IEC 62453-309

Edition 3.0 2022-09
REDLINE VERSION

INTERNATIONAL STANDARD

Field device tool (FDT) interface specification – Part 309: Communication profile integration – IEC 61784 CPF 9

IEC 62453-309:2022-09 RLV(en)

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

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

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

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

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

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

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC 62453-309

Edition 3.0 2022-09
REDLINE VERSION

INTERNATIONAL STANDARD

colour inside

Field device tool (FDT) interface specification –
Part 309: Communication profile integration – IEC 61784 CPF 9

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.05; 35.110

ISBN 978-2-8322-5665-7

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

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 309: Communication profile integration – IEC 61784 CPF 9

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

This redline version of the official IEC Standard allows the user to identify the changes made to the previous edition IEC 62453-309:2016. A vertical bar appears in the margin wherever a change has been made. Additions are in green text, deletions are in strikethrough red text.

IEC 62453-309 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

This third edition cancels and replaces the second edition published in 2016. This edition constitutes a technical revision.

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

- corrections in regard to accessing information in the respective device and
- corrections in regard to describing support for different protocol versions.

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

Draft	Report on voting
65E/907/FDIS	65E/936/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

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

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

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

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

INTRODUCTION

This part of IEC 62453 is an interface specification for developers of FDT[1] (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

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

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

Figure 1 shows how IEC 62453-309 is aligned in the structure of the IEC 62453 series.

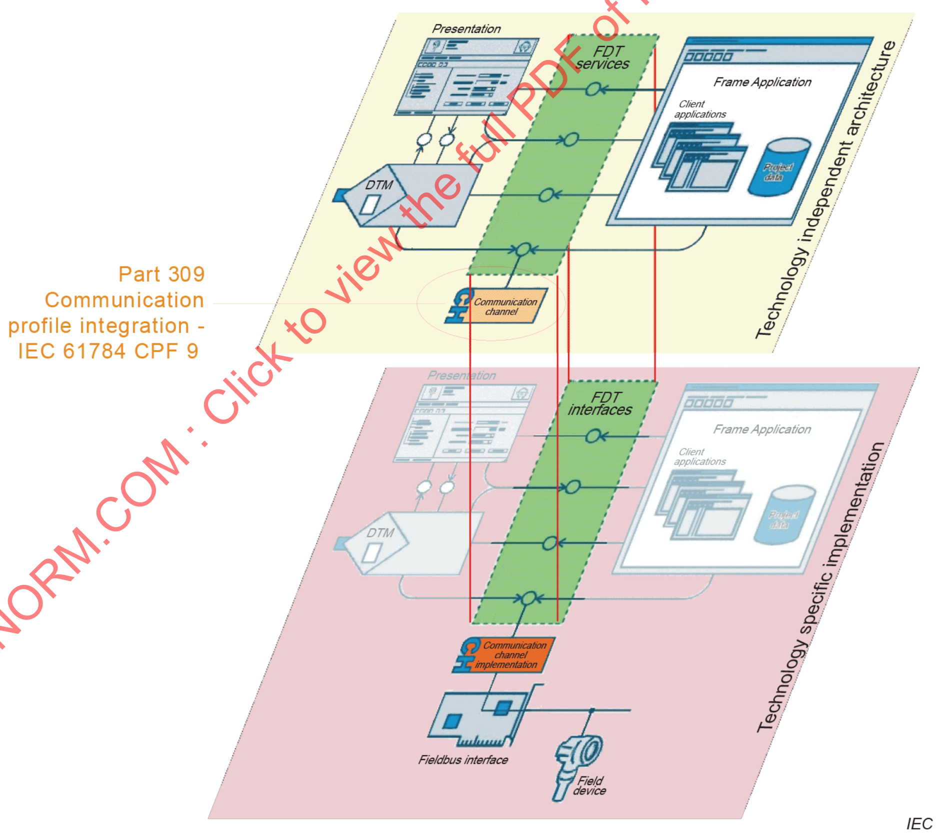


Figure 1 – Part 309 of the IEC 62453 series

[1] FDT® is a trademark of products supplied by FDT Group AISBL. This information is given for convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 309: Communication profile integration – IEC 61784 CPF 9

1 Scope

Communication Profile Family 9 (commonly known as HART®[2]) defines communication profiles based on IEC 61158-5-20 and IEC 61158-6-20. The basic profile CP 9/1 is defined in IEC 61784-1.

This part of IEC 62453 provides information for integrating the HART® technology into the FDT standard (IEC 62453-2).

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

This document neither contains the FDT specification nor modifies it.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61158-5-20, *Industrial communication networks – Fieldbus specifications – Part 5-20: Application layer service definition – Type 20 elements*

IEC 61158-6-20, *Industrial communication networks – Fieldbus specifications – Part 6-20: Application layer protocol specification – Type 20 elements*

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

IEC 62453-1:–[3], *Field device tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:–[3], *Field device tool (FDT) interface specification – Part 2: Concepts and detailed description*

---

[2] HART® and WirelessHART® are trade names of products supplied by ~~HART Communication Foundation~~ FieldComm Group. This information is given for convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

[3] ~~To be published concurrently with this standard.~~ Under preparation. Respective stage at the time of publication: IEC/CCDV 62453-1:2022 and IEC/RFDIS 62453-2:2022.

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http:www.iso.org/obp

3.1.1
burst mode
mode in which the field device generates response telegrams without request telegram from the master

3.2 Abbreviated terms

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

BACK	Burst ACKnowledge
C8PSK	Coherent 8-way Phase Shift Keying, HART communication layer as defined in HCF_SPEC-60, Revision 1.0
DR	delayed response
EDD	Electronic Device Description
FSK	Frequency Shift Keying, HART communication layer as defined in HCF_SPEC-54, Revision 8.1
HART	Highway Addressable Remote Transducer

3.3 Conventions

3.3.1 Data type names and references to data types

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

3.3.2 Vocabulary for requirements

The following expressions are used when specifying requirements:

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

3.3.3 Use of UML

Figures in this document are using UML notation as defined in IEC 62453-1:–, Annex A.

4 Bus category

IEC 61784 CPF 9 protocol is identified in the protocolId element of structured data type 'fdt:BusCategory' by the following unique identifiers (see Table 1):

Table 1 – Protocol identifiers

Identifier value	ProtocolId	Display String	Description
036D1498-387B-11D4-86E1-00E0987270B9	HART_Basic	'HART'	Support of IEC 61784 CPF 9 protocol over FSK communication with basic functionality (deprecated)
98503B8F-0FFB-4EB7-BB67-F4D6BD16DB8D	HART_FSK	'HART FSK'	Support of HART protocol over FSK communication with complete functionality
74D29D22-F752-40EF-A747-ACA72C791155	HART_Wireless	'HART Wireless'	Support of WirelessHART protocol
58001A08-C178-4A59-A76B-9EF9111CB83D	HART_RS485	'HART RS485'	Support of HART protocol over RS485 communication
EF708CB7-A2A1-42AF-890C-15CEB680CC12	HART_Infrared	'HART Infrared'	Support of HART protocol over Infrared communication
D122D172-F0C7-4B03-965B-512CD4C0871E	HART_IP	'HART IP'	Support of HART over IP protocol

The 'HART_Basic' protocol is maintained for backward compatibility only (e.g. for interaction with DTMs according to IEC 62453-309:2009). The other protocol identifiers provide a better support for planning of network topologies and for establishment of connections between DTM and respective device. For DTMs complying with this document, support for one of the other protocols is mandatory.

Within this document, the other protocols (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP) are referenced as 'Extended_HART' protocols. (E.g. for definitions that apply to all protocols except 'HART_Basic'.)

Table 2 defines which PhysicalLayer can be used together with the BusCategory defined in Table 1.

Table 2 – Definition of PhysicalLayer

PhysicalLayer Id value	PhysicalLayer name value	Description
BAB2091A-C0A7-4614-B9DE-FCC2709DCF5D	HART FSK Physical Layer	Support of HART FSK physical layer
B9F1A250-AC94-4487-8F25-A8F3F8F89DC5	WirelessHART Physical Layer	Support of WirelessHART physical layer
036D1591-387B-11D4-86E1-00E0987270B9	HART RS-485 Physical Layer	Support of HART devices using RS-485 communication
AE4119EF-B9FD-429c-B244-134DB182296A	HART Infrared Physical Layer	Support of HART devices using infrared communication
307dd808-c010-11db-90e7-0002b3ecdcbe	10BASET	HART Ethernet based Physical Layers
307dd809-c010-11db-90e7-0002b3ecdcbe	10BASETXHD	
307dd80a-c010-11db-90e7-0002b3ecdcbe	10BASETXFD	
307dd80b-c010-11db-90e7-0002b3ecdcbe	10BASEFLHD	
307dd80c-c010-11db-90e7-0002b3ecdcbe	10BASEFLFD	
307dd80d-c010-11db-90e7-0002b3ecdcbe	10BASEFXHD	
307dd80e-c010-11db-90e7-0002b3ecdcbe	10BASEFXFD	

PhysicalLayer Id value	PhysicalLayer name value	Description
307dd80f-c010-11db-90e7-0002b3ecdcbe	100BASETXHD	
307dd810-c010-11db-90e7-0002b3ecdcbe	100BASETXFD	
307dd811-c010-11db-90e7-0002b3ecdcbe	100BASEFXHD	
307dd812-c010-11db-90e7-0002b3ecdcbe	100BASEFXFD	
307dd813-c010-11db-90e7-0002b3ecdcbe	100BASELX10	
307dd814-c010-11db-90e7-0002b3ecdcbe	100BASEPX10	
307dd815-c010-11db-90e7-0002b3ecdcbe	1000BASEXHD	
307dd816-c010-11db-90e7-0002b3ecdcbe	1000BASEXFD	
307dd817-c010-11db-90e7-0002b3ecdcbe	1000BASELXHD	
307dd818-c010-11db-90e7-0002b3ecdcbe	1000BASELXFD	
307dd819-c010-11db-90e7-0002b3ecdcbe	1000BASESXHD	
307dd81a-c010-11db-90e7-0002b3ecdcbe	1000BASESXFD	
307dd81b-c010-11db-90e7-0002b3ecdcbe	1000BASETHD	
307dd81c-c010-11db-90e7-0002b3ecdcbe	1000BASETFD	
307dd81d-c010-11db-90e7-0002b3ecdcbe	10GigBASEFX	

The significant information for topology planning is the BusCategory. The PhysicalLayer (which is provided in the BusInformation data type) shall be used only for additional information.

The DataLinkLayer property is not applicable for HART and ~~has to~~ shall be set to null.

5 Access to instance and device data

5.1 General

The HART protocol has semantics defined that allow in a wide range the identification of device variables and device parameters. Most of this semantic information is defined in the standard EDD import libraries.

Clause 5 describes how the semantic information defined with the HART protocol shall be used to export device data, instance data and process data.

5.2 Process Channel objects provided by DTM

The minimum set of provided data shall be the first four provided process related values (PV, SV, …) – if available – modeled as channel references. The referenced channel shall include ranges and scaling.

A HART device communicates the process data either via its analogue channels or via digital information (e.g. by request or by burst mode). Analogue channels are always related to a dynamic variable, as specified in [1][4] chapter 8 and therefore the description of an analogue channel ~~has to~~ shall be accessed using the respective dynamic variable (e.g. the attributes of dynamic variable PV always describe the first analogue channel).

[4] Figures in square brackets refer to the bibliography.

HART distinguishes between three methods to access digital signals:

1) Access to analogue value and assigned dynamic variables (Command #3)

   IO signals can be assigned to one of the four dynamic variables PV, SV, TV, and QV. Using the command #3 the analogue value and the dynamic variables can be read without specific device knowledge.

2) Indexed access to device variables (Command #33)

   All device variable values and their units can be read using the related ~~index~~ device variable code information in command #33. ~~Up to four device variables can be read with one call of command #33. It is up to the command initiator to identify the requested variable using the related index information.~~

3) Indexed access to device variable classification and device variable status (Command #9)

   Command #9 ~~is an extension of~~ provides more information than command #33. Beside of the value and unit also a classification and the variable status can be determined. ~~The status information contains data quality, limit status, and device family status.~~

The command initiator determines by means of the HART specification which commands will be used.

5.3 DTM services to access instance and device data

The services InstanceDataInformation and DeviceDataInformation shall provide access to at least all parameters of the Universal and Common Practice commands (as far as the device supports the function).

Furthermore, the Response Byte 0 and the Response Byte 1 for each command shall be exposed.

The services InstanceDataInformation and DeviceDataInformation may also provide access to device specific parameters (e.g. diagnostic information).

6 Protocol-specific behavior

6.1 Overview

There is only one protocol-specific sequence defined for IEC 61784 CPF 9: burst mode subscription.

This sequence explains how the sequence “Device initiated data transfer”, defined in IEC 62453-2, is applied in context of burst telegrams as defined by IEC 61784 CPF 9.

Additionally, Clause 6 provides information regarding:

- usage of device addressing information,
- support of extended command codes,
- handling of communication failures,
- handling of delayed responses, and
- management of physical topologies.

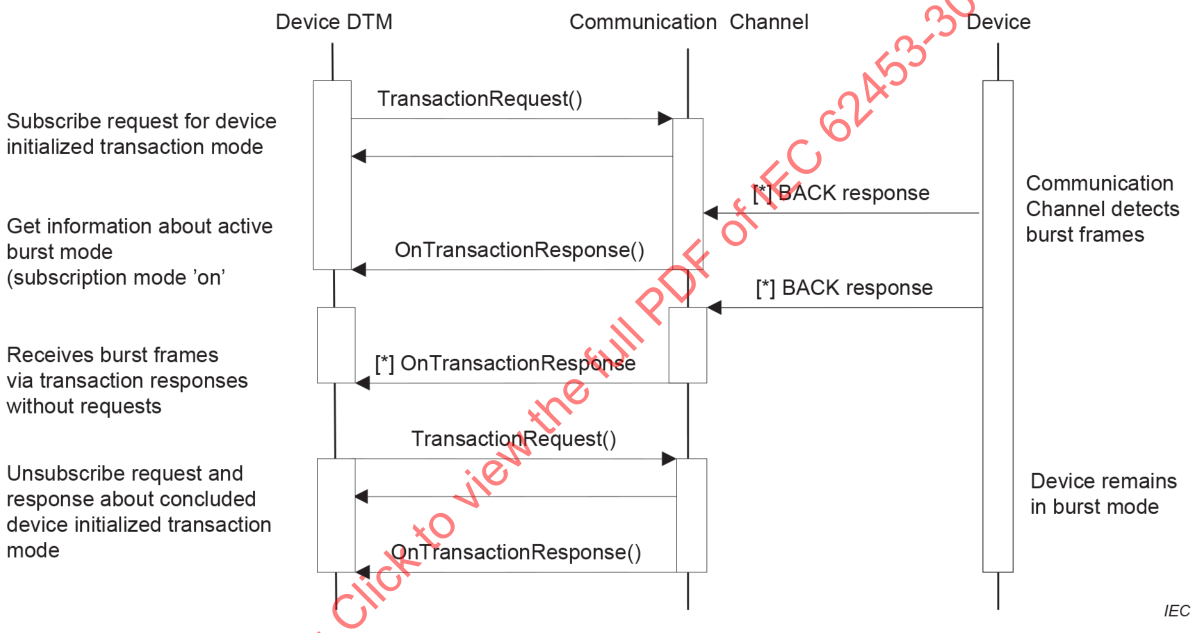
6.2 Burst mode subscription

A subscription to device-initiated data transfer can be requested by sending a transaction request with SubscribeRequest content (see Figure 2). The Communication Channel may detect if the device is already in burst mode.

NOTE In HART 5 this can be detected only when burst frames are received from the device. In HART 6 the burst mode can be detected using command #105.

The Communication Channel answers to a SubscribeRequest with a SubscribeResponse content. If burst frames are received, the device is in burst mode and burstModeDetected value is set to TRUE. This means that Device DTM will start to receive burst messages via the transaction response mechanism. In the case that no burst messages were received, burstModeDetected value is set to FALSE. It is up to Device DTM to set device into burst mode. Then Device DTM may call a transaction request with SubscribeRequest content again in order to receive burst messages.

In order to unsubscribe, the Device DTM sends a transaction request with a UnsubcribeRequest. The Communication Channel answers with a UnsubscribeResponse where burstModeDetected value is set to FALSE. The Device DTM will not receive any more burst information via the transaction response mechanism. The Communication Channel does not switch off the burst mode in the device. The Device DTM may switch burst mode on or off by using normal transaction requests (command #109). This is independent of the subscription.



NOTE BACK means Burst ACKnowledge.

Figure 2 – Burst mode subscription

6.3 Usage of device addressing information

HART is a connectionless master/slave protocol. Transaction requests are always addressed using unique device address information (a 5 byte integer), the so called long address.

Device addressing in HART therefore is mainly focused to determine this long address.

There are currently three ways possible to determine the long address.

1) short address

   The short address is a number between 0 and 63 (for HART version 5 only 0 to 15). In the context of a direct connection to the device the short address is unique and allows to read the long address using command #0.

2) short tag

   With command #11 the long address information can be requested for a device with a specific short tag. Such requests are especially used for installations with a huge amount of connected HART devices. All HART multiplexer devices and other HART communication structures have to shall support this command.

3) long tag

Since HART version 6 the long tag was introduced. The long tag can store more information. For devices with HART version less than 6 instead of long tag, message is used. With command 21 a similar method to determine the long address is possible. Command 21 is usually supported by highly modular devices or Gateways.

