-String>
92)	<Module_Reference> = <Unsigned16>
91)	<Mod-Ref-String> = [<WS>]<Module_Reference> | <Mod-Ref-String>[<WS>],[<WS>]<Module_Reference>
90)	<Slot_Number> = <Unsigned8>
89)	<Slot_Name> = <Visible-String> { Length <= 32}
88)	<Bit-Area> = BITAREA(<First_Bit>- <Last_Bit>){0<=First_Bit<=Last_Bit<=7} {Value Range: UNSIGNED(Last_Bit-First_Bit+1)}
87)	<Data_Type_Name> = UNSIGNED8 | UNSIGNED16 | UNSIGNED32 | SIGNED8 | SIGNED16 | SIGNED32 | BIT(<Bit>) | <Bit-Area> {0<=Bit<=7}

86)	<Data_Type> = <Unsigned8> | <Unsigned16> | <Unsigned32> | <Signed8> | <Signed16> | <Signed32> | <Bit>
85)	<Data_Type_Array> = [<WS>]<Data_Type> | <Data_Type_Array>[<WS>],[<WS>]<Data_Type>
84)	<Default_Value> = <Data_Type>
83)	<Min_Value> = <Data_Type>
82)	<Max_Value> = <Data_Type>
81)	<Allowed_Values> = <Data_Type_Array>
80)	<Prm_Data_Value> = <Data_Type>
79)	<Prm_Text_Ref> = Prm_Text_Ref[<WS>]= [<WS>]<Reference_Number><LineEnd>
78)	<Ext_User_Prm_Data_Name> = <Visible-String> { Length <= 32}
77)	<Text> = <Visible-String> { Length <= 32}
76)	–
75)	–
74)	<X_Value_Item> = X_Value[<WS>](<Value>)[<WS>]= [<WS>]<Diag_Text><LineEnd> | X_Value_Help[<WS>](<Value>)[<WS>]= [<WS>]<Help_Text><LineEnd>
73)	<Value_Item> = Value[<WS>](<Value>)[<WS>]= [<WS>]<Diag_Text><LineEnd> | Value_Help[<WS>](<Value>)[<WS>]= [<WS>]<Help_Text><LineEnd>
72)	<X_Value_List> = <X_Value_Item> | <X_Value-List><X_Value-Item>
71)	<Value_List> = <Value_Item> | <Value-List><Value-Item>
70)	<X-Unit-Diag-Area-Def> = X_Unit_Diag_Area[<WS>]= [<WS>]<First_Bit>-<Last_Bit><LineEnd><X_Value_List> X_Unit_Diag_Area_End {24<=First_Bit<=Last_Bit<=495}
69)	<Unit-Diag-Area-Def> = Unit_Diag_Area[<WS>]= [<WS>]<First_Bit>-<Last_Bit><LineEnd><Value_List> Unit_Diag_Area_End {0<=First_Bit <= Last_Bit<=495}
68)	<X-Unit-Diag-Def> = X_Unit_Diag_Bit[<WS>](<Bit>)[<WS>]= [<WS>]<Diag_Text> {24<=Bit<=495} | X_Unit_Diag_Not_Bit[<WS>](<Bit>)[<WS>]=

	[<WS>]<Diag_Text> {24<=Bit<=495} | X_Unit_Diag_Bit_Help[<WS>](<Bit>)[<WS>]= [<WS>]<Help_Text> {24<=Bit<=495} | X_Unit_Diag_Not_Bit_Help[<WS>](<Bit>)[<WS>]= [<WS>]<Help_Text> {24<=Bit<=495} | <X-Unit-Diag-Area-Def>
67)	<Diag_Type_Number> = <Unsigned8>
66)	<Unit-Diag-List> = <X-Unit-Diag-Def> [|<Unit-Diag-List><X-Unit-Diag-Def>]<LineEnd>
65)	<Unit-Diag-Type-Def> = UnitDiagType[<WS>]= [<WS>]<Diag_Type_Number><LineEnd> <Unit-Diag-List> EndUnitDiagType
64)	<Channel-Diag-Definition>= Channel_Diag[<WS>](<Error_Type>)[<WS>]= [<WS>]<Diag_Text> | Channel_Diag_Help[<WS>](<Error_Type>)[<WS>]= [<WS>]<Help_Text><LineEnd>
63)	<Ph_Delay_Item> = Transmission_Delay_9.6[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_19.2[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_31.25[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_45.45[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_93.75[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_187.5[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_500[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_1.5M[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_3M[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_6M[<WS>]= [<WS>]<Transmission_Delay> | Transmission_Delay_12M[<WS>]= [<WS>]<Transmission_Delay> | Reaction_Delay_9.6[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_19.2[<WS>]=[<WS>]<Reaction_Delay>

	| Reaction_Delay_31.25[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_45.45[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_93.75[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_187.5[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_500[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_1.5M[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_3M[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_6M[<WS>]=[<WS>]<Reaction_Delay> | Reaction_Delay_12M[<WS>]=[<WS>]<Reaction_Delay> <LineEnd>
62)	<Ph-Delay-List> = <Ph_Delay_Item> | <Ph-Delay-List><Ph_Delay_Item>
61)	<Ph-Interface-Def> = Physical_Interface[<WS>]=[<WS>]<Unsigned8><LineEnd> <Ph-Delay-List> End_Physical_Interface
60)	<Slot_Item> = Slot(<Slot_Number>)[<WS>]=[<WS>]<Slot_Name> <WS><Module_Reference> [<WS><Module_Reference>[<WS>]- [<WS>]<Module_Reference> | <WS><Mod-Ref-String>] <LineEnd>
59)	<Slot-List> = <Slot_Item> | <Slot-List><Slot_Item>
58)	<Slot-Def> = SlotDefinition<LineEnd> <Slot-List> EndSlotDefinition
57)	<Data-Type-Item> = Data-Type[<WS>]=[<WS>]<Unsigned8>
56)	<Data-Type-List> = <Data-Type-Item> | <Data-Type-List><Data-Type-Item>
55)	<Data-Area-Item> = Area_Name[<WS>]= [<WS>]<Visible-String> { Length <= 32 }<LineEnd> Related_CFG_Identifier[<WS>]=[<WS>]<Unsigned8> <LineEnd> IO_Direction<WS>=[<WS>]<Boolean><LineEnd> Length[<WS>]=[<WS>]<Unsigned8><LineEnd> Consistency<WS>=[<WS>]<Unsigned8><LineEnd> Publisher_allowed[<WS>]=[<WS>]<Boolean><LineEnd> DP_Master_allowed[<WS>]=[<WS>]<Boolean><LineEnd>

	<Data-Type-List>
54)	<Data-Area-List> = <Data-Area-Item> | <Data-Area-List><Data-Area-Item>
53)	<Data-Area-Def> = Data_Area_Beg<LineEnd> <Data-Area-List> Data_Area_End
52)	<Alarm_Support> = Diagnostic_Alarm_supp[<WS>] = [<WS>]<Alarm_supp> | Process_Alarm_supp[<WS>] = [<WS>]<Alarm_supp> | Pull_Plug_Alarm_supp[<WS>] = [<WS>]<Alarm_supp> | Status_Alarm_supp[<WS>] = [<WS>]<Alarm_supp> | Update_Alarm_supp[<WS>] = [<WS>]<Alarm_supp> | Manufacturer_Specific_Alarm_supp[<WS>]= [<WS>]<Alarm_supp>
51)	<Alarm_Requirement> = Diagnostic_Alarm_required[<WS>] = [<WS>]<Alarm_required> | Process_Alarm_required[<WS>] = [<WS>]<Alarm_required> | Pull_Plug_Alarm_required[<WS>] = [<WS>]<Alarm_required> | Status_Alarm_required[<WS>]= [<WS>]<Alarm_required> | Update_Alarm_required[<WS>] = [<WS>]<Alarm_required> | Manufacturer_Specific_Alarm_required[<WS>]= [<WS>]<Alarm_required>
50)	<DXB_Subscribertable_Block_Location> = DXB_Subscribertable_Block_Location [<WS>]= [<WS>]<Unsigned8>
49)	<DXB_Max_Data_Length> = DXB_Max_Data_Length[<WS>]=[<WS>]<Unsigned8>
48)	<DXB_Max_Link_Count> = DXB_Max_Link_Count[<WS>]=[<WS>]<Unsigned8>
47)	<Subscriber_supp> = Subscriber_supp[<WS>]=[<WS>]<Boolean>
46)	<Publisher_supp> = Publisher_supp[<WS>]=[<WS>]<Boolean>
45)	<DXB-List> = [<WS>]<Publisher_supp> | <Subscriber_supp> | <DXB_Max_Link_Count> | <DXB_Max_Data_Length>