HART Version 6 introduced the long tag, which at 32 characters has 24 more characters than the short tag, and can thus provide for a larger number of device label options. For devices with HART version less than 6 instead of long tag, message is used. With command #21 the long address information can be requested for a device with a specific long tag. Command #21 is usually supported by highly modular devices or gateways.

A Device DTM is responsible to provide and store all information that is used for resolving the long address of a connected device. The support of the addressing methods depends on the type of DTM:

- DTMs that only have 'HART' protocol defined as required protocol support the device addressing using the short address only. This information is managed according to the description in 9.3.
- DTMs that have at least one of the 'Extended_HART' protocols defined as required protocol use the storage of the short tag for compatibility reason like described in 9.3, but also store additionally required information as defined in 9.4.

It has therefore to maintain the data for all three address resolving methods. The DTM responsible to connect to the communication hardware shall select the method and provide means for a user to input the address information.

Besides the addressing topic, there are also different approaches for manufacturer and device type identification depending on the supported version of HART. HART versions up to HART 6 use one byte values. HART versions starting from HART 7 (and newer) use a two byte value. The two byte values are also stored in the data types described in 9.4.

A Communication DTM uses the addressing information provided by the Device DTM in order to resolve the long address as described above.

6.4 Extended Command Numbers

The HART command number is defined as a one byte unsigned integer. When starting with the specification of device family commands for HART, 6 HCF started to define extended command number for better specification clarity. Extended command numbers are applied only for device family commands defined by HART. Beginning with HART Version 6, an extended command number format, using two bytes instead of one, was defined to allow for more than 255 command numbers. This format uses command number 31, which was previously reserved, to indicate that the request is using the extended command number format.

According to the specification in [2] 8.1.2, extended commands are implemented with command #31 by using the extended command number as first two bytes in the request and response section.

In FDT, all commands with extended command numbers have to shall be implemented by the Child DTM using command #31.

6.5 Handling of communication failures and time-outs

HART uses a device-specific handling of communication errors. The protocol defines a section in the response frame that can carry communication failure information.

If, during execution of a communication request to a Communication Channel, a communication error occurs on the HART physical layers (this also includes time-outs), no Abort message shall be sent to the Child DTM, but the transaction request shall be responded with a set of data that describes the communication error as defined in HART [1].

In case of such a communication failure, the Device DTM has the responsibility to perform the error handling to recover from the communication failure.

Only in case of a connection-based communication break (e.g. Ethernet connection to a HART modem), the Communication Channel shall send an Abort signal to the device DTM.

6.6 Handling of delayed responses

HART defines strict time constraints for responses to a request within a HART transaction. In case a device is unable to fulfill the time constraints, it can initiate a delayed response (DR) sequence. In order to support DR handling within nested communication, Subclause 6.6 defines the handling within FDT.

The responsibility to handle the DR responses from the device is located at the DTM that represents the device. The Communication DTM and Gateway DTMs (if used) ~~have to~~ shall ensure that DR responses are communicated correctly to the respective DTM. An example for such a delayed response handling is shown in Figure 3.

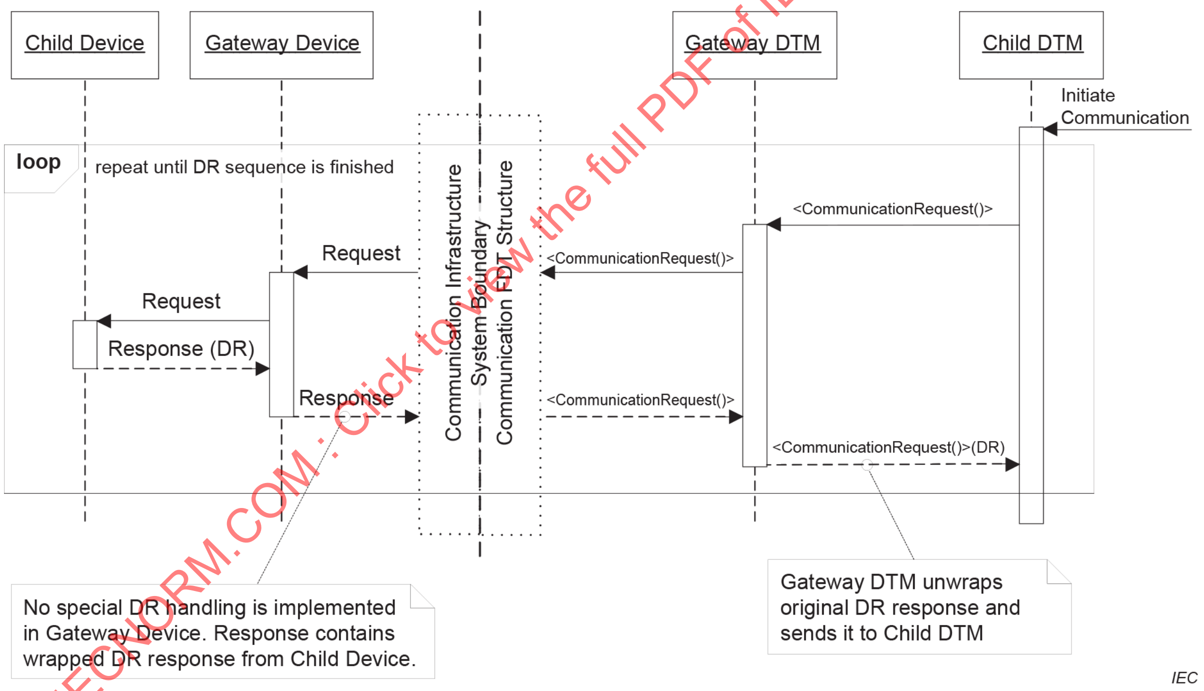


Figure 3 – Handling of delayed responses (scenario 1)

It is also possible that the two partners of a DR sequence are both devices. For example, a gateway device (e.g. WirelessHART Gateway) might execute a delayed response sequence with a child device (e.g. WirelessHART Adapter). In this case, the gateway device is responsible to handle the DR of the child device. The delayed responses will not reach the respective Child DTM. If the gateway device is unable to handle the DR directly, the gateway device itself could send DRs to the Gateway DTM. In such a case, the DRs would have to be handled by the respective Gateway DTM. Usually, the nested communication concept reflects the interaction between the devices. In the case described here, this is not possible and the implementation ~~has to~~ shall follow the sequence shown in Figure 4.

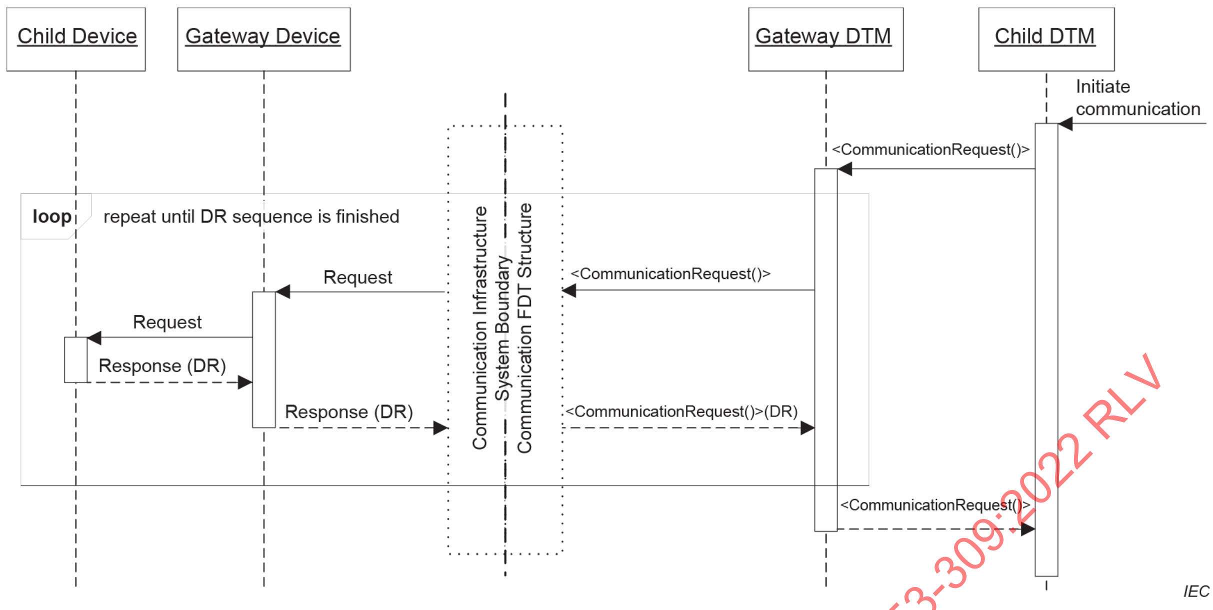


Figure 4 – Handling of delayed responses (scenario 2)

A DR sequence might take a long time that might disturb the usage of the FDT Frame Application and that might block user interaction. There is no timeout time definition existing for DR sequences and neither the DTM itself nor any other DTM in the nested communication chain is capable to initiate a timeout that could recover the system. Timeout time in such a case is application-dependent and shall be configurable by the user. When a DR sequence lasts an unreasonable amount of time, it shall be an aim to involve the user. If a DR sequence is used in a user interface,~~ free environments~~ then configurable timeout mechanisms shall be implemented.

To handle DR responses with a reliable interoperability, the following rules~~ have to~~ shall be fulfilled:

- The DTM of a device that might send DR responses~~ has to~~ shall handle the DR responses of the device.
- DR responses that are not handled by other devices~~ have to~~ shall be propagated to the DTM that represents the device that sends the responses.
- A DTM shall be aware that it will not receive DR responses from the device, when the DR responses are handled by the parent device.
- A DTM that handles DR responses~~ has to~~ shall implement a user configurable timeout management that must allow the user to set a timeout.

6.7 Topologies with mixed HART protocols

6.7.1 General

HART DTMs using 'Extended_HART' protocols may also support the 'HART_Basic' protocol, in order to ensure compatibility with existing HART DTMs.

'Extended_HART' protocols were defined for better distinction between the different HART communication types. Using 'Extended_HART' and 'HART_Basic' protocols at the same time needs well defined processes to guarantee interoperability.

6.7.2 Behavior of DTMs supporting 'Extended_HART' only

The topology validation is performed by the Frame Application (reference). If the Communication Channel receives a call to ValidateAddChild(), it~~ has to~~ shall verify whether the given device type requires a suitable protocolId.

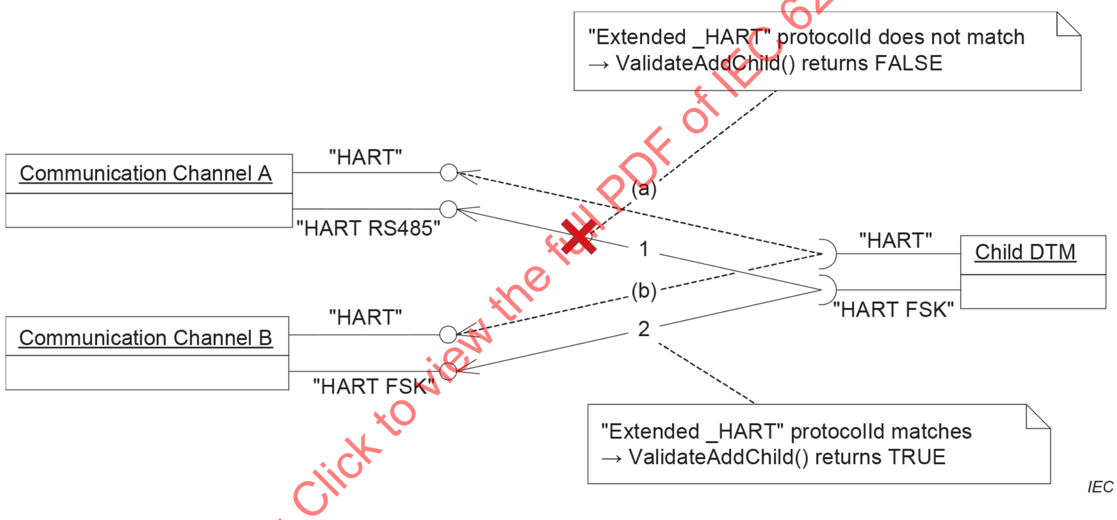
The behavior of such a DTM in a ValidateAddChild call is:

- If a match cannot be found, the ValidateAddChild() call shall be answered with FALSE.
- If a match was found, the ValidateAddChild() call shall be answered with TRUE. During the call to OnAddChild(), the Parent DTM sets the activeProtolID in the Child DTM to the current protocolId.

6.7.3 Behavior of DTMs supporting ‘Extended_HART’ and ‘HART_Basic’

When creating topologies, a Frame Application ~~will usually~~ shall check the communication compatibility of a Child DTM and a Parent DTM by comparing the lists of supported and required protocols. Based on ‘Extended_HART’ protocols, a more effective topology validation is possible, but if both DTMs support additionally the ‘HART_Basic’ protocol, this may result in invalid topologies.

When for example a Communication Channel, which supports ‘HART_RS485’ and ‘HART_Basic’, and a Device DTM, which requires ‘HART_FSK’ and ‘HART_Basic’, are connected (see Communication Channel A in Figure 5), a Frame Application will allow to connect those DTMs because of the matching ‘HART_Basic’ protocolId. But in fact this is an invalid topology.



Frame Application will allow to attach Child DTM to both Communication Channels because at least (a) and (b) are possible. But the mismatch in (1) allows the ‘Communication Channel A’ to detect the mismatch and decline the attachment of the ‘Child DTM’.

Figure 5 – Behavior of DTMs supporting ‘Extended_HART’ and ‘HART_Basic’

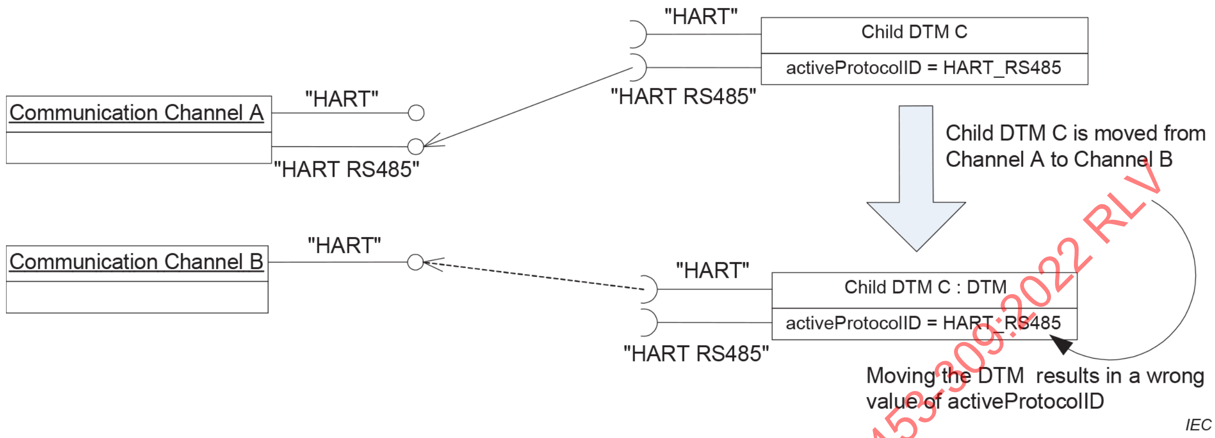
To prevent such a situation, a Communication Channel supporting ‘Extended_HART’ protocols and ‘HART_Basic’ protocol ~~has to~~ shall check during a ValidateAddChild() call if a DTM is connected that requires also ‘Extended_HART’ protocol and ‘HART_Basic’ protocol and if there exists a matching ‘Extended_HART’ protocol.

- If no match can be found in the ‘Extended_HART’ buscategories, the Communication Channel shall answer the ValidateAddChild() call with FALSE.
- If a match can be found, the ValidateAddChild() call shall be answered with TRUE. During the OnAddChild() call, the Communication Channel sets the activeProtocolID in the Child DTM.

6.7.4 Behavior of DTMs that require ‘Extended_HART’ or ‘HART_Basic’

With the attribute ‘activeProtocolID’, a DTM is informed about the current connection type in the topology. But this procedure may fail when the DTM is connected to a Communication Channel that does not implement the ‘activeProtocolID’ management.

Assuming for example that a Child DTM was connected to Communication Channel using ‘HART_RS485’ with the result that ‘activeProtocolID’ is set to ‘HART_RS485’ (see Figure 6). The Frame Application now moves the Child DTM to a Communication Channel that only supports ‘HART_Basic’. When the Child DTM now tries to establish a ‘HART_RS485’ connection, this might result in an error.



Moving DTM C from Channel A to Channel B results in an inconsistent activeProtocolID value, because DTM B does not support Enhanced_HART protocol and therefore does not set activeProtocolID.
→ Before DTM C executes a connect request, it shall validate whether the Parent DTM provides the communication related to the current activeProtoclID value.

Figure 6 – Behavior of DTMs requires ‘Extended_HART’ or ‘HART_Basic’

Therefore, it is required that a DTM which requires both ‘Extended_HART’ and ‘HART_Basic’ needs to check the capabilities of the Communication Channel before establishing a connection.

6.8 Nested communication with multiple gateways

HART supports topologies in the physical network that allow having multiple gateways in a communication chain. An example for such a topology are wired HART devices connected to a wireless adapter communicating to a wireless gateway (see 6.9).

General concept of nested communication is that a device receives the command data that was generated by its respective DTM and that the DTM receives the response data of its respective device. Also required in nested communication is that the Child DTM always is the active sender and therefore is not allowed to pass through communication sent by its Child DTM without encapsulation or transformation.