	| <DXB_Subscribertable_Block_Location>
44)	<X_Prm_Block_Structure_supp> = [<WS>] X_Prm_Block_Structure_supp[<WS>]= [<WS>]<Boolean><LineEnd>
43)	<X_Ext_User_Prm_Data_Const> = [<WS>] X_Ext_User_Prm_Data_Const(<Const_Offset>)[<WS>]= [<WS>]<Const_Prm_Data><LineEnd>
42)	<X_Ext_User_Prm_Data_Ref> = [<WS>] X_Ext_User_Prm_Data_Ref(<Reference_Offset>)[<WS>]= [<WS>]<Reference_Number><LineEnd>
41)	<X_Max_User_Prm_Data_Len> = [<WS>] X_Max_User_Prm_Data_Len[<WS>]= [<WS>]<Unsigned8><LineEnd>
40)	<X_Prm_SAP_supp> = [<WS>] X_Prm_SAP_supp[<WS>]= [<WS>]<Boolean><LineEnd>
39)	<X-Prm-List> = [<WS>]<X_Prm_SAP_supp> | <X_Max_User_Prm_Data_Len> | <X_Ext_User_Prm_Data_Ref> | <X_Ext_User_Prm_Data_Const> | <X_Prm_Block_Structure_supp>
38)	<Isochron-Mode-supp> = [<WS>] Isochron_Mode_supp[<WS>]= [<WS>]<Boolean><LineEnd>
37)	<Isochron-Mode-required> = [<WS>] Isochron_Mode_required[<WS>]= [<WS>]<Boolean><LineEnd>
36)	<TBASE-DP> = [<WS>] TBASE_DP[<WS>]=[<WS>]<Unsigned32><LineEnd>
35)	<TDP-MIN> = [<WS>] TDP_MIN[<WS>]=[<WS>]<Unsigned16><LineEnd>
34)	<TDP-MAX> = [<WS>] TDP_MAX[<WS>]=[<WS>]<Unsigned16><LineEnd>
33)	<T_PLL_W_MAX> = [<WS>] T_PLL_W_MAX[<WS>]=[<WS>]<Unsigned16><LineEnd>
32)	<TBASE-IO> = [<WS>] TBASE_IO[<WS>]=[<WS>]<Unsigned32><LineEnd>
31)	<TI-MIN> = [<WS>] TI_MIN[<WS>]=[<WS>]<Unsigned16><LineEnd>
30)	<TO-MIN> = [<WS>] TO_MIN[<WS>]=[<WS>]<Unsigned16><LineEnd>
29)	<Isochron-Mode-List> = [<WS>]<Isochron-Mode-supp> | <Isochron-Mode-required> | <T_PLL_W_MAX>

	| <TBASE-DP-List> | <TDP-MIN> | <TDP-MAX> | <TBASE-IO-List> | <TI-MIN> | <TO-MIN>
28)	<Visible> = [<WS>] Visible[<WS>]=[<WS>]<Boolean><LineEnd>
27)	<Changeable> = [<WS>] Changeable[<WS>]=[<WS>]<Boolean><LineEnd>
26)	<F_Ext_User_Prm_Data_Const> = [<WS>] F_Ext_User_Prm_Data_Const(<Const_Offset>)[<WS>]= [<WS>]<Const_Prm_Data><LineEnd>
25)	<F_Ext_User_Prm_Data_Ref> = [<WS>] F_Ext_User_Prm_Data_Ref(<Reference_Offset>)[<WS>]= [<WS>]<Reference_Number><LineEnd>
24)	<F_ParamDescCRC> = [<WS>] F_ParamDescCRC[<WS>]=[<WS>]<Unsigned16><LineEnd>
23)	<F-Param-List> = [<WS>]<F_ParamDescCRC> | <F_IO_StructureDescVersion> | <F_IO_StructureDescCRC> | <F_Ext_User_Prm_Data_Ref> | <F_Ext_User_Prm_Data_Const>
22)	<Subsys-Type>=<Unsigned8>
21)	<Subsys-Dir-Index>=<Unsigned8>
20)	<Subsys-Module-Dir-Index-Def> = Subsys_Module_Dir_Index[<WS>] (<Subsys-Type>)[<WS>]=[<WS>]<Subsys-Dir-Index>
19)	<Subsys-Dir-Index-Def> = Subsys_Dir_Index[<WS>] (<Subsys- Type>)[<WS>]=[<WS>]<Subsys-Dir-Index>
18)	<Ext-User-Prm-Data-Const> = Ext_User_Prm_Data_Const(<Const_Offset>)[<WS>]= [<WS>]<Const_Prm_Data>
17)	<Ext-User-Prm-Data-Ref> = Ext_User_Prm_Data_Ref(<Reference_Offset>)[<WS>]= [<WS>]<Reference_Number>
16)	<Unit-Def-Item> = GSD_Revision[<WS>]=[<WS>]<GSD_Revision> | Vendor_Name[<WS>]=[<WS>]<Vendor_Name> | Model_Name[<WS>]=[<WS>]<Model_Name> | Revision[<WS>]=[<WS>]<Revision> | Revision_Number[<WS>]=[<WS>]<Revision_Number> | Ident_Number[<WS>]=[<WS>]<Ident_Number> | Protocol_Ident[<WS>]=[<WS>]<Protocol_Ident> | Station_Type[<WS>]=[<WS>]<Station_Type> | FMS_supp[<WS>]=[<WS>]<FMS_supp> | Hardware_Release[<WS>]=[<WS>]<Hardware-Release>

	| Software_Release[<WS>]=[<WS>]<Software-Release> | <Info_Text> | 9.6_supp[<WS>]=[<WS>]<Baudrate_supp> | 19.2_supp[<WS>]=[<WS>]<Baudrate_supp> | 31.25_supp[<WS>]=[<WS>]<Baudrate_supp> | 45.45_supp[<WS>]=[<WS>]<Baudrate_supp> | 93.75_supp[<WS>]=[<WS>]<Baudrate_supp> | 187.5_supp[<WS>]=[<WS>]<Baudrate_supp> | 500_supp[<WS>]=[<WS>]<Baudrate_supp> | 1.5M_supp[<WS>]=[<WS>]<Baudrate_supp> | 3M_supp[<WS>]=[<WS>]<Baudrate_supp> | 6M_supp[<WS>]=[<WS>]<Baudrate_supp> | 12M_supp[<WS>]=[<WS>]<Baudrate_supp> | MaxTsdr_9.6[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_19.2[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_31.25[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_45.45[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_93.75[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_187.5[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_500[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_1.5M[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_3M[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_6M[<WS>]=[<WS>]<MaxTsdr> | MaxTsdr_12M[<WS>]=[<WS>]<MaxTsdr> | Redundancy[<WS>]=[<WS>]<Redundancy> | Repeater_Ctrl_Sig[<WS>]=[<WS>]<Repeater_Ctrl_Sig> | 24V_Pins[<WS>]=[<WS>]<24V_Pins> | Implementation_Type[<WS>]= [<WS>]<Implementation_Type> | Bitmap_Device[<WS>]=[<WS>]<Bitmap_Device> | Bitmap_Diag[<WS>]=[<WS>]<Bitmap_Diag> | Bitmap_SF[<WS>]=[<WS>]<Bitmap_SF> | Master_Freeze_Mode_supp[<WS>]= [<WS>]<Master_Freeze_Mode_supp> | Master_Sync_Mode_supp[<WS>]= [<WS>]<Master_Sync_Mode_supp> | Master_Fail_Safe_supp[<WS>]= [<WS>]<Master_Fail_Safe_supp> | Download_supp[<WS>]=[<WS>]<Download_supp> | Upload_supp[<WS>]=[<WS>]<Upload_supp> | Act_Para_Brct_supp[<WS>]= [<WS>]<Act_Para_Brct_supp>

	| Act_Param_supp[<WS>]= [<WS>]<Act_Param_supp> | Max_MPS_Length[<WS>]=[<WS>]<Max_MPS_Length> | Max_Lsdu_MM[<WS>]=[<WS>]<Max_Lsdu_MM> | Max_Lsdu_MS[<WS>]=[<WS>]<Max_Lsdu_MS> | Min_Poll_Timeout[<WS>]=[<WS>]<Min_Poll_Timeout> | Trdy_9.6[<WS>]=[<WS>]<Trdy> | Trdy_19.2[<WS>]=[<WS>]<Trdy> | Trdy_31.25[<WS>]=[<WS>]<Trdy> | Trdy_45.45[<WS>]=[<WS>]<Trdy> | Trdy_93.75[<WS>]=[<WS>]<Trdy> | Trdy_187.5[<WS>]=[<WS>]<Trdy> | Trdy_500[<WS>]=[<WS>]<Trdy> | Trdy_1.5M[<WS>]=[<WS>]<Trdy> | Trdy_3M[<WS>]=[<WS>]<Trdy> | Trdy_6M[<WS>]=[<WS>]<Trdy> | Trdy_12M[<WS>]=[<WS>]<Trdy> | Tqui_9.6[<WS>]=[<WS>]<Tqui> | Tqui_19.2[<WS>]=[<WS>]<Tqui> | Tqui_31.25[<WS>]=[<WS>]<Tqui> | Tqui_45.45[<WS>]=[<WS>]<Tqui> | Tqui_93.75[<WS>]=[<WS>]<Tqui> | Tqui_187.5[<WS>]=[<WS>]<Tqui> | Tqui_500[<WS>]=[<WS>]<Tqui> | Tqui_1.5M[<WS>]=[<WS>]<Tqui> | Tqui_3M[<WS>]=[<WS>]<Tqui> | Tqui_6M[<WS>]=[<WS>]<Tqui> | Tqui_12M[<WS>]=[<WS>]<Tqui> | Tset_9.6[<WS>]=[<WS>]<Tset> | Tset_19.2[<WS>]=[<WS>]<Tset> | Tset_31.25[<WS>]=[<WS>]<Tset> | Tset_45.45[<WS>]=[<WS>]<Tset> | Tset_93.75[<WS>]=[<WS>]<Tset> | Tset_187.5[<WS>]=[<WS>]<Tset> | Tset_500[<WS>]=[<WS>]<Tset> | Tset_1.5M[<WS>]=[<WS>]<Tset> | Tset_3M[<WS>]=[<WS>]<Tset> | Tset_6M[<WS>]=[<WS>]<Tset> | Tset_12M[<WS>]=[<WS>]<Tset> | Tsdi_9.6[<WS>]=[<WS>]<Tsdi> | Tsdi_19.2[<WS>]=[<WS>]<Tsdi> | Tsdi_31.25[<WS>]=[<WS>]<Tsdi>