With command #77 (send to sub-device), HART defines a standard encapsulation mechanism to propagate communication through a network topology. Each request that was sent to a sub-device ~~has to~~ shall be encapsulated in a command #77 request before forwarding it to the gateway device. When a response to a command #77 is returned, the Gateway DTM ~~has to~~ shall unpack this command and send its contained response data to the respective Child DTM.

Depending on the implementation in the gateway, a command #77 might be restructured to another command structure. In this case, the Gateway DTM has the responsibility to transform incoming command #77 requests from the Child DTM to the gateway specific commands and also to restructure the resulting responses back again respectively to responses on the originally received command #77 request.

6.9 Communication- and network structures in WirelessHART

6.9.1 General

WirelessHART defines a rich and secure protocol between devices using 2,4 GHz wireless technology. Host systems are not intended to interact with the WirelessHART network directly. ~~The complex mechanism of WirelessHART is transparent for the host system.~~ Using a WirelessHART gateway device, a host system can communicate with any WirelessHART device using HART master/slave transactions, without requiring specific knowledge of the WirelessHART protocol.

HART specifies three standard types of WirelessHART devices:

1) WirelessHART gateway device:

   This device connects a WirelessHART network to the world via HART or other protocols that allow data transfer with high baud rates. It is possible to have more than one WirelessHART gateway device active in a WirelessHART network. The WirelessHART gateway devices are responsible to manage the network directory and propagate information from and to the WirelessHART devices.

   For better readability in 6.9, a WirelessHART gateway device is simply named Gateway.

2) WirelessHART field device:

   The WirelessHART field device is a device that can participate in a WirelessHART network.

   For better readability in 6.9, a WirelessHART field device is simply named Field Device.

3) WirelessHART adapter device:

   The WirelessHART adapter device is a specialized WirelessHART field device that allows the connection of HART FSK and/or 4 mA to 20 mA sub-devices to the WirelessHART network.

   For better readability in 6.9, a WirelessHART adapter device is simply named Adapter and devices connected to an Adapter are simply called Sub-Devices.

Subclause 6.9 will focus on specialties of WirelessHART and define implementation rules within FDT that are required for nested communication.

6.9.2 Network topology

Adapters are special devices that connect other HART physical layers (usually HART FSK) with the WirelessHART network as shown in Figure 7.

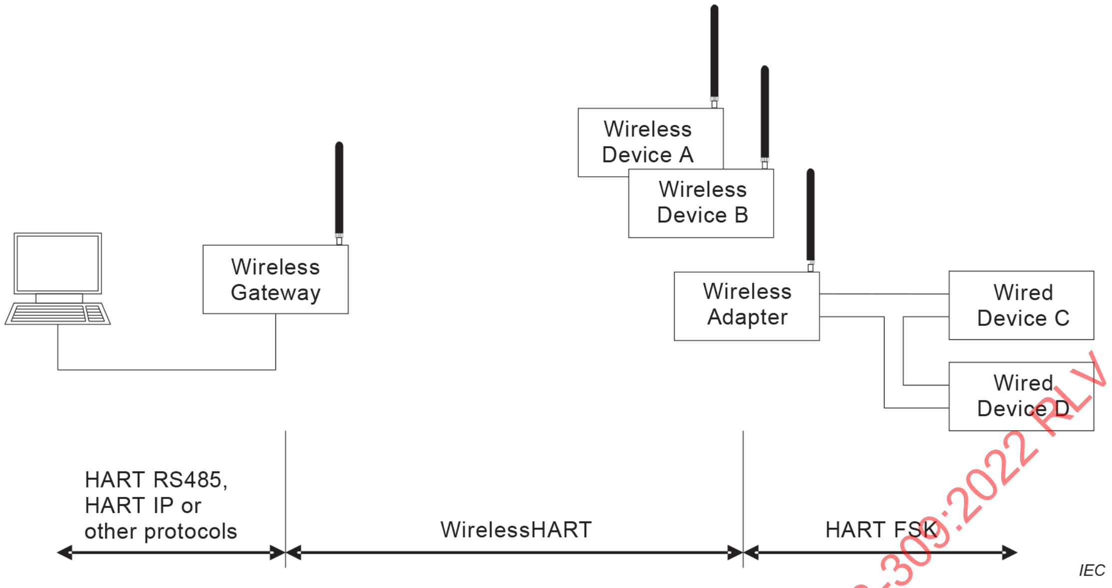
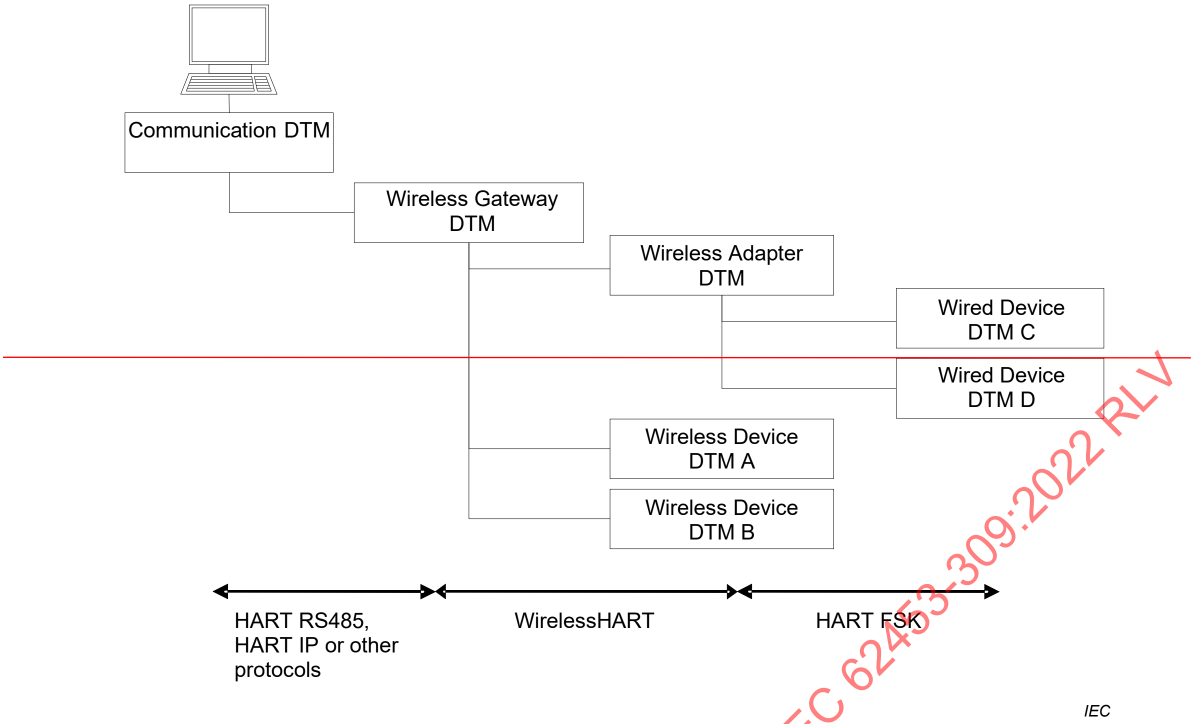


Figure 7 – Host connected to a WirelessHART gateway device

From the perspective of nested communication in FDT, the Gateway and the Adapter are both gateway devices that shall be presented as such in the network topology of an FDT Frame Application. The resulting FDT topology is shown in Figure 8.



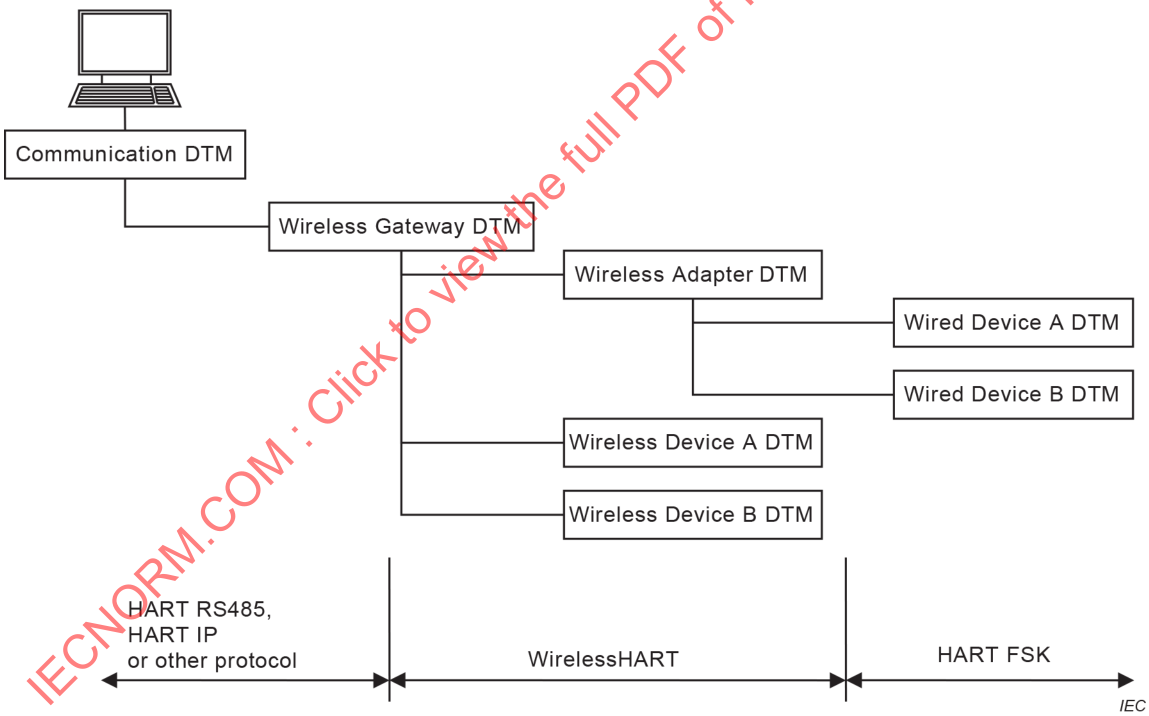


Figure 8 – FDT Topology of a WirelessHART network

An Adapter interacts with the HART FSK loop like a common HART device. It is acting as a HART Master but can also be addressed with HART transactions from another Master. Especially in service use cases, an FDT Frame Application might be connected to the HART FSK loop to directly access the Adapter. In this case, the Adapter is connected to the FDT Frame Application as a usual device in a HART FSK multidrop and multimaster scenario like shown in Figure 9.

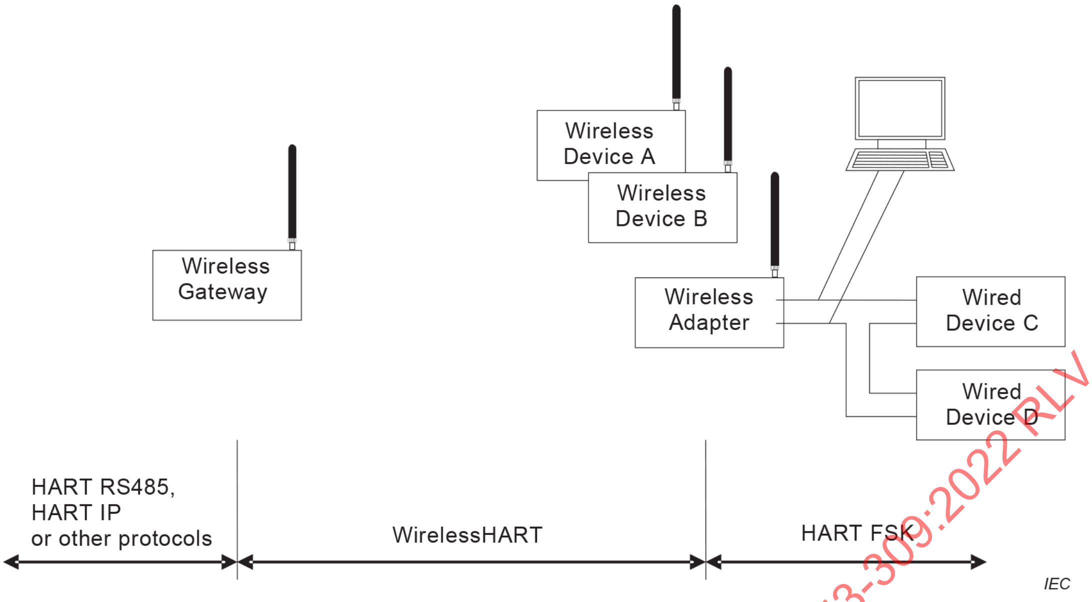


Figure 9 – Host connected to HART FSK

In this use case, the network topology in the FDT Frame shall be structured as shown in Figure 10.

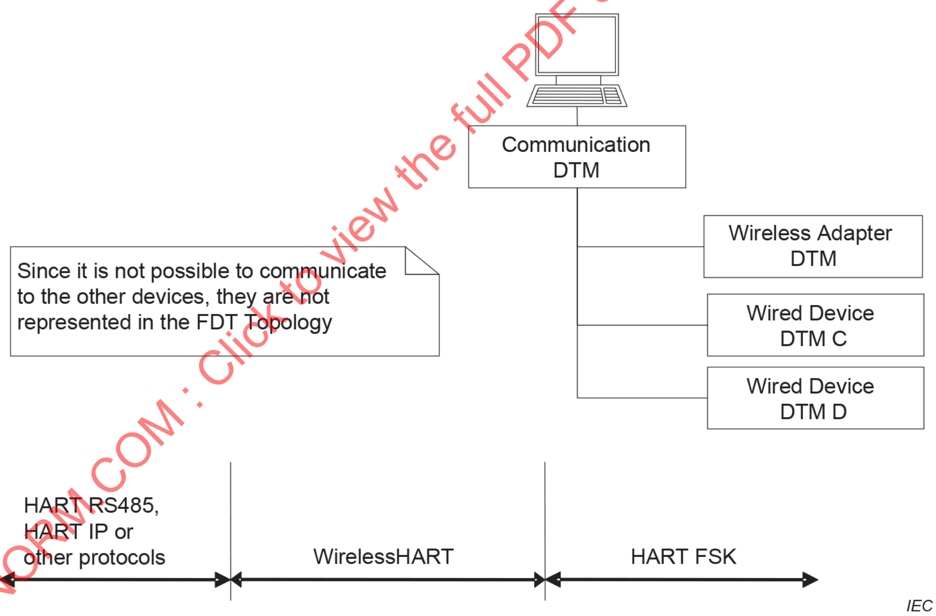


Figure 10 – FDT Topology when directly connected to a WirelessHART adapter device

Figure 8 and Figure 10 show that an Adapter DTM ~~has to~~ shall implement gateway functions when it is used in a WirelessHART FDT environment and on the other hand ~~has to~~ shall behave like a simple device when used in a HART FSK environment. In FDT 2, a DTM is always informed about changes in communication type. Using the information about the current connection type, the Adapter DTM ~~has to~~ shall implement the respective specific structural behavior.

As a summary of the above paragraphs of 6.9.2, the following rules shall be implemented:

If the Adapter is connected via WirelessHART, it:

- ~~has to~~ shall interact as a Gateway DTM;

- ~~has to~~ shall handle communication to the connected Sub-Devices (as specified in 6.7) that are attached in the topology as child DTMs;
- ~~has to~~ shall handle DR transactions as described in 6.8.

If the Adapter is connected via HART FSK, it:

- ~~has to~~ shall deny attachment of Child DTMs.
- ~~has to~~ shall deny connection to Child DTMs.

If an instance of an Adapter DTM is moved from a WirelessHART Communication Channel to a HART FSK Communication Channel, it:

- ~~has to~~ shall keep all instances of the Child DTMs untouched;
- ~~has to~~ shall allow moving Child DTMs away from its node.

7 Protocol-specific usage of general data types

Table 3 shows how general data types, defined in IEC 62453-2 within the namespace 'fdt', are used with HART devices.

Table 3 – Protocol specific usage of general data types

Data type	Description for use
fdt:address	The address property is not mandatory for the exposed parameters in the DTMs. But if the address property is used, the string shall be constructed according to the rules of the semanticId. That means the property 'semanticId' is always the same as the property 'address'
fdt:protocolId	See Clause 4
fdt:deviceTypeId	The property "fdt:DtmDeviceType.deviceTypeId" shall contain the DeviceTypeID of the supported physical device according to the ~~HCF~~ FieldComm Group's online product catalog
fdt:manufacturerId	Enter manufacturer according ~~HCF~~ FieldComm Group's list
fdt:semanticId fdt:applicationDomain	The applicationDomain attribute is: FDT_HART The sematicId for protocol related parameter is directly related to the protocol specification. The definition of the commands is the base for the semanticId. The semanticId for a parameter follows the following definition: CMDxxBy and CMD31EXTENDEDxxBy for extended HART 6 device family commands. The semanticIds for the Response Byte 0 and 1 defined in the IEC 61784 CPF 9 specification are: CMDxxRESPONSE_BYTE_0 CMDxxRESPONSE_BYTE_1 xx: represents the command number, getting the parameter via IEC 61784 CPF 9 protocol or the device family command number y: start byte within the command definition xx, y are based on decimal format without leading '0'
subDeviceType	Enter manufacturer specific value

8 Protocol-specific common data types

Not applicable.

9 Network management data types

9.1 General

The data types specified in 9.1 are used in the following services:

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

9.2 Addressing modes

The addressing mode depends on the type of the used HART protocol. Also additional addressing information might be necessary for some types of HART protocols. Table 4 shows the dependency of usable addressing modes and additional address information in dependency of the HART protocol in use.

Table 4 – Relation of ProtocolId and supported features

ProtocolId	Supported Addressing Modes	Address Data Type	Exposed Data	Comment
HART_Basic	ShortAddress	Used attributes: – shortAddress	As described in 9.3	This is defined for backward compatibility. New products should not use this bus category. Only single byte ManufacturerID and DeviceTypeID are supported.
HART_FSK HART_Wireless HART_RS485 HART_Infrared	ShortAddress, ShortTag, LongTag	Used attributes: – shortAddress – shortTag – longTag	As described in 9.4	A DTM may use more than one of these Ids if the device supports multiple physical connections e.g. WirelessHART and FSK.
HART_IP	ShortAddress, ShortTag, LongTag	Used attributes: – shortAddress – shortTag – longTag – ipAddress – port	As described in 9.4	

NOTE The 'HART_Basic' protocol is maintained for backward compatibility only~~, because support for one of the other protocols is required~~. In this document, the other protocols (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP) are referenced as 'Extended_HART' protocols (e.g. for definitions that apply to all protocols except 'HART_Basic'.)

9.3 Address information

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

9.4 Additional address information for 'Extended HART' protocols

DTMs that implement 'Extended_HART' protocols as required protocol ~~have to~~ shall provide address information as defined in 9.4 (in addition to address information according to 9.3). The information shall be provided as described in Table 5 and Table 6.

Support for all datatypes described in Table 5 and Table 6 is mandatory. If the data is not used (e.g. ipAddress), they have to shall be set to a type correct default value. The information in 9.4 is used for data interchange purpose between Parent DTM and Child DTM. All network information provided by a Child DTM may be changed by the Parent DTM (i.e. access is read-/write-able), except for hartVersion and pollingAddressRange which can be read only.

Table 5 – Simple address information data types

Data type	Definition	Description
activeProtocolId	enumeration (<Identifier values from Table 1>)	activeProtocolId is set by the Parent DTM to inform the child DTM what bus category is used (see Clause 4). If not used, set to identifier value that stands for HART.
addressingMode	enumeration (shortAddress | shortTag | longTag | longAddress)	Specifies the way how the communication will be established during the connect request.
hartVersion	INT	This value has to shall be set by the DTM itself to document the HART major version the device supports.
ipAddress	STRING	This value is set to the IP address used to connect to the device when using an IP based physical layer.
ipProtocolVersion	enumeration (IPv4 | IPv6)	This value specifies the version of the IP protocol which is used.
longAddressByte1	USINT	First byte of unique device identifier (long frame address). For HART 7: First byte of 16-bit Extended Device Type For HART 5 and 6: Composed from manufacturer id, master address bit and burst mode bit.
longAddressByte2	USINT	Second byte of unique device identifier (long frame address). For HART 7: First byte of 16-bit Extended Device Type For HART 5 and 6: 1 byte of device type code.
longAddressByte3	USINT	Third byte of unique device identifier (long frame address). First byte of unique device identifier.
longAddressByte4	USINT	Forth byte of unique device identifier (long frame address). Second byte of unique device identifier.
longAddressByte5	USINT	Fifth byte of unique device identifier (long frame address). Third byte of unique device identifier.
longTag	STRING	Value containing the long tag information that is used when connecting using addressingMode =longTag
networkID	INT	??? Stores the Network ID for a HART Wireless network (if applicable).
pollingAddressRange	enumeration ('0 to 15' | '0 to 63')	This value is set by the DTM itself to document the address range for the polling address of the device.
port	INT	This value is set to the port used to connect to the device when using an IP based physical layer. If not used, set to 0.
shortTag	STRING	Value containing the 8 character PACKED_ASCII tag that is used when connecting using addressingMode = shortTag

Table 6 – Structured address information data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
HartNetworkData	STRUCT			Data in this section is set by the DTM itself to provide general information
	hartVersion	M	[1..1]	
	pollingAddressRange	M	[1..1]	
HartDeviceAddress	STRUCT			Data in this section is communication relevant data that the parent DTM sets and that ~~has to~~ shall be ~~send~~ sent to the parent DTM back (e.g. with ConnectRequest)
	shortTag	M	[1..1]	
	longTag	M	[1..1]	
	addressingMode	M	[1..1]	
	ipAddress	M	[1..1]	
	port	M	[1..1]	
	ipProtocolVersion	M	[1..1]	
	networkID	M	[1..1]	
	longAddressByte1	M	[1..1]	
	longAddressByte2	M	[1..1]	
	longAddressByte3	M	[1..1]	
	longAddressByte4	M	[1..1]	
	longAddressByte5	M	[1..1]	
FdtHartExtension	STRUCT			Data in the root of FdtHartExtension is set by the parent DTM directly after attachment during topology management
	activeProtocolId	M	[1..1]	The initial value ~~has to~~ shall be 'HART_Basic'
	HartNetworkData	M	[1..1]	
	HartDeviceAddress	M	[1..1]	

10 Communication data types

10.1 General

The data types described in Clause 10 are used in the following services:

- connect service;
- disconnect service;
- transaction service.

The service arguments contain the address information and the communication data (explained in Table 7 and Table 8).

10.2 Protocol-specific Addressing Information

With the 'Extended_HART' protocols additional addressing information needs to be exchanged in order to establish a connection with the device. The additional addressing information is specific to the protocolId and version of the HART protocol.

For example HART 6 FSK device supports LongTag, and HART IP ~~has to~~ shall handle IP Address additionally.

During OnScanResponse(), the Communication Channel shall provide the additional addressing information for each device in the scan result, through the in-line schema extensions.

Frame Application can use this additional information to set the addressing information to the Child DTM through SetParameters().

~~For 1.2.1 Comm DTMs, Frame Application can use the new protocol specific xsl to transform this information to protocol independent document. The protocol specific xsl transforms the addressing information into a defined format and generates the IdAddress attribute in the protocol independent document. Frame Application uses this attribute in DTMDeviceListSchema to pass it to the parent DTM during IFdtChannelSubTopology2:SetChildrenAddresses(). The Parent DTM has to interpret this addressing information, to set the address information to the Child DTM using IDtmParameter:SetParameters().~~

10.3 Datatype definitions

To establish connection with a device, the DTM ~~has to~~ shall send protocol specific address information during ConnectRequest(). This information is used by the Communication Channel to address the device.

Child DTMs supporting an 'Extended_HART' protocol shall send additional addressing information as part of the ConnectRequest(), using the respective protocol-specific datatypes (see Table 7 and Table 8).

The Communication Channel supporting the 'Extended_HART' protocol can read the additional addressing information available in the ConnectRequest(), and use this information to address the device.

The data types described in 10.3 are defined for the following namespace.
Namespace: fdthart

Table 7 – Simple communication data types

Data type	Definition	Description
address1	USINT	Address information according to the IEC 61784 CPF 9 specification
address2	USINT	Address information according to the IEC 61784 CPF 9 specification
address3	USINT	Address information according to the IEC 61784 CPF 9 specification
addressingMode	enumeration (shortAddress | shortTag | longTag | longAddress)	Specifies which information will be used for creating the connection.
burstFrame	BOOL	Information whether the IEC 61784 CPF 9 response is a burst frame (message) or not
burstModeDetected	BOOL	Indicates whether the Communication Channel has detected that the device is already in burst mode. This is detected during a subscription request
commandNumber	USINT	Address information according to the IEC 61784 CPF 9 specification
communicationReference	UUID	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during the connect. The address information ~~has to~~ shall be used for all following communication calls
delayTime	UDINT	Minimum delay time in [ms] between two communication calls
deviceStatus	USINT	Status information. This is the second status byte returned in command responses according to the IEC 61784 CPF 9 specification
deviceTypeId	USINT	Address information according to the IEC 61784 CPF 9 specification
ipAddress	STRING	This value is set to the IP address used to connect to the device when using an IP based physical layer.
ipProtocolVersion	enumeration (IPv4 | IPv6)	This value specifies the version of the IP protocol which is used.
longFrameRequired	BOOL	Address information according to the IEC 61784 CPF 9 specification
longAddressByte1	USINT	First byte of unique device identifier (long frame address). Composed from manufacturer id, master address bit and burst mode bit.
longAddressByte2	USINT	Second byte of unique device identifier (long frame address). 1 byte of device type code.
longAddressByte3	USINT	Third byte of unique device identifier (long frame address). First byte of unique device identifier.
longAddressByte4	USINT	Forth byte of unique device identifier (long frame address). Second byte of unique device identifier.
longAddressByte5	USINT	Fifth byte of unique device identifier (long frame address). Third byte of unique device identifier.
longTag	STRING	Value containing the long tag information that is used when connecting using addressingMode =longTag
manufacturerId	USINT	Address information according to the IEC 61784 CPF 9 specification (Table: VIII, MANUFACTURER IDENTIFICATION CODES)
networkID	INT	~~???~~ Stores the Network ID for a HART Wireless network (if applicable).
port	INT	This value is set to the port used to connect to the device when using an IP based physical layer.
preambleCount	USINT	At the connect request the attribute is optional and contains a hint for the communication component about the number of preambles, required by the device type. At the connect response the attribute is mandatory and contains the information about the currently used preambleCount

Data type	Definition	Description
primaryMaster	BOOL	At the connect request the attribute is optional and contains a hint for a communication component that a DTM requires communication as primary or secondary master. At the connect response the attribute is mandatory and contains the information about the current state of the master
sequenceTime	UDINT	Period of time in [ms] for the whole sequence
shortAddress	USINT	Address information according to the IEC 61784 CPF 9 specification. This value is accessible via the attribute slaveAddress. SlaveAddress is part of the BusInformation structure. These values shall be set by the responsible component as described in clause Nested Communication of IEC 62453-2
shortTag	STRING	Value containing the 8 character PACKED_ASCII tag that is used when connecting using addressingMode =shortTag
value	USINT	Variable for status information
fdt:systemTag	STRING	System Tag of a DTM. It is strongly recommended to provide the attribute in the Request document.

Table 8 – Structured communication data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Abort	STRUCT			Describes the abort
	communicationReference	O	[0..1]	
CommandResponse	STRUCT			Status information. This is computed from the first status byte returned in command responses according to the IEC 61784 CPF 9 specification. If bit 7 of the first status byte is clear this value contains the value in the first status byte. If bit 7 is set this element is not returned in the status structure
	value	M	[1..1]	
CommunicationStatus	STRUCT			Status information. This is computed from the first status byte returned in command responses according to the IEC 61784 CPF 9 specification. If bit 7 of the first status byte is set this value contains the value in the first status byte (This is where we need to state whether it is the first status byte or bits 0-6 of the first status byte). If bit 7 is clear this element is not returned in the status structure
	value	M	[1..1]	
ConnectRequest	STRUCT			Describes the communication request for 'HART_Basic' protocol.
	fdt:tag	M	[1..1]	
	preambleCount	O	[0..1]	
	primaryMaster	O	[0..1]	
	longFrameRequired	O	[0..1]	
	fdt:systemTag	O	[0..1]	
	LongAddress	O	[0..1]	
	ShortAddress	M	[1..1]	
ExConnectRequest	STRUCT			Describes the communication request for 'Extended_HART' protocols.
	fdt:tag	M	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	shortAddress	~~M~~O	[1..1]	
	addressingMode	M	[1..1]	
	ipAddress	O	[0..1]	
	port	O	[0..1]	
	preambleCount	O	[0..1]	
	primaryMaster	O	[0..1]	
	fdt:systemTag	O	[0..1]	
	LongAddress	O	[0..1]	
ConnectResponse	STRUCT			Describes the communication response
	fdt:tag	M	[1..1]	
	preambleCount	M	[1..1]	
	primaryMaster	M	[1..1]	
	communicationReference	M	[1..1]	
	LongAddress	O	[0..1]	
	ShortAddress	~~M~~O	[1..1]	
DataExchange-Request	STRUCT			Describes the communication request
	commandNumber	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	O	[0..1]	
DataExchange-Response	STRUCT			Describes the communication response
	commandNumber	M	[1..1]	
	communicationReference	M	[1..1]	
	burstFrame	O	[0..1]	
	fdt:CommunicationData	O	[0..1]	
	Status	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response
	communicationReference	M	[1..1]	
SubscribeRequest	STRUCT			Describes the subscription request for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
SubscribeResponse	STRUCT			Describes the subscription response request for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
	burstModeDetected	M	[1..1]	
	fdt:communicationError	O	[0..1]	
UnsubscribeRequest	STRUCT			Describes the request to release the subscription for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
UnsubscribeResponse	STRUCT			Describes the response request to release the subscription for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
	fdt:communicationError	O	[0..1]	
LongAddress	STRUCT			Address information according to the IEC 61784 CPF 9 specification (only supported by devices based on HART revision > 5, see related documentation) In the IEC 61784 CPF 9 protocol Manufacturer ID and Device type ID are contained in the longaddress If the channel delivers different values in fdthart:manufacturerId / fdthart:deviceTypeId and in the corresponding bytes in fdthart:LongAddress, the following rule applies: * the fdthart:LongAddress ~~has to~~ shall be used for communication and * the fdthart:manufacturerId and fdthart:deviceTypeId may be used only as information about the manufacturer and the type of device
	~~manufacturerId~~longAddressByte1	M	[1..1]	
	~~deviceTypeId~~longAddressByte2	M	[1..1]	
	~~address1~~longAddressByte3	M	[1..1]	
	~~address2~~longAddressByte4	M	[1..1]	
	~~address3~~longAddressByte5	M	[1..1]	
SequenceBegin	STRUCT			Describes the sequence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	
SequenceEnd	STRUCT			Describes the sequence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Describes the sequence start
	communicationReference	M	[1..1]	
ShortAddress	STRUCT			Address information according to the IEC 61784 CPF 9 specification
	shortAddress	M	[1..1]	
Status	STRUCT			Status information according to the IEC 61784 CPF 9 specification
	deviceStatus	M	[1..1]	
	choice of	M	[1..1]	
	CommunicationStatus	S	[1..1]	
	CommandResponse	S	[1..1]	

The property ‘fdt:tag’, is part of the DtmDevice data type and contains the IEC 61784 CPF 9-specific value called TAG, which is used, for example within command #11, ‘READ UNIQUE IDENTIFIER ASSOCIATED WITH TAG’. This value shall be set by the responsible component as described in the Nested Communication publication IEC 62453-2.

11 Channel parameter data types

It is up to a DTM whether it provides any channels. If a DTM allows a Frame Application, other DTMs, or a controller to directly accessing the process values of its device via IEC 61784 CPF 9 protocol, then the DTM should provide FDT-Channel objects as described in Clause 11. Only the complete description of all channels belonging to a command allows proper access for external applications.

The description of channels, especially of the process values, allows the Frame Application to support the device in a more efficient way.

Used at ReadChannelData service and WriteChannelData service.

The information returned by the ReadChannelData service describes how to access an I/O value via a command (see Table 9 and Table 10).

The data types described in Clause 11 are defined for the following namespace.
Namespace: hartchannel

Table 9 – Simple channel parameter data types

Data type	Definition	Description
byteLength	USINT	Number of static bytes in a Request or in a Reply
commandNumber	UDINT	Number of the command containing the channel value
frameApplicationTag	STRING	Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces
gatewayBusCategory	UUID	Unique identifier for a supported bus type according to the FDT specific CATID
protectedByChannelAssignment	BOOL	TRUE if the channel is set to read only by the Frame Application. Usually set to TRUE if a channel assignment exists
value	STRING	Current value of a channel for read or write

Table 10 – Structured channel parameter data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
CommandParameters	STRUCT			Static command parameter bytes in a Request or in a Reply
	fdt:binData	O	[0..1]	
	byteLength	M	[1..1]	
FDTChannel	STRUCT			Description of the channel
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannel-Assignment	M	[1..1]	
	fdt:dataType	M	[1..1]	
	byteLength	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	appId:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	ReadCommand	O	[0..1]	
	WriteCommand	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
FDTChannelType	STRUCT			Description of the channel component in case of channels with gateway functionality
	fdt:VersionInformation	M	[1..1]	
	gatewayBusCategory	O	[0..1]	
ReadCommand	STRUCT			Description of the command to read the channel from a device
	commandNumber	M	[1..1]	
	Request	O	[0..1]	
	Reply	O	[0..1]	
	ResponseCodes	O	[0..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Reply	STRUCT			Description of the reply structure of a command according to the IEC 61784 CPF 9 specification
	collection of	M	[1..1]	
	fdt:ChannelReference		[0..*]	
	CommandParameters		[0..*]	
	ResponseCodes	O	[0..1]	
Request	STRUCT			Description of the request structure of a command according to the IEC 61784 CPF 9 specification
	collection of	M	[1..1]	
	fdt:ChannelReference		[0..*]	
	CommandParameters		[0..*]	
ResponseCodes	STRUCT			Collection of specific response codes according to the IEC 61784 CPF 9 specification (known as COMMAND-SPECIFIC RESPONSE CODES)
	fdt:EnumeratorEntry	M	[1..*]	
WriteCommand	STRUCT			Description of the command to write the channel to a device
	commandNumber	M	[1..1]	
	Request	O	[0..1]	
	Reply	O	[0..1]	
	ResponseCodes	O	[0..1]	

12 Device identification

12.1 Protocol-specific handling of data type STRING

IEC 61784 CPF 9 char array rules:

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

12.2 Address Range for Scan

The Frame Application can specify the bus address range to the Communication Channel for scanning. The supported scan range is specific to the protocol. Table 11 describes how the BusAddressRange and ScanMode attributes can be used for different HART protocols.

Table 11 – Address range for device identification

Protocol	Comments
HART	The Frame Application can specify the short address range for scan.
HART_FSK	Only if the addressingMode of the Communication DTM is shortAddress, this is applicable.
HART_Wireless	The Frame Application can use the ScanMode, to specify to scan all addresses or to request the Communication DTM to open GUI
HART_RS485	Only if the addressingMode of the Communication DTM is shortAddress, this is applicable.
HART_Infrared	The Frame Application can use the ScanMode, to specify to scan all addresses or to request the Communication DTM to open GUI
HART_IP	The Frame Application can use the ScanMode, to specify to scan all addresses or to request the Communication DTM to open GUI to specify select IP addresses.