	| Tsdi_45.45[<WS>]=[<WS>]<Tsdi> | Tsdi_93.75[<WS>]=[<WS>]<Tsdi> | Tsdi_187.5[<WS>]=[<WS>]<Tsdi> | Tsdi_500[<WS>]=[<WS>]<Tsdi> | Tsdi_1.5M[<WS>]=[<WS>]<Tsdi> | Tsdi_3M[<WS>]=[<WS>]<Tsdi> | Tsdi_6M[<WS>]=[<WS>]<Tsdi> | Tsdi_12M[<WS>]=[<WS>]<Tsdi> | LAS_Len[<WS>]=[<WS>]<LAS_Len> | Max_Slaves_supp[<WS>]=[<WS>]<Max_Slaves_supp> | Max_Master_Input_Len[<WS>]= [<WS>]<Max_Master_Input_Len> | Max_Master_Output_Len[<WS>]= [<WS>]<Max_Master_Output_Len> | Max_Master_Data_Len[<WS>]= [<WS>]<Max_Master_Data_Len> | DPV1_Master[<WS>]=[<WS>]<DPV1_Master> | DPV1_Conformance_Class[<WS>]= [<WS>]<DPV1_Conformance_Class> | C1_Master_Read_Write_supp[<WS>]= [<WS>]< C1_Master_Read_Write_supp> | Master_DPV1_Alarm_supp[<WS>]= [<WS>]<Master_DPV1_Alarm_supp> | Master_Diagnostic_Alarm_[<WS>]= [<WS>]<Master_Diagnostic_Alarm_supp> | Master_Process_Alarm_supp[<WS>]= [<WS>]<Master_Process_Alarm_supp> | Master_Pull_Plug_Alarm_supp[<WS>]= [<WS>]<Master_Pull_Plug_Alarm_supp> | Master_Status_Alarm_supp[<WS>]= [<WS>]<Master_Status_Alarm_supp> | Master_Update_Alarm_supp[<WS>]= [<WS>]<Master_Update_Alarm_supp> | Master_Manufacturer_Specific_Alarm_supp[<WS>]= [<WS>]<Master_Manufacturer_Specific_Alarm_supp> | Master_Extra_Alarm_SAP_supp[<WS>]= [<WS>]<Master_Extra_Alarm_SAP_supp> | Master_Alarm_Sequence_Mode[<WS>]= [<WS>]<Master_Alarm_Sequence_Mode> | Master_Alarm_Type_Mode_supp[<WS>]= [<WS>]<Master_Alarm_Type_Mode_supp> | X_Master_Prm_SAP_supp[<WS>]=