12.3 Support for Extended Manufacturer and Device Type Code

HART 7 devices support extended manufacturer id and device type codes. With all 'Extended_HART' protocols, the extended manufacturer id and device type codes are supported.

Parent DTMs supporting the 'Extended_HART' protocols shall use the in-line schema extension during the OnScanResponse(). Through the in-line schema the Parent DTM can provide additional information, e.g. extended manufacturer id, device type code, Device Id, HART long tag and short tag.

A Frame Application can use this additional information for assigning the DTM based on the extended manufacturer id and device type code. The extended FDT 1.2 scan result document format is specified in Table 25.

12.4 Device type identification data types for protocol 'HART_Basic'

The data types described in 12.4 are reused as defined by 12.6 and 12.9.

The IEC 61784 CPF 9 device type identification data types provide general data types with a protocol-specific semantic (see Table 12 and Table 13) as well as data types without such a mapping (see Table 14 and Table 15).

The data types described in 12.4 are defined for following namespace.
Namespace: hartident

Table 12 – Identification data types with protocol-specific mapping for protocol 'HART_Basic'

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
shortAddress	IdAddress	Poll possible address range (HART5: [0-15], HART 6: [0-63]) by calling Cmd #0. If Cmd #0 response is available, a physical device is connected to this address. Cmd #0 response does not contain short address value whether the short or long format is used. If master using short address for polling receives a response, it can assume that short address of device is the same as used in the polling request. In addition to this, polling address can be read from HART 6 device with cmd #7	Polling Address	Unsigned 8	USINT	See [3] Chapter 6.8 Command 7 Read Loop Configuration
busProtocol	IdBusProtocol	CommChannel ~~has to~~ shall pass "HART" in this attribute	HART	Enumeration: "HART" for HART5 and HART6	enumeration (HART)	
universalCommandRevisionLevel	IdBusProtocolVersion	Command #0 Byte 4	HART Revision	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
manufacturerIdentificationCode	IdManufacturer	~~Command 0 Byte 1 –~~ ~~HART6: Manufacturer Identification Code~~ ~~HART 5: Manufacturer Device Type Code~~ HART7: Command #0 Byte 17+18. HART<7: Command #0 Byte 1.	Manufacturer Identification Code	8 bit unsigned integer Example: Endress+Hauser: 17 (0x11)	~~USINT~~UINT (dec)	See [4] Chapter 5.8 Manufacturer Identification Codes
deviceTypeID	IdTypeID	~~Command 0 Byte 2 – Manufacturers Device Type code~~ HART 7: Command #0 Byte 1+2. HART<7: Command #0 Byte 2.	Device Type Code	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
softwareRevision	IdSoftwareRevision	Command #0 Byte 6	Software Revision	8 bit unsigned integer	USINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
hardwareRevision	IdHardwareRevision	Command #0 Byte 7 (Most significant 5 bits).	Hardware Revision	~~8~~5 bit unsigned integer ~~(mapped to float: xxxxx.yyy)~~ ~~First 5 bits (x) refers to HW revision level.~~ ~~Last 3 bits (y) to Physical Signaling Code~~	~~REAL~~INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
physicalSignalingCode	-	Command #0 Byte 7 (Least significant 3 bits).	Physical Signaling Code	3 bit unsigned enumerator	INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
tag	IdTag	Command #13 Bytes 0 – 5	Tag	6 Bytes or Packed ASCII characters	STRING	See [3] Chapter 6.13 Command 13 Read Tag, Descriptor, Date
deviceID	IdSerialNumber	Command #0 Bytes 9 – 11	Device Identification Number	Unsigned 24	UDINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
N/A	IdDTMSupportLevel	Not applicable for scan / physical device. Attribute to be used only in DTMDeviceType identification. Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	DTM Support Level	-	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	-

Table 13 – Identification data types with semantics for protocol 'HART_Basic'

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	XML-FDT format (display format)	Specification reference
deviceCommandRevisionLevel	-	Command #0 Byte 5	Device Revision Level	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
deviceFlag	-	Command #0 Byte 8	Flags	Bit value according Flag Assignment table. 8 bit – unsigned int	USINT (hex)	See [4] Chapter 5.11 Table Flag Assignments
manufacturerSpecificExtension		Can be used by DTM for vendor specific device identification information, for example by combining a number of device parameter values into one string value. This can be used to identify a specific device variant			STRING	

Table 14 – Simple identification data types for protocol 'HART_Basic' with protocol independent semantics

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	enumeration genericSupport profileSupport blockspecificProfileSupport specificSupport
match	STRING	Used by Device DTM to define a regular expression which shall match to scanned physical define identification information
nomatch	STRING	Used by Device DTM to define a regular expression which shall not match to scanned physical define identification information. Used by Device DTM to indicate if identification information may not match

Table 15 – Structured identification data types for protocol 'HART_Basic' with protocol independent semantics

Elements	Definition			Description
	Elementary data types	Usage	Multiplicity	
RegExpr	STRUCT			Includes regular expression string – either for match or for nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.5 Common device type identification data types for 'Extended_HART' protocols

The data types described in 12.5 are reused as defined by 12.6 and 12.9.

The IEC 61784 CPF 9 device type identification data types provide general data types with a protocol-specific semantic (see Table 16 and Table 17) as well as data types without such a mapping (see Table 18 and Table 19).

The data types described in 12.5 are defined for following namespace.
Namespace: hartident2

Table 16 – Identification data types for ‘Extended_HART’ protocols with protocol-specific mapping

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
shortAddress	IdAddress	Poll possible address range (HART5: [0-15], HART >=6: [0-63]) by calling Cmd #0. If Cmd #0 response is available, a physical device is connected to this address. Cmd #0 response does not contain short address value whether the short or long format is used. If master using short address for polling receives a response, it can assume that short address of device is the same as used in the polling request. In addition to this, polling address can be read from HART 6 device with cmd #7	Polling Address	Unsigned 8	USINT	See [3] Chapter 6.8 Command 7 Read Loop Configuration
busProtocol	IdBusProtocol	Communication Channel ~~has to~~ shall pass ”HART” in this attribute	HART	Enumeration: “HART” for HART5 and HART6	enumeration (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP)	
universalCommandRevisionLevel	IdBusProtocolVersion	Command #0 Byte 4	HART Revision	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
manufacturerIdentificationCode	IdManufacturer	For HART 7: 16 Bit Command #0 Byte 17+18 – Manufacturer Identification Code For HART 5 and 6: 8 bit Command #0 Byte 1- Manufacturer Identification Code	Manufacturer Identification Code	16 bit unsigned integer Example: Endress+Hauser: 17 (0x11)	UINT (dec)	See [3] Chapter 5.8 Manufacturer Identification Codes See [4] Manufacturer Identification Codes
deviceTypeID	IdTypeID	For HART 7: 16 Bit Command #0 Byte 1+2 – Manufacturer Identification Code For HART 5 and 6: 8 bit Command #0 Byte 2 – Manufacturers Device Type code	Device Type Code	16 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
softwareRevision	IdSoftwareRevision	Command #0 Byte 6	Software Revision	8 bit unsigned integer	USINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
hardwareRevision	IdHardware Revision	Command #0 Byte 7 (Most significant 5 bits)	Hardware Revision	~~8~~5 bit unsigned integer ~~(mapped to float: xxxxx.yyy)~~ ~~First 5 bits (x) refers to HW revision level.~~ ~~Last 3 bits (y) to Physical Signaling Code~~	~~REAL~~INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
physicalSignalingCode	-	Command #0 Byte 7 (Least significant 3 bits).	Physical Signaling Code	3 bit unsigned enumerator	INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
tag	IdTag	For HART >=6: Command #20 For HART <6: Command #13 Bytes 0 – 5	Tag	6 Bytes ~~or~~ of 8 Packed ASCII characters	STRING	See [4] Chapter 6.20 – Command 20 Read Long Tag See [3] Chapter 6.13 Command 13 Read Tag, Descriptor, Date
longTag		For HART >=6: Command #20 For HART <6: Command #12 Bytes 0 – 23		32 Bytes char or 24 Bytes or 32 Packed ASCII characters	STRING	See [4] Chapter 6.20 – Command 20 Read Long Tag See [4] Chapter 6.12 – Command 12 Read Message
shortTag		Command #13 Bytes 0 – 5.		6 Bytes ~~or~~ of 8 Packed ASCII characters	STRING	See [4] Chapter 6.13 – Command 13 Read Tag, Descriptor, Date
deviceID	IdSerialNumber	Command #0 Bytes 9 – 11	Device Identification Number	Unsigned 24	UDINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
ipAddress		host name or IP address conformant to IPv4 or IPv6 standard (additionally including the port number, if required) of a HART UDP or HART TCP device.			STRING	

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
port		port of a HART TCP or UDP device		64 bit unsigned	USINT (dec)	
N/A	IdDTMSupportLevel	Not applicable for scan / physical device. Attribute to be used only in DTMDeviceType identification. Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	DTM Support Level	-	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	-

Table 17 – Identification data types for ‘Extended_HART’ protocols without protocol independent semantics

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	XML-FDT format (display format)	Specification reference
deviceCommandRevisionLevel	-	Command #0 Byte 5	Device Revision Level	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
deviceFlag	-	Command #0 Byte 8	Flags	Bit value according Flag Assignment table. 8 bit – unsigned int	USINT (hex)	See [4] Chapter 5.11 Table Flag Assignments
manufacturerSpecificExtension		Can be used by DTM for vendor specific device identification information, for example by combining a number of device parameter values into one string value. This can be used to identify a specific device variant.			STRING	

Table 18 – Simple identification data types for 'Extended_HART' protocols with protocol independent semantics

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	enumeration genericSupport profileSupport blockspecificProfileSupport specificSupport
match	STRING	Used by Device DTM to define a regular expression which shall match to scanned physical define identification information
nomatch	STRING	Used by Device DTM to define a regular expression which shall not match to scanned physical define identification information. Used by Device DTM to indicate if identification information may not match
schemaVersion	STRING	Version number that is used by a Frame Application to identify an updated schema. The value for schemas redefined with this document ~~has to~~ shall be set to "1.3"
addressingMode	enumeration (shortAddress | shortTag | longTag)	With this attribute the Parent DTM defines which address property shall be used for the connection

Table 19 – Structured identification data types for 'Extended_HART' protocols with protocol independent semantics

Elements	Definition			Description
	Elementary data types	Usage	Multiplicity	
RegExpr	STRUCT			Includes regular expression string – either for match or for nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.6 Topology scan data types

This data type is used at Scan service response.

The data types describe one entry in the list of scanned devices (see Table 20).

The data types described in 12.6 are defined for the following namespace.
Namespace: fdthartdevice

Table 20 – Structured device type identification data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
HARTDevice	STRUCT			Definition of a IEC 61784 CPF 9 device concerning the scan response
	fdthart:LongAddress	O	[0..1]	
	fdthart:manufacturerId	O	[0..1]	
	fdthart:deviceTypeId	O	[0..1]	
	fdt:subDeviceType	O	[0..1]	
	fdt:tag	M	[1..1]	
	fdthart:shortAddress	O	[0..1]	

12.7 Scan identification data types for protocol 'HART_Basic'

Subclause 12.7 defines data types that are used to provide protocol-specific scanning (see Table 21 and Table 22).

The data types described in 12.7 are used at following services: scan service.

The data types described in 12.7 are defined for the following namespace.
Namespace: hartscan

Table 21 – Simple scan identification data types for protocol 'HART_Basic'

Data type	Definition	Description
resultState	enumeration (provisional | final | error)	Identifies if the result is one of the provisional results or the final result of the split scan results
configuredState	enumeration (configuredAndPhysicallyAvailable | | configuredAndNotPhysicallyAvailable | availableButNotConfigured | notApplicable)	A communication master shall indicate in this attribute, if the scan response is related to a detected physical device which is configured or unconfigured

Table 22 – Structured scan identification data types for protocol 'HART_Basic'

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdAddress	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix "Id" for better identification
	hartident:shortAddress	M	[1..1]	
IdBusProtocol	STRUCT			
	hartident:busProtocol	M	[1..1]	
IdBusProtocolVersion	STRUCT			
	hartident:universalCommandRevisionLevel	M	[1..1]	
IdManufacturer	STRUCT			
	hartident:manufacturerIdentificationCode	M	[1..1]	
IdTypeID	STRUCT			
	hartident:deviceTypeID	M	[1..1]	
IdSoftwareRevision	STRUCT			
	hartident:softwareRevision	M	[1..1]	
IdHardwareRevision	STRUCT			
	hartident:hardwareRevision	M	[1..1]	
IdTag	STRUCT			
	hartident:tag	M	[1..1]	
IdSerialNumber	STRUCT			
	hartident:deviceID	M	[1..1]	
DeviceCommandRevision Level	STRUCT			All elements without semantic prefix "Id" are transformed by XSL to name value pairs. These elements contain exactly one attribute defined in IEC 62453-2, each including one value of the physical device
	hartident:deviceCommandRevisionLevel	M	[1..1]	
DeviceFlag	STRUCT			
	hartident:deviceFlag	M	[1..1]	
ManufacturerSpecific-Extension	STRUCT			
	hartident:manufacturerSpecificExtension	M	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ScanIdentification	STRUCT			All IEC 61784 CPF 9 scan identification elements for one scanned physical device
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdTag	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	DeviceCommandRevisionLevel	M	[1..1]	
	DeviceFlag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentifications	STRUCT			Collection of ScanIdentification elements
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	

12.8 Scan identification data types for 'Extended_HART' protocols

Subclause 12.8 defines data types that are used to provide protocol-specific scanning (see Table 23 and Table 24).

The data types described in 12.8 are used at following services: scan service.

The data types described in 12.8 are defined for the following namespace.
Namespace: hartscan2

Table 23 – Simple scan identification data types for 'Extended_HART' protocols

Data type	Definition	Description
resultState	enumeration (provisional | final | error)	Identifies if the result is one of the provisional results or the final result of the split scan results
configuredState	enumeration (configuredAndPhysicallyAvailable | | configuredAndNotPhysicallyAvailable | availableButNotConfigured | notApplicable)	A communication master shall indicate in this attribute, if the scan response is related to a detected physical device which is configured or unconfigured

Table 24 – Structured scan identification data types for ‘Extended_HART’ protocols

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdAddress	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix “Id” for better identification
	hartident:shortAddress	M	[1..1]	
	hartident:shortTag	M	[1..1]	
	hartident:longTag	M	[1..1]	
	hartident:addressingMode	M	[1..1]	
	hartident:ipAddress	O	[1..1]	
	hartident:port	O	[1..1]	
	hartident:ipVersion	O	[1..1]	
	hartident:networkId	O	[1..1]	
	hartident:longAddressByte1	M	[1..1]	
	hartident:longAddressByte2	M	[1..1]	
	hartident:longAddressByte3	M	[1..1]	
	hartident:longAddressByte4	M	[1..1]	
	hartident:longAddressByte5	M	[1..1]	
IdBusProtocol	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix “Id” for better identification
	hartident:busProtocol	M	[1..1]	
IdBusProtocolVersion	STRUCT			
	hartident:universalCommandRevisionLevel	M	[1..1]	
IdManufacturer	STRUCT			
	hartident:manufacturerIdentificationCode	M	[1..1]	
IdTypeID	STRUCT			
	hartident:deviceTypeID	M	[1..1]	
IdSoftwareRevision	STRUCT			
	hartident:softwareRevision	M	[1..1]	
IdHardwareRevision	STRUCT			
	hartident:hardwareRevision	M	[1..1]	
IdTag	STRUCT			
	hartident:tag	M	[1..1]	
IdSerialNumber	STRUCT			
	hartident:deviceID	M	[1..1]	
DeviceCommandRevision Level	STRUCT			All elements without semantic prefix “Id” are transformed by XSL to name value pairs. These elements contain exactly one attribute defined in IEC 62453-2, each including one value of the physical device
	hartident:deviceCommandRevisionLevel	M	[1..1]	
DeviceFlag	STRUCT			
	hartident:deviceFlag	M	[1..1]	
ManufacturerSpecific-Extension	STRUCT			
	hartident:manufacturerSpecificExtension	M	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ScanIdentification	STRUCT			All IEC 61784 CPF 9 scan identification elements for one scanned physical device
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdTag	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	DeviceCommandRevisionLevel	M	[1..1]	
	DeviceFlag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentifications	STRUCT			Collection of ScanIdentificati on elements
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	

12.9 Device type identification data types – provided by DTM

Subclause 12.9 defines data types that are used to provide protocol-specific information for device types (see Table 25).

The data types described in 12.9 are used in the following services:

- GetIdentificationInformation service.