	[<WS>]< X_Master_Prm_SAP_supp> | DXB_Master_supp[<WS>]= [<WS>]<DXB_Master_supp> | Isochron_Mode_Synchronised[<WS>]= [<WS>]<Isochron_Mode_Synchronised> | Freeze_Mode_supp[<WS>]=[<WS>]<Freeze_Mode_supp> | Sync_Mode_supp[<WS>]=[<WS>]<Sync_Mode_supp> | Auto_Baud_supp[<WS>]=[<WS>]<Auto_Baud_supp> | Set_Slave_Add_supp[<WS>]= [<WS>]<Set_Slave_Add_supp> | User_Prm_Data_Len[<WS>]=[<WS>]<User_Prm_Data_Len> | User_Prm_Data[<WS>]=[<WS>]<User_Prm_Data> | Min_Slave_Intervall[<WS>]= [<WS>]<Min_Slave_Intervall> | Modular_Station[<WS>]=[<WS>]<Modular_Station> | Max_Module[<WS>]=[<WS>]<Max_Module> | Max_Input_Len[<WS>]=[<WS>]<Max_Input_Len> | Max_Output_Len[<WS>]=[<WS>]<Max_Output_Len> | Max_Data_Len[<WS>]=[<WS>]<Max_Data_Len> | Fail_Safe[<WS>]=[<WS>]<Fail_Safe> | Fail_Safe_required[<WS>]= [<WS>]<Fail_Safe_required> | Diag_Update_Delay[<WS>]=[<WS>]<Diag_Update_Delay> | Max_Diag_Data_Len[<WS>]=[<WS>]<Max_Diag_Data_Len> | Modul_Offset[<WS>]=[<WS>]<Modul_Offset> | Max_User_Prm_Data_Len[<WS>]= [<WS>]<Max_User_Prm_Data_Len> | Slave_Family[<WS>]=[<WS>]<Family_Name> | Prm_Block_Structure_supp[<WS>]= [<WS>]<Prm_Block_Structure_supp> | Prm_Block_Structure_req[<WS>]= [<WS>]<Prm_Block_Structure_req> | Jokerblock_supp[<WS>]=[<WS>]<Jokerblock_supp> | [<Jokerblock-Def>] | PrmCmd_supp[<WS>]=[<WS>]<PrmCmd_supp> | Slave_Max_Switch_Over_Time[<WS>]= [<WS>]<Max_Switch_Over_Time> | Slave_Redundancy_supp[<WS>]= [<WS>]<Slave_Redundancy_supp> | Ident_Maintenance_supp[<WS>]= [<WS>]<Ident_Maintenance_supp> | Time_Sync_supp[<WS>]=

	[<WS>]<Time_Sync_supp> | DPV1_Slave[<WS>]=[<WS>]<DPV1_Slave> | C1_Read_Write_supp[<WS>]= [<WS>]<C1_Read_Write_supp> | C2_Read_Write_supp[<WS>]= [<WS>]<C2_Read_Write_supp> | C1_Max_Data_Len[<WS>]=[<WS>]<Max_C1_Data_Len> | C2_Max_Data_Len[<WS>]=[<WS>]<Max_C2_Data_Len> | C1_Response_Timeout[<WS>]= [<WS>]<C1_Response_Timeout> | C2_Response_Timeout[<WS>]= [<WS>]<C2_Response_Timeout> | C1_Read_Write_required[<WS>]= [<WS>]<C1_Read_Write_required> | C2_Read_Write_required[<WS>]= [<WS>]<C2_Read_Write_required> | C2_Max_Count_Channels[<WS>]= [<WS>]<Max_Count_C2_Channels> | Max_Initiate_PDU_Length[<WS>]= [<WS>]<Max_Initiate_PDU_Length> | <Alarm_Support> | Extra_Alarm_SAP_supp[<WS>]= [<WS>]<Extra_Alarm_SAP_supp> | Alarm_Sequence_Mode_Count[<WS>]= [<WS>]<Alarm_Sequence_Mode_Count> | Alarm_Type_Mode_supp[<WS>]= [<WS>]<Alarm_Type_Mode_supp> | <Alarm_Requirement> | DPV1_Data_Types[<WS>]=[<WS>]<DPV1_Data_Types> | WD_Base_1ms_supp[<WS>]=[<WS>]<WD_Base_1ms_supp> | Check_Cfg_Mode[<WS>]=[<WS>]<Check_Cfg_Mode> | <Unit_Diag_Bit[<WS>](<Bit>)[<WS>]= [<WS>]<Diag_Text> {0<=Bit<=495} | Unit_Diag_Not_Bit[<WS>](<Bit>)[<WS>]= [<WS>]<Diag_Text> {0<=Bit<=495} | Unit_Diag_Bit_Help[<WS>](<Bit>)[<WS>]= [<WS>]<Help_Text> {0<=Bit<=495} | Unit_Diag_Not_Bit_Help[<WS>](<Bit>)[<WS>]= [<WS>]<Help_Text> {0<=Bit<=495} | <Unit-Diag-Area-Def> | <Channel-Diag-Definition> | <Ext-User-Prm-Data-Const>