The data types described in 12.9 are defined for the following namespace.
Namespace: hartdevtype

Table 25 – Structured device type identification data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdBusProtocol	STRUCT			All elements contain exactly one attribute, each including the value of the scanned physical device. All elements with semantic meaning have a prefix "Id" for better identification
	hartident:busProtocol	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdBusProtocolVersion	STRUCT			
	hartident:universalCommandRevisionLevel	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdManufacturer	STRUCT			
	hartident:manufacturerIdentificationCode	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdTypeID	STRUCT			
	hartident:deviceTypeID	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdSoftwareRevision	STRUCT			
	hartident:softwareRevision	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdHardwareRevision	STRUCT			
	hartident:hardwareRevision	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
DeviceCommand-RevisionLevel	STRUCT			All elements without semantic prefix "Id" are transformed by XSL to name value pairs. These elements contain exactly one attribute defined in IEC 62453-2 each including one value of the physical device
	hartident:deviceCommandRevisionLevel	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
DeviceFlag	STRUCT			
	hartident:deviceFlag	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
ManufacturerSpecific-Extension	STRUCT			
	hartident:manufacturerSpecificExtension	M	[1..1]	
DeviceIdentification	STRUCT			IEC 61784 CPF 9 specific identification elements
	hartident:idDTMSupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	DeviceCommandRevisionLevel	M	[1..1]	
	DeviceFlag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..*]	
DeviceIdentifications	STRUCT			List of device identifications
	fdt:protocolId	M	[1..1]	
	DeviceIdentification	M	[1..*]	

Bibliography

[1] HCF_SPEC-99 Revision 9.0, *HCF Command Summary Specification*, July 2007

[2] HCF_SPEC-085 Revision 2.0, *HCF Network Management Specification*, June 2012

[3] HCF SPEC-127 Revision 7.1, *Universal Command Specification*, May 2008

[4] HCF SPEC-183 Revision 16.0, *Common Tables Specification*, 19 December 2005

[5] IEC TR 62453-42, *Field device tool (FDT) interface specification – Part 42: Object model integration profile – Common Language Infrastructure*

[6] ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*

____

IEC

IEC 62453-309

Edition 3.0 2022-09

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Field device tool (FDT) interface specification – Part 309: Communication profile integration – IEC 61784 CPF 9

Spécification des interfaces des outils des dispositifs de terrain (FDT) – Partie 309: Intégration des profils de communication – CPF 9 de l'IEC 61784

IEC 62453-309:2022-09(en-fr)

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 309: Communication profile integration – IEC 61784 CPF 9

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 62453-309 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

This third edition cancels and replaces the second edition published in 2016. This edition constitutes a technical revision.

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

- corrections in regard to accessing information in the respective device and
- corrections in regard to describing support for different protocol versions.

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

Draft	Report on voting
65E/907/FDIS	65E/936/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

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

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

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

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

INTRODUCTION

This part of IEC 62453 is an interface specification for developers of FDT[1] (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

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

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

Figure 1 shows how IEC 62453-309 is aligned in the structure of the IEC 62453 series.

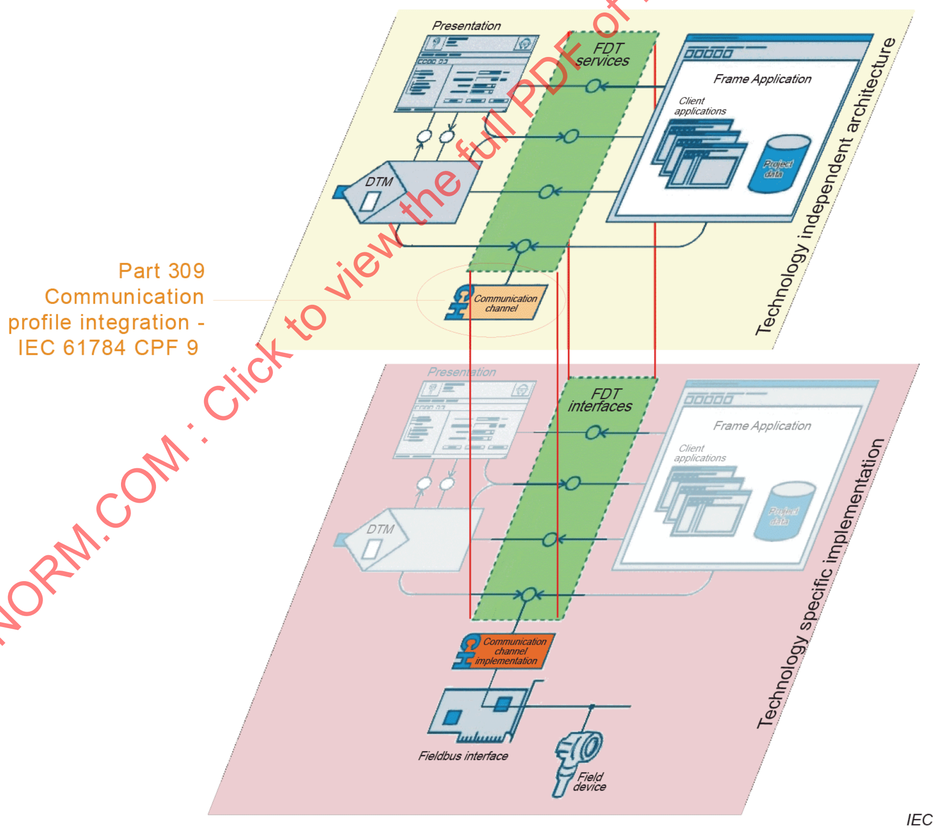


Figure 1 – Part 309 of the IEC 62453 series

1 FDT® is a trademark of products supplied by FDT Group AISBL. This information is given for convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 309: Communication profile integration – IEC 61784 CPF 9

1 Scope

Communication Profile Family 9 (commonly known as HART®[2]) defines communication profiles based on IEC 61158-5-20 and IEC 61158-6-20. The basic profile CP 9/1 is defined in IEC 61784-1.

This part of IEC 62453 provides information for integrating the HART® technology into the FDT standard (IEC 62453-2).

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

This document neither contains the FDT specification nor modifies it.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61158-5-20, *Industrial communication networks – Fieldbus specifications – Part 5-20: Application layer service definition – Type 20 elements*

IEC 61158-6-20, *Industrial communication networks – Fieldbus specifications – Part 6-20: Application layer protocol specification – Type 20 elements*

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

IEC 62453-1:–[3], *Field device tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:–[3], *Field device tool (FDT) interface specification – Part 2: Concepts and detailed description*

[2] HART® and WirelessHART® are trade names of products supplied by FieldComm Group. This information is given for convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

[3] Under preparation. Respective stage at the time of publication: IEC/CCDV 62453-1:2022 and IEC/RFDIS 62453-2:2022.

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http:www.iso.org/obp

3.1.1
burst mode
mode in which the field device generates response telegrams without request telegram from the master

3.2 Abbreviated terms

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

BACK	Burst ACKnowledge
C8PSK	Coherent 8-way Phase Shift Keying, HART communication layer as defined in HCF_SPEC-60, Revision 1.0
DR	delayed response
EDD	Electronic Device Description
FSK	Frequency Shift Keying, HART communication layer as defined in HCF_SPEC-54, Revision 8.1
HART	Highway Addressable Remote Transducer

3.3 Conventions

3.3.1 Data type names and references to data types

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

3.3.2 Vocabulary for requirements

The following expressions are used when specifying requirements:

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

3.3.3 Use of UML

Figures in this document are using UML notation as defined in IEC 62453-1:–, Annex A.

4 Bus category

IEC 61784 CPF 9 protocol is identified in the protocolId element of structured data type 'fdt:BusCategory' by the following unique identifiers (see Table 1):

Table 1 – Protocol identifiers

Identifier value	ProtocolId	Display String	Description
036D1498-387B-11D4-86E1-00E0987270B9	HART_Basic	'HART'	Support of IEC 61784 CPF 9 protocol over FSK communication with basic functionality (deprecated)
98503B8F-0FFB-4EB7-BB67-F4D6BD16DB8D	HART_FSK	'HART FSK'	Support of HART protocol over FSK communication with complete functionality
74D29D22-F752-40EF-A747-ACA72C791155	HART_Wireless	'HART Wireless'	Support of WirelessHART protocol
58001A08-C178-4A59-A76B-9EF9111CB83D	HART_RS485	'HART RS485'	Support of HART protocol over RS485 communication
EF708CB7-A2A1-42AF-890C-15CEB680CC12	HART_Infrared	'HART Infrared'	Support of HART protocol over Infrared communication
D122D172-F0C7-4B03-965B-512CD4C0871E	HART_IP	'HART IP'	Support of HART over IP protocol

The 'HART_Basic' protocol is maintained for backward compatibility only (e.g. for interaction with DTMs according to IEC 62453-309:2009). The other protocol identifiers provide a better support for planning of network topologies and for establishment of connections between DTM and respective device. For DTMs complying with this document, support for one of the other protocols is mandatory.

Within this document, the other protocols (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP) are referenced as 'Extended_HART' protocols. (E.g. for definitions that apply to all protocols except 'HART_Basic'.)

Table 2 defines which PhysicalLayer can be used together with the BusCategory defined in Table 1.

Table 2 – Definition of PhysicalLayer

PhysicalLayer Id value	PhysicalLayer name value	Description
BAB2091A-C0A7-4614-B9DE-FCC2709DCF5D	HART FSK Physical Layer	Support of HART FSK physical layer
B9F1A250-AC94-4487-8F25-A8F3F8F89DC5	WirelessHART Physical Layer	Support of WirelessHART physical layer
036D1591-387B-11D4-86E1-00E0987270B9	HART RS-485 Physical Layer	Support of HART devices using RS-485 communication
AE4119EF-B9FD-429c-B244-134DB182296A	HART Infrared Physical Layer	Support of HART devices using infrared communication
307dd808-c010-11db-90e7-0002b3ecdcbe	10BASET	HART Ethernet based Physical Layers
307dd809-c010-11db-90e7-0002b3ecdcbe	10BASETXHD	
307dd80a-c010-11db-90e7-0002b3ecdcbe	10BASETXFD	
307dd80b-c010-11db-90e7-0002b3ecdcbe	10BASEFLHD	
307dd80c-c010-11db-90e7-0002b3ecdcbe	10BASEFLFD	
307dd80d-c010-11db-90e7-0002b3ecdcbe	10BASEFXHD	
307dd80e-c010-11db-90e7-0002b3ecdcbe	10BASEFXFD	

PhysicalLayer Id value	PhysicalLayer name value	Description
307dd80f-c010-11db-90e7-0002b3ecdcbe	100BASETXHD	
307dd810-c010-11db-90e7-0002b3ecdcbe	100BASETXFD	
307dd811-c010-11db-90e7-0002b3ecdcbe	100BASEFXHD	
307dd812-c010-11db-90e7-0002b3ecdcbe	100BASEFXFD	
307dd813-c010-11db-90e7-0002b3ecdcbe	100BASELX10	
307dd814-c010-11db-90e7-0002b3ecdcbe	100BASEPX10	
307dd815-c010-11db-90e7-0002b3ecdcbe	1000BASEXHD	
307dd816-c010-11db-90e7-0002b3ecdcbe	1000BASEXFD	
307dd817-c010-11db-90e7-0002b3ecdcbe	1000BASELXHD	
307dd818-c010-11db-90e7-0002b3ecdcbe	1000BASELXFD	
307dd819-c010-11db-90e7-0002b3ecdcbe	1000BASESXHD	
307dd81a-c010-11db-90e7-0002b3ecdcbe	1000BASESXFD	
307dd81b-c010-11db-90e7-0002b3ecdcbe	1000BASETHD	
307dd81c-c010-11db-90e7-0002b3ecdcbe	1000BASETFD	
307dd81d-c010-11db-90e7-0002b3ecdcbe	10GigBASEFX	

The significant information for topology planning is the BusCategory. The PhysicalLayer (which is provided in the BusInformation data type) shall be used only for additional information.

The DataLinkLayer property is not applicable for HART and shall be set to null.

5 Access to instance and device data

5.1 General

The HART protocol has semantics defined that allow in a wide range the identification of device variables and device parameters. Most of this semantic information is defined in the standard EDD import libraries.

Clause 5 describes how the semantic information defined with the HART protocol shall be used to export device data, instance data and process data.

5.2 Process Channel objects provided by DTM

The minimum set of provided data shall be the first four provided process related values (PV, SV, …) – if available – modeled as channel references. The referenced channel shall include ranges and scaling.

A HART device communicates the process data either via its analogue channels or via digital information (e.g. by request or by burst mode). Analogue channels are always related to a dynamic variable, as specified in [1][4] chapter 8 and therefore the description of an analogue channel shall be accessed using the respective dynamic variable (e.g. the attributes of dynamic variable PV always describe the first analogue channel).

4 Figures in square brackets refer to the bibliography.

HART distinguishes between three methods to access digital signals:

1) Access to analogue value and assigned dynamic variables (Command #3)

   IO signals can be assigned to one of the four dynamic variables PV, SV, TV, and QV. Using the command #3 the analogue value and the dynamic variables can be read without specific device knowledge.

2) Indexed access to device variables (Command #33)

   All device variable values and their units can be read using the related device variable code information in command #33.

3) Indexed access to device variable classification and device variable status (Command #9)

   Command #9 provides more information than command #33. Beside of the value and unit also a classification and the variable status can be determined.

The command initiator determines by means of the HART specification which commands will be used.

5.3 DTM services to access instance and device data

The services InstanceDataInformation and DeviceDataInformation shall provide access to at least all parameters of the Universal and Common Practice commands (as far as the device supports the function).

Furthermore, the Response Byte 0 and the Response Byte 1 for each command shall be exposed.

The services InstanceDataInformation and DeviceDataInformation may also provide access to device specific parameters (e.g. diagnostic information).

6 Protocol-specific behavior

6.1 Overview

There is only one protocol-specific sequence defined for IEC 61784 CPF 9: burst mode subscription.

This sequence explains how the sequence “Device initiated data transfer”, defined in IEC 62453-2, is applied in context of burst telegrams as defined by IEC 61784 CPF 9.

Additionally, Clause 6 provides information regarding:

- usage of device addressing information,
- support of extended command codes,
- handling of communication failures,
- handling of delayed responses, and
- management of physical topologies.

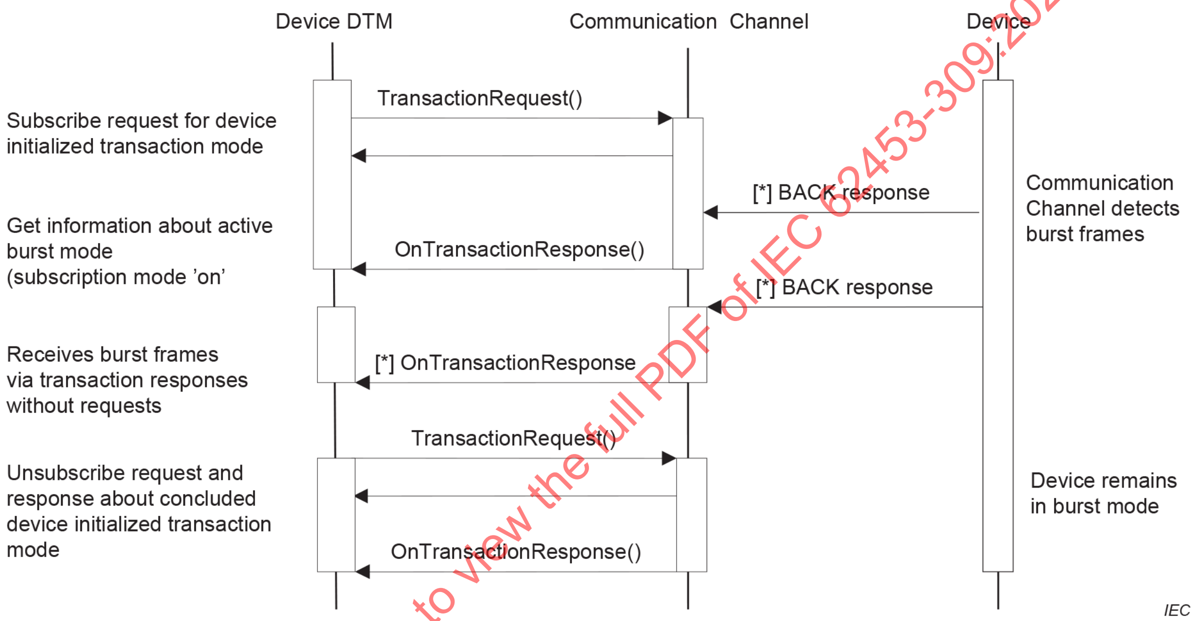
6.2 Burst mode subscription

A subscription to device-initiated data transfer can be requested by sending a transaction request with SubscribeRequest content (see Figure 2). The Communication Channel may detect if the device is already in burst mode.

NOTE In HART 5 this can be detected only when burst frames are received from the device. In HART 6 the burst mode can be detected using command #105.

The Communication Channel answers to a SubscribeRequest with a SubscribeResponse content. If burst frames are received, the device is in burst mode and burstModeDetected value is set to TRUE. This means that Device DTM will start to receive burst messages via the transaction response mechanism. In the case that no burst messages were received, burstModeDetected value is set to FALSE. It is up to Device DTM to set device into burst mode. Then Device DTM may call a transaction request with SubscribeRequest content again in order to receive burst messages.

In order to unsubscribe, the Device DTM sends a transaction request with a UnsubcribeRequest. The Communication Channel answers with a UnsubscribeResponse where burstModeDetected value is set to FALSE. The Device DTM will not receive any more burst information via the transaction response mechanism. The Communication Channel does not switch off the burst mode in the device. The Device DTM may switch burst mode on or off by using normal transaction requests (command #109). This is independent of the subscription.



NOTE BACK means Burst ACKnowledge.

Figure 2 – Burst mode subscription

6.3 Usage of device addressing information

HART is a connectionless master/slave protocol. Transaction requests are always addressed using unique device address information (a 5 byte integer), the so called long address.

Device addressing in HART therefore is mainly focused to determine this long address.

There are currently three ways possible to determine the long address.

1) short address

   The short address is a number between 0 and 63 (for HART version 5 only 0 to 15). In the context of a direct connection to the device the short address is unique and allows to read the long address using command #0.