	| <Ext-User-Prm-Data-Ref> | <X-Prm-List> | <User-Definition> | <Max_iParameter_Size>
15)	<Ext_Module_Prm_Len> = <Unsigned8>
14)	<F-Ext-Module-Prm-Data-Len> = F_Ext_Module_Prm_Data_Len[<WS>]= [<WS>]<Ext_Module_Prm_Len><LineEnd>
13)	<X-Ext-Module-Prm-Data-Len> = X_Ext_Module_Prm_Data_Len[<WS>]= [<WS>]<Ext_Module_Prm_Len><LineEnd>
12)	<Ext-Module-Prm-Data-Len> = Ext_Module_Prm_Data_Len[<WS>]= [<WS>]<Ext_Module_Prm_Len><LineEnd>
11)	<Ext-User-Prm-Data-Def> = ExtUserPrmData[<WS>]=[<WS>]<Reference_Number><WS> <Ext_User_Prm_Data_Name> | <"[SlotNumber]"><LineEnd> <Data_Type_Name><WS><Default_Value> [<WS><Min_Value>[<WS>]-[<WS>]<Max_Value> | <WS><Allowed_Values>]<LineEnd> [<Prm-Text-Ref>] [<Changeable>] [<Visible>] EndExtUserPrmData
10)	<Text_Item> = Text(<Prm_Data_Value>)[<WS>]= [<WS>]<Text><LineEnd>
9)	<Text_List> = <Text_Item> | <Text_List><Text_Item>
8)	<Prm-Text-Def> = PrmText[<WS>]=[<WS>]<Reference_Number><LineEnd> <Text_List> EndPrmText
7)	<Module-Def-Item> = <Info_Text> | <Channel-Diag-Definition> | <Ext-User-Prm-Data-Const> | <Ext-User-Prm-Data-Ref> | <X_Ext_User_Prm_Data_Const> | <X_Ext_User_Prm_Data_Ref> | <F_Ext_User_Prm_Data_Const> | <F_Ext_User_Prm_Data_Ref>

	| <Alarm_Requirement> | <Ext-Module-Prm-Data-Len> | <X-Ext-Module-Prm-Data-Len> | <F-Param-List> | <F-Ext-Module-Prm-Data-Len> | [<Data-Area-Def>] | Ident_Maintenance_supp[<WS>]= [<WS>]<Ident_Maintenance_supp> | C1_Read_Write_supp[<WS>]= [<WS>]<C1_Read_Write_supp> | C1_Read_Write_required[<WS>]= [<WS>]<C1_Read_Write_required> | <Alarm_Support> | <Alarm_Requirement> | <Subsys-Module-Dir-Index-Def> | <User-Definition>
6)	<Module-Def-List> = <Module-Def-Item> | <Module-Def-List><Module-Def-Item>
5)	<Module-Definition> = Module[<WS>]=[<WS>]<Mod_Name><WS><Config><LineEnd> <Module-Reference> [<Module-Def-List>] EndModule
4)	<GSD-Item> = [<Prm-Text-Def>] | [<Ext-User-Prm-Data-Def>] | [<X-Prm-List>] | <Unit-Def-Item> | <Module-Definition> | [<Slot-Def>] | [<Ph-Interface-Def>] | [<Subsys-Dir-Index-Def>] | [<Isochron-Mode-List>] | [<DXB-List>] | [Unit-Diag-Type-Def]
3)	<GSD-Line> = <LineStart><GSD-Item><LineEnd>
2)	<GSD-List> = <GSD-Line>|<GSD-List><GSD-Line>
1)	<GSD> = [<Any-Text>] <LineStart>#Profibus_DP<LineEnd> <GSD-List>

	[<LineStart>#<Keyword><LineEnd> [<Any-Text>]]

Annex B
(informative)

Evolution of GSD

B.1 Preface to GSD Revision 5

PROFIBUS is standardized in EN 50170. The new versions of PROFIBUS named DP-V1 and DP-V2 are specified in IEC 61158-series Edition 3 and IEC 61784-1. In GSD Revision 5 now all references to EN are rewritten to these IEC International Standards and the terms are adjusted.

Extensions

- Jokerblock according to the Block Structure of the extended parameterization within the user parameterization data
- Slave Redundancy according [1]
- I&M functions according [6]
- Clock Synchronisation
- Help Text for status and error messages
- Data Area description for modules
- Automatically Slot Number insertion
- Extended description for Block Structure
- Extended description for Isochronous Mode

B.2 Preface to GSD Revision 4

After its version DP-V1 has succeeded in proving its quality in production and process industry, PROFIBUS now takes a new leap up to version DP-V2.

In process and production technology, the trend towards more "intelligent" – i.e. more powerful – sensors and actuators is unstoppable. More and more powerful and increasingly fast microprocessors take over tasks from central controllers or permit the utilization of physical effects that were unexploited up to now. Where drives are concerned, for example, the palette of different capacities ranges from speed-controlled units via position-controlled units up to devices that are designed for special technological tasks. Since usually several drives (axes) must run strictly synchronously, there are different requirements on the structure of a closed loop. With "simple" speed-controlled drives, this still means a high control effort in the controller and high clock synchronization between controller, bus and drive. With increasing performance, more and more information is relocated "downwards" and the drives need more direct data exchange among each other.
Due to the required high performance values, PROFIBUS has now implemented the requirements based upon the base communication, and specified in the new version DP-V2.