2) short tag

   With command #11 the long address information can be requested for a device with a specific short tag. Such requests are especially used for installations with a huge amount of connected HART devices. All HART multiplexer devices and other HART communication structures shall support this command.

3) long tag

   HART Version 6 introduced the long tag, which at 32 characters has 24 more characters than the short tag, and can thus provide for a larger number of device label options. For devices with HART version less than 6 instead of long tag, message is used. With command #21 the long address information can be requested for a device with a specific long tag. Command #21 is usually supported by highly modular devices or gateways.

A Device DTM is responsible to provide and store all information that is used for resolving the long address of a connected device. It has therefore to maintain the data for all three address resolving methods. The DTM responsible to connect to the communication hardware shall select the method and provide means for a user to input the address information.

Besides the addressing topic, there are also different approaches for manufacturer and device type identification depending on the supported version of HART. HART versions up to HART 6 use one byte values. HART versions starting from HART 7 (and newer) use a two byte value. The two byte values are also stored in the data types described in 9.4.

A Communication DTM uses the addressing information provided by the Device DTM in order to resolve the long address as described above.

6.4 Extended Command Numbers

The HART command number is defined as a one byte unsigned integer. Beginning with HART Version 6, an extended command number format, using two bytes instead of one, was defined to allow for more than 255 command numbers. This format uses command number 31, which was previously reserved, to indicate that the request is using the extended command number format.

According to the specification in [2] 8.1.2, extended commands are implemented with command #31 by using the extended command number as first two bytes in the request and response section.

In FDT, all commands with extended command numbers shall be implemented using command #31.

6.5 Handling of communication failures and time-outs

HART uses a device-specific handling of communication errors. The protocol defines a section in the response frame that can carry communication failure information.

If, during execution of a communication request to a Communication Channel, a communication error occurs on the HART physical layers (this also includes time-outs), no Abort message shall be sent to the Child DTM, but the transaction request shall be responded with a set of data that describes the communication error as defined in HART [1].

In case of such a communication failure, the Device DTM has the responsibility to perform the error handling to recover from the communication failure.

Only in case of a connection-based communication break (e.g. Ethernet connection to a HART modem), the Communication Channel shall send an Abort signal to the device DTM.

6.6 Handling of delayed responses

HART defines strict time constraints for responses to a request within a HART transaction. In case a device is unable to fulfill the time constraints, it can initiate a delayed response (DR) sequence. In order to support DR handling within nested communication, Subclause 6.6 defines the handling within FDT.

The responsibility to handle the DR responses from the device is located at the DTM that represents the device. The Communication DTM and Gateway DTMs (if used) shall ensure that DR responses are communicated correctly to the respective DTM. An example for such a delayed response handling is shown in Figure 3.

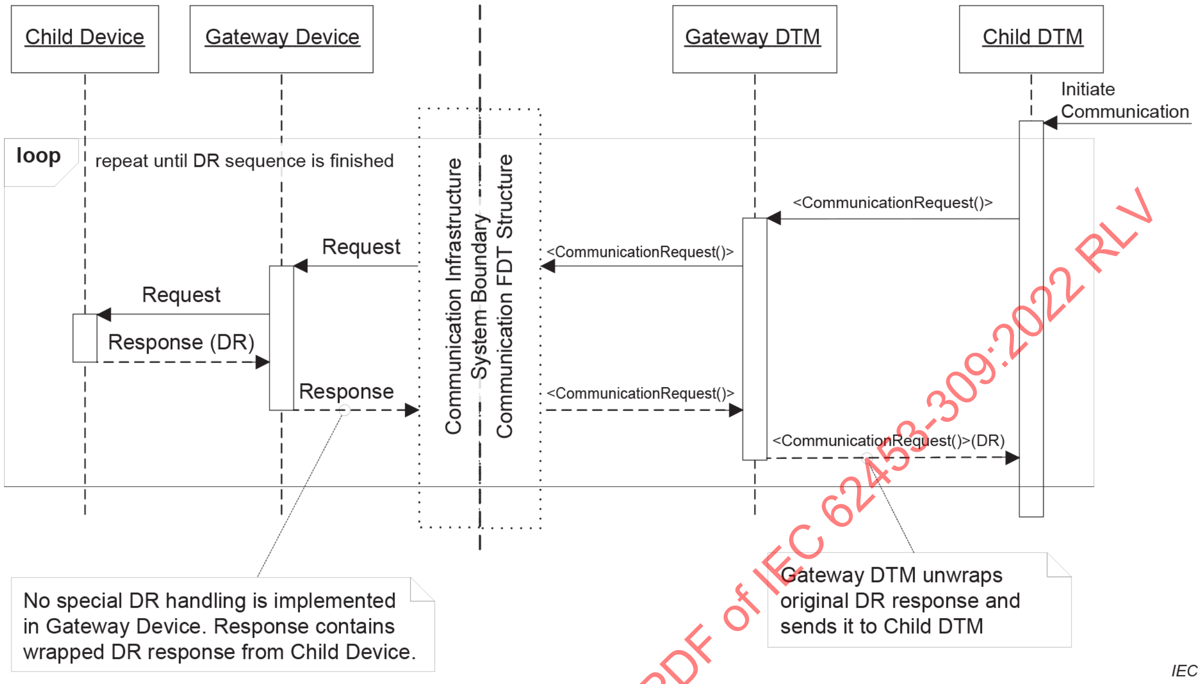


Figure 3 – Handling of delayed responses (scenario 1)

It is also possible that the two partners of a DR sequence are both devices. For example, a gateway device (e.g. WirelessHART Gateway) might execute a delayed response sequence with a child device (e.g. WirelessHART Adapter). In this case, the gateway device is responsible to handle the DR of the child device. The delayed responses will not reach the respective Child DTM. If the gateway device is unable to handle the DR directly, the gateway device itself could send DRs to the Gateway DTM. In such a case, the DRs would have to be handled by the respective Gateway DTM. Usually, the nested communication concept reflects the interaction between the devices. In the case described here, this is not possible and the implementation shall follow the sequence shown in Figure 4.

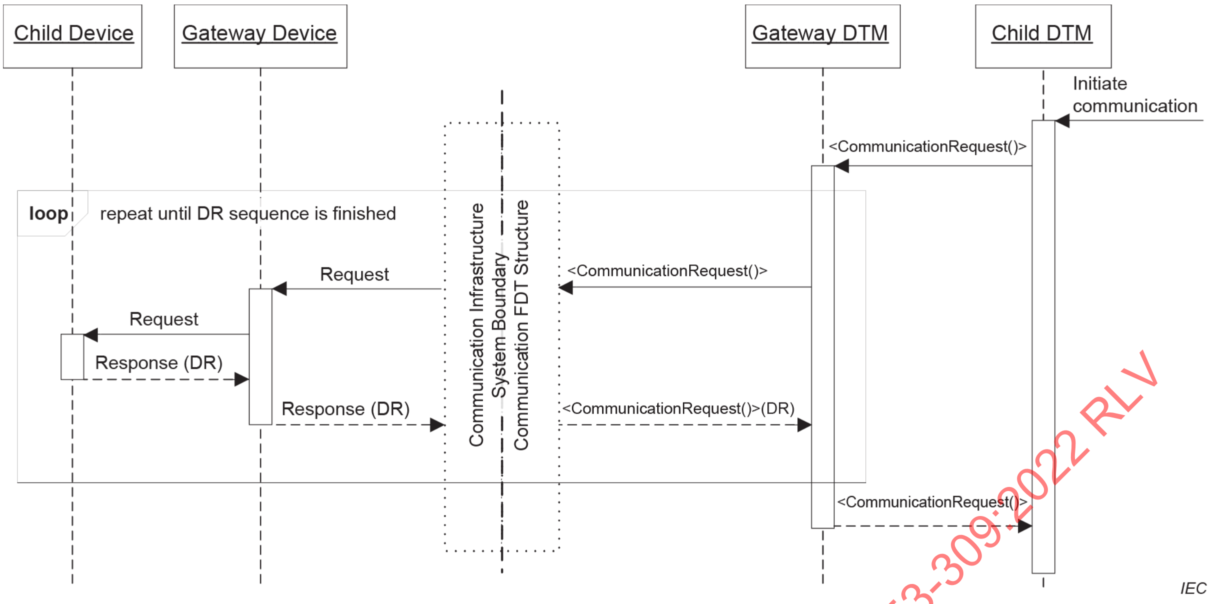


Figure 4 – Handling of delayed responses (scenario 2)

A DR sequence might take a long time that might disturb the usage of the FDT Frame Application and that might block user interaction. There is no timeout time definition existing for DR sequences and neither the DTM itself nor any other DTM in the nested communication chain is capable to initiate a timeout that could recover the system. Timeout time in such a case is application-dependent and shall be configurable by the user. When a DR sequence lasts an unreasonable amount of time, it shall be an aim to involve the user. If a DR sequence is used in a user interface, then configurable timeout mechanisms shall be implemented.

To handle DR responses with a reliable interoperability, the following rules shall be fulfilled:

- The DTM of a device that might send DR responses shall handle the DR responses of the device.
- DR responses that are not handled by other devices shall be propagated to the DTM that represents the device that sends the responses.
- A DTM shall be aware that it will not receive DR responses from the device, when the DR responses are handled by the parent device.
- A DTM that handles DR responses shall implement a user configurable timeout management that must allow the user to set a timeout.

6.7 Topologies with mixed HART protocols

6.7.1 General

HART DTMs using ‘Extended_HART’ protocols may also support the ‘HART_Basic’ protocol, in order to ensure compatibility with existing HART DTMs.

‘Extended_HART’ protocols were defined for better distinction between the different HART communication types. Using ‘Extended_HART’ and ‘HART_Basic’ protocols at the same time needs well defined processes to guarantee interoperability.

6.7.2 Behavior of DTMs supporting ‘Extended_HART’ only

The topology validation is performed by the Frame Application (reference). If the Communication Channel receives a call to ValidateAddChild(), it shall verify whether the given device type requires a suitable protocolId.

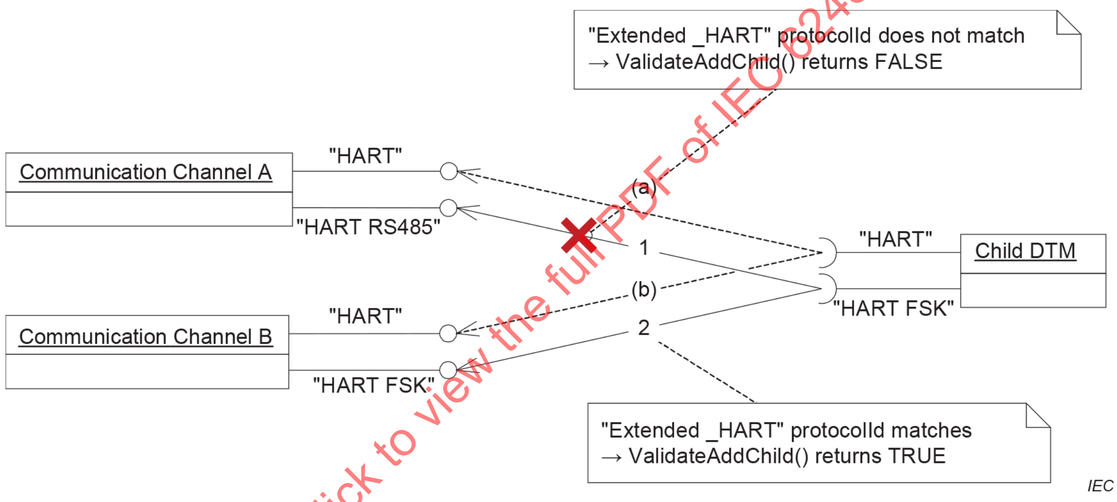
The behavior of such a DTM in a ValidateAddChild call is:

- If a match cannot be found, the ValidateAddChild() call shall be answered with FALSE.
- If a match was found, the ValidateAddChild() call shall be answered with TRUE. During the call to OnAddChild(), the Parent DTM sets the activeProtolID in the Child DTM to the current protocolId.

6.7.3 Behavior of DTMs supporting 'Extended_HART' and 'HART_Basic'

When creating topologies, a Frame Application shall check the communication compatibility of a Child DTM and a Parent DTM by comparing the lists of supported and required protocols. Based on 'Extended_HART' protocols, a more effective topology validation is possible, but if both DTMs support additionally the 'HART_Basic' protocol, this may result in invalid topologies.

When for example a Communication Channel, which supports 'HART_RS485' and 'HART_Basic', and a Device DTM, which requires 'HART_FSK' and 'HART_Basic', are connected (see Communication Channel A in Figure 5), a Frame Application will allow to connect those DTMs because of the matching 'HART_Basic' protocolId. But in fact this is an invalid topology.



Frame Application will allow to attach Child DTM to both Communication Channels because at least (a) and (b) are possible. But the mismatch in (1) allows the 'Communication Channel A' to detect the mismatch and decline the attachment of the 'Child DTM'.

Figure 5 – Behavior of DTMs supporting 'Extended_HART' and 'HART_Basic'

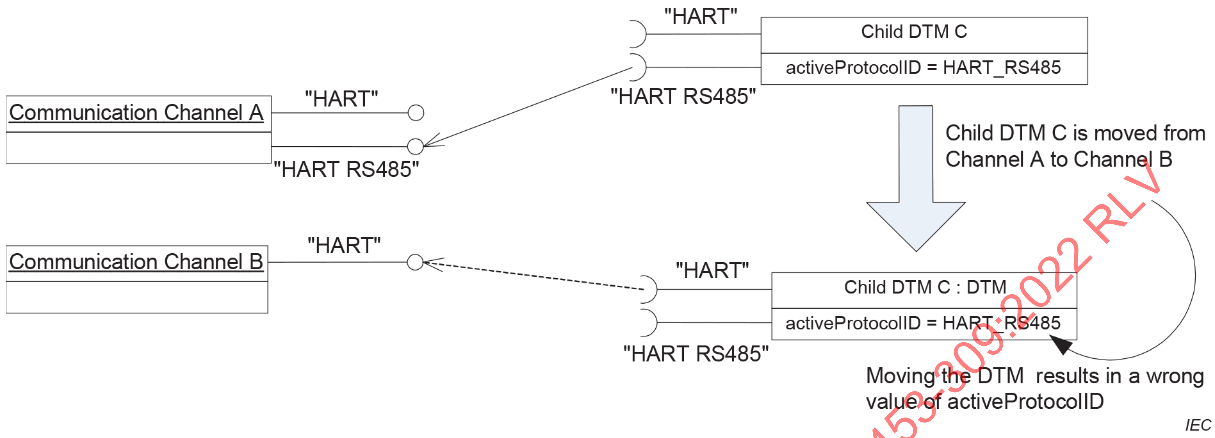
To prevent such a situation, a Communication Channel supporting 'Extended_HART' protocols and 'HART_Basic' protocol shall check during a ValidateAddChild() call if a DTM is connected that requires also 'Extended_HART' protocol and 'HART_Basic' protocol and if there exists a matching 'Extended_HART' protocol.

- If no match can be found in the 'Extended_HART' buscategories, the Communication Channel shall answer the ValidateAddChild() call with FALSE.
- If a match can be found, the ValidateAddChild() call shall be answered with TRUE. During the OnAddChild() call, the Communication Channel sets the activeProtocolID in the Child DTM.

6.7.4 Behavior of DTMs that require 'Extended_HART' or 'HART_Basic'

With the attribute 'activeProtocolID', a DTM is informed about the current connection type in the topology. But this procedure may fail when the DTM is connected to a Communication Channel that does not implement the 'activeProtocolID' management.

Assuming for example that a Child DTM was connected to Communication Channel using ‘HART_RS485’ with the result that ‘activeProtocolID’ is set to ‘HART_RS485’ (see Figure 6). The Frame Application now moves the Child DTM to a Communication Channel that only supports ‘HART_Basic’. When the Child DTM now tries to establish a ‘HART_RS485’ connection, this might result in an error.



Moving DTM C from Channel A to Channel B results in an inconsistent activeProtocolID value, because DTM B does not support Enhanced_HART protocol and therefore does not set activeProtocolID.
→ Before DTM C executes a connect request, it shall validate whether the Parent DTM provides the communication related to the current activeProtoclID value.

Figure 6 – Behavior of DTMs requires ‘Extended_HART’ or ‘HART_Basic’

Therefore, it is required that a DTM which requires both ‘Extended_HART’ and ‘HART_Basic’ needs to check the capabilities of the Communication Channel before establishing a connection.

6.8 Nested communication with multiple gateways

HART supports topologies in the physical network that allow having multiple gateways in a communication chain. An example for such a topology are wired HART devices connected to a wireless adapter communicating to a wireless gateway (see 6.9).

General concept of nested communication is that a device receives the command data that was generated by its respective DTM and that the DTM receives the response data of its respective device. Also required in nested communication is that the Child DTM always is the active sender and therefore is not allowed to pass through communication sent by its Child DTM without encapsulation or transformation.

With command #77 (send to sub-device), HART defines a standard encapsulation mechanism to propagate communication through a network topology. Each request that was sent to a sub-device shall be encapsulated in a command #77 request before forwarding it to the gateway device. When a response to a command #77 is returned, the Gateway DTM shall unpack this command and send its contained response data to the respective Child DTM.

Depending on the implementation in the gateway, a command #77 might be restructured to another command structure. In this case, the Gateway DTM has the responsibility to transform incoming command #77 requests from the Child DTM to the gateway specific commands and also to restructure the resulting responses back again respectively to responses on the originally received command #77 request.