Extensions

- Isochronous Mode
- Data Exchange with Broadcast
  - Subscriber_supp, ...
- F-Parameter
- Extended Parameterization
- Extended Diagnostic Description
- Automatically SlotNumber mapping
- Subsystems
  - HMD ...(HART Master Devices)
- Extended Description of MaxTsdr for Optimizing

B.3 Preface to GSD Revision 3

The development of the PROFIBUS product range also entails enhancements as regards the device properties and features to be described in the GSD files. These developments, in particular the introduction of DP-V1, new physical interfaces and requirements from PROFIBUS PA are the reason for the extension to the GSD file for the present GSD Revision 3.

A primary goal of the revision was to define new keywords to support the configuration of PROFIBUS devices with new features.

NOTE Examples as templates for own developments are available at www.profibus.com -> GSD Library.

In general new PROFIBUS devices supporting new features should get a new Ident_Number. But with introducing PROFIBUS extensions, existing devices will be updated. These devices will be compatible regarding the original DP-V0 functions. That's why it is possible, that they keep the same Ident_Number.

In practice this will result into the following scenario:

	Original GSD	New GSD
Original device	OK	See b)
New device	See a)	OK

a.) Case of replacement and maintenance. New devices have to be compatible with original GSD. Otherwise a new Ident_Number has to be assigned to the new device.

b.) With the new GSD new features can be selected for the original device, which are not supported. This can cause malfunctions. This is the reason why both GSD can be administrated by the configuration tool. The versions of the GSD must differ in the following items:

- Manufacturer specific characters of the GSD file name
- Keyword Revision
- Keyword Model_Name

Additionally it has to be ensured that:

- Shipping of old devices will be stopped when new devices are available
- The assignment from device release to GSD is well described

B.4 Preface to GSD Revision 2

The development of the PROFIBUS product range also entails enhancements as regards the device properties and features to be described in the GSD files. These developments, in particular the introduction of PROFIBUS PA and the associated new transmission rates, are the reason for the extension to the GSD file for the present GSD Revision 2.

A primary aim of the revision was to improve the readability of the formal description of the GSD file. The individual rules in this clause have been numbered in order to enable better referencing. Rules that left room for interpretation have been made more precise. Rules that unnecessarily limited the format of the GSD file and thus made it more difficult to create and read GSD files have been relaxed.

The changes to the informal description of the keywords since GSD Revision 1 essentially boil down to the addition of the keywords for the new transmission rates.

In the formal description the following changes have been made since GSD Revision 1.

Change

- Description of continuation lines
- Description of a beginning of a GSD line
- Description of white spaces in octet strings
- Description of white spaces in User definitions
- Support of new transmission rates
- MaxTsdr for new transmission rates
- Trdy for new transmission rates
- Tqui for new transmission rates
- Tset for new transmission rates
- Tsdi for new transmission rates
- Subfamily_Name description
- Change of the reference number to Unsigned16
- Last_Bit limited to 495
- Extension of Unit-Def-Items
- Value range to {0<=Bit<=495}
- Ext-Module-Prm-Len description
- Is concluded with EndExtUserPrmData
- Replacement of the previous module definition
- Replacement of the previous GSD description
- User_Prm_Data_Def has been deleted

Bibliography

[1] PNO Document 2.212: *PROFIBUS Guideline "Specification Slave Redundancy"*
[see www.PROFIBUS.com]

[2] PNO Document 2.262: *PROFIBUS Guideline "Profibus RS 485-IS User and Installation Guideline"*
[see www.PROFIBUS.com]

[3] PNO Document 2.312: *Application Guideline "Profile for HART on PROFIBUS"*
[see www.PROFIBUS.com]

[4] PNO Document 3.092: *PROFIBUS Profile "PROFIsafe - Profile for Safety Technology"*
[see www.PROFIBUS.com]

[5] PNO Document 3.172: *Technical Specification "PROFIdrive - Profile Drive Technology"*
[see www.PROFIBUS.com]

[6] PNO Document 3.502: *PROFIBUS Profile Guidelines "Part 1 – Identification & Maintenance Functions"*
[see www.PROFIBUS.com]

PROFIBUS Nutzerorganisation e.V.
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany
Phone: +49 (0) 721 / 96 58 590
Fax: +49 (0) 721 / 96 58 589
e-mail: info@profibus.com
http://www.profibus.com