6.9 Communication- and network structures in WirelessHART

6.9.1 General

WirelessHART defines a rich and secure protocol between devices using 2,4 GHz wireless technology. Host systems are not intended to interact with the WirelessHART network directly. Using a WirelessHART gateway device, a host system can communicate with any WirelessHART device using HART master/slave transactions, without requiring specific knowledge of the WirelessHART protocol.

HART specifies three standard types of WirelessHART devices:

1) WirelessHART gateway device:

   This device connects a WirelessHART network to the world via HART or other protocols that allow data transfer with high baud rates. It is possible to have more than one WirelessHART gateway device active in a WirelessHART network. The WirelessHART gateway devices are responsible to manage the network directory and propagate information from and to the WirelessHART devices.

   For better readability in 6.9, a WirelessHART gateway device is simply named Gateway.

2) WirelessHART field device:

   The WirelessHART field device is a device that can participate in a WirelessHART network.

   For better readability in 6.9, a WirelessHART field device is simply named Field Device.

3) WirelessHART adapter device:

   The WirelessHART adapter device is a specialized WirelessHART field device that allows the connection of HART FSK and/or 4 mA to 20 mA sub-devices to the WirelessHART network.

   For better readability in 6.9, a WirelessHART adapter device is simply named Adapter and devices connected to an Adapter are simply called Sub-Devices.

Subclause 6.9 will focus on specialties of WirelessHART and define implementation rules within FDT that are required for nested communication.

6.9.2 Network topology

Adapters are special devices that connect other HART physical layers (usually HART FSK) with the WirelessHART network as shown in Figure 7.

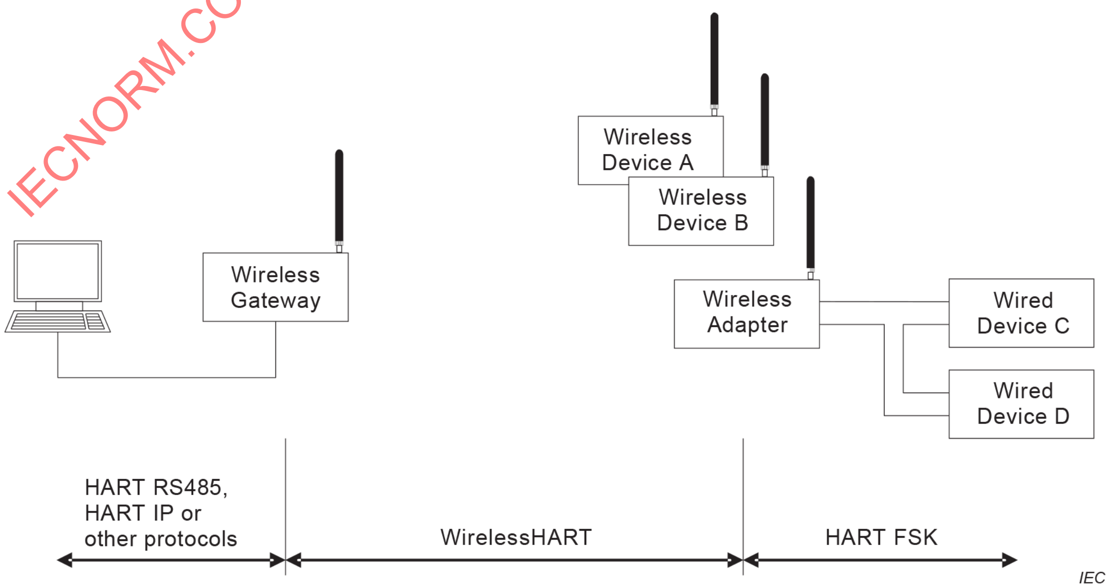


Figure 7 – Host connected to a WirelessHART gateway device

From the perspective of nested communication in FDT, the Gateway and the Adapter are both gateway devices that shall be presented as such in the network topology of an FDT Frame Application. The resulting FDT topology is shown in Figure 8.

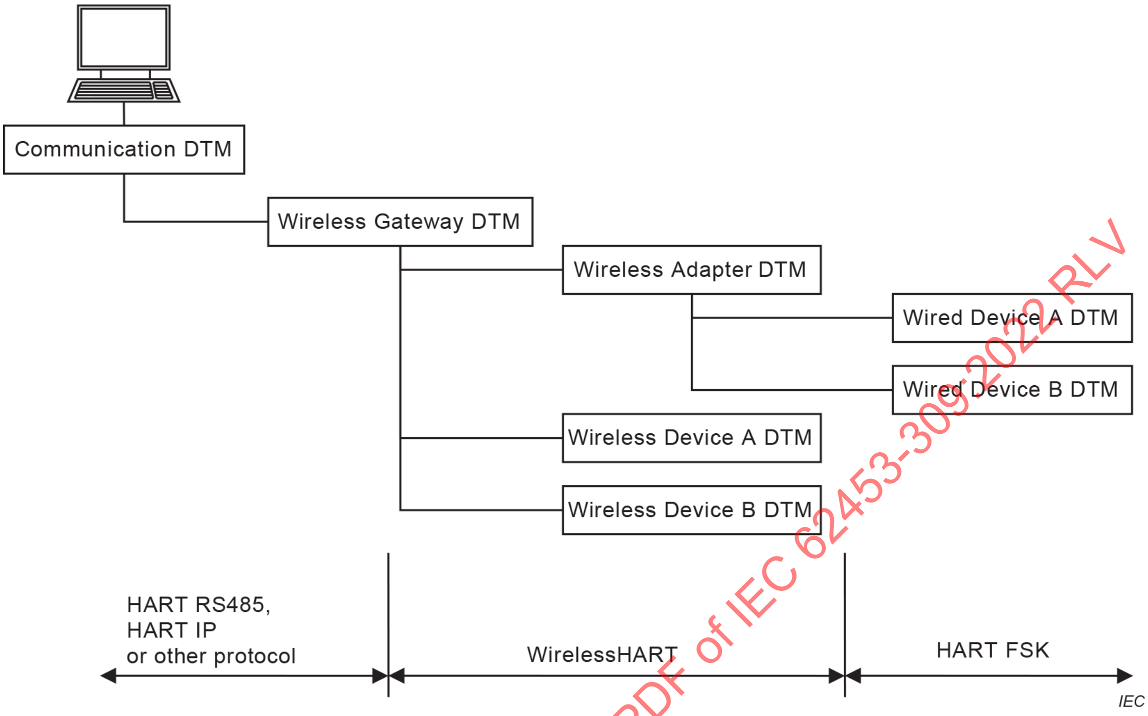


Figure 8 – FDT Topology of a WirelessHART network

An Adapter interacts with the HART FSK loop like a common HART device. It is acting as a HART Master but can also be addressed with HART transactions from another Master. Especially in service use cases, an FDT Frame Application might be connected to the HART FSK loop to directly access the Adapter. In this case, the Adapter is connected to the FDT Frame Application as a usual device in a HART FSK multidrop and multimaster scenario like shown in Figure 9.

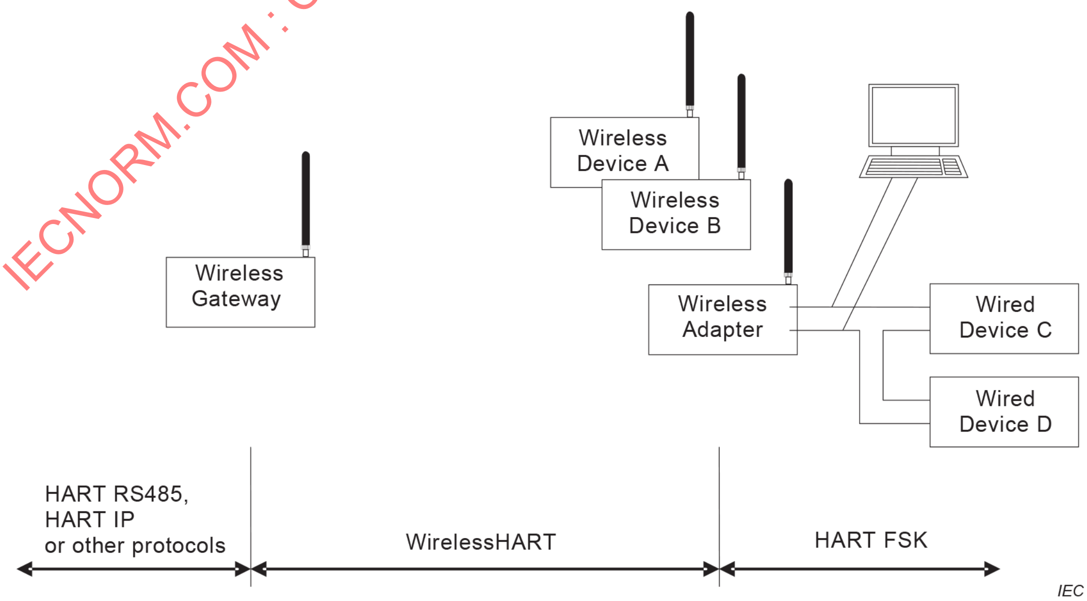


Figure 9 – Host connected to HART FSK

In this use case, the network topology in the FDT Frame shall be structured as shown in Figure 10.

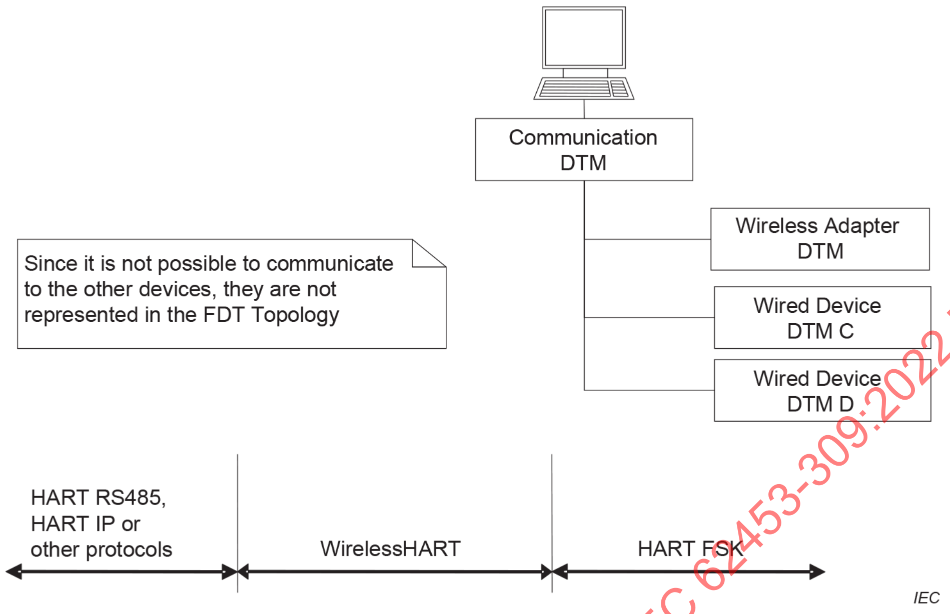


Figure 10 – FDT Topology when directly connected to a WirelessHART adapter device

Figure 8 and Figure 10 show that an Adapter DTM shall implement gateway functions when it is used in a WirelessHART FDT environment and on the other hand shall behave like a simple device when used in a HART FSK environment. In FDT 2, a DTM is always informed about changes in communication type. Using the information about the current connection type, the Adapter DTM shall implement the respective specific structural behavior.

As a summary of the above paragraphs of 6.9.2, the following rules shall be implemented:

If the Adapter is connected via WirelessHART, it:

- shall interact as a Gateway DTM;
- shall handle communication to the connected Sub-Devices (as specified in 6.7) that are attached in the topology as child DTMs;
- shall handle DR transactions as described in 6.8.

If the Adapter is connected via HART FSK, it:

- shall deny attachment of Child DTMs.
- shall deny connection to Child DTMs.

If an instance of an Adapter DTM is moved from a WirelessHART Communication Channel to a HART FSK Communication Channel, it:

- shall keep all instances of the Child DTMs untouched;
- shall allow moving Child DTMs away from its node.

7 Protocol-specific usage of general data types

Table 3 shows how general data types, defined in IEC 62453-2 within the namespace 'fdt', are used with HART devices.

Table 3 – Protocol specific usage of general data types

Data type	Description for use
fdt:address	The address property is not mandatory for the exposed parameters in the DTMs. But if the address property is used, the string shall be constructed according to the rules of the semanticId. That means the property 'semanticId' is always the same as the property 'address'
fdt:protocolId	See Clause 4
fdt:deviceTypeId	The property "fdt:DtmDeviceType.deviceTypeId" shall contain the DeviceTypeID of the supported physical device according to the FieldComm Group's online product catalog
fdt:manufacturerId	Enter manufacturer according FieldComm Group's list
fdt:semanticId fdt:applicationDomain	The applicationDomain attribute is: FDT_HART The sematicId for protocol related parameter is directly related to the protocol specification. The definition of the commands is the base for the semanticId. The semanticId for a parameter follows the following definition: CMDxxBy and CMD31EXTENDEDxxBy for extended HART 6 device family commands. The semanticIds for the Response Byte 0 and 1 defined in the IEC 61784 CPF 9 specification are: CMDxxRESPONSE_BYTE_0 CMDxxRESPONSE_BYTE_1 xx: represents the command number, getting the parameter via IEC 61784 CPF 9 protocol or the device family command number y: start byte within the command definition xx, y are based on decimal format without leading '0'
subDeviceType	Enter manufacturer specific value

8 Protocol-specific common data types

Not applicable.

9 Network management data types

9.1 General

The data types specified in 9.1 are used in the following services:

- NetworkManagementInfoRead service;
- NetworkManagementInfoWrite service.

9.2 Addressing modes

The addressing mode depends on the type of the used HART protocol. Also additional addressing information might be necessary for some types of HART protocols. Table 4 shows the dependency of usable addressing modes and additional address information in dependency of the HART protocol in use.

Table 4 – Relation of ProtocolId and supported features

ProtocolId	Supported Addressing Modes	Address Data Type	Exposed Data	Comment
HART_Basic	ShortAddress	Used attributes: – shortAddress	As described in 9.3	This is defined for backward compatibility. New products should not use this bus category. Only single byte ManufacturerID and DeviceTypeID are supported.
HART_FSK	ShortAddress, ShortTag, LongTag	Used attributes: – shortAddress – shortTag – longTag	As described in 9.4	A DTM may use more than one of these Ids if the device supports multiple physical connections e.g. WirelessHART and FSK.
HART_Wireless				
HART_RS485				
HART_Infrared				
HART_IP	ShortAddress, ShortTag, LongTag	Used attributes: – shortAddress – shortTag – longTag – ipAddress – port	As described in 9.4	

NOTE The 'HART_Basic' protocol is maintained for backward compatibility only. In this document, the other protocols (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP) are referenced as 'Extended_HART' protocols (e.g. for definitions that apply to all protocols except 'HART_Basic'.)

9.3 Address information

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

9.4 Additional address information for 'Extended HART' protocols

DTMs that implement 'Extended_HART' protocols as required protocol shall provide address information as defined in 9.4 (in addition to address information according to 9.3). The information shall be provided as described in Table 5 and Table 6.

Support for all datatypes described in Table 5 and Table 6 is mandatory. If the data is not used (e.g. ipAddress), they shall be set to a type correct default value. The information in 9.4 is used for data interchange purpose between Parent DTM and Child DTM. All network information provided by a Child DTM may be changed by the Parent DTM (i.e. access is read-/write-able), except for hartVersion and pollingAddressRange which can be read only.

Table 5 – Simple address information data types

Data type	Definition	Description
activeProtocolId	enumeration (<Identifier values from Table 1>)	activeProtocolId is set by the Parent DTM to inform the child DTM what bus category is used (see Clause 4). If not used, set to identifier value that stands for HART.
addressingMode	enumeration (shortAddress | shortTag | longTag | longAddress)	Specifies how the communication will be established during the connect request.
hartVersion	INT	This value shall be set by the DTM itself to document the HART major version the device supports.
ipAddress	STRING	This value is set to the IP address used to connect to the device when using an IP based physical layer.
ipProtocolVersion	enumeration (IPv4 | IPv6)	This value specifies the version of the IP protocol which is used.
longAddressByte1	USINT	First byte of unique device identifier (long frame address). For HART 7: First byte of 16-bit Extended Device Type For HART 5 and 6: Composed from manufacturer id, master address bit and burst mode bit.
longAddressByte2	USINT	Second byte of unique device identifier (long frame address). For HART 7: First byte of 16-bit Extended Device Type For HART 5 and 6: 1 byte of device type code.
longAddressByte3	USINT	Third byte of unique device identifier (long frame address). First byte of unique device identifier.
longAddressByte4	USINT	Forth byte of unique device identifier (long frame address). Second byte of unique device identifier.
longAddressByte5	USINT	Fifth byte of unique device identifier (long frame address). Third byte of unique device identifier.
longTag	STRING	Value containing the long tag information that is used when connecting using addressingMode =longTag
networkID	INT	Stores the Network ID for a HART Wireless network (if applicable).
pollingAddressRange	enumeration ('0 to 15' | '0 to 63')	This value is set by the DTM itself to document the address range for the polling address of the device.
port	INT	This value is set to the port used to connect to the device when using an IP based physical layer. If not used, set to 0.
shortTag	STRING	Value containing the 8 character PACKED_ASCII tag that is used when connecting using addressingMode = shortTag

Table 6 – Structured address information data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
HartNetworkData	STRUCT			Data in this section is set by the DTM itself to provide general information
	hartVersion	M	[1..1]	
	pollingAddressRange	M	[1..1]	
HartDeviceAddress	STRUCT			Data in this section is communication relevant data that the parent DTM sets and that shall be sent to the parent DTM back (e.g. with ConnectRequest)
	shortTag	M	[1..1]	
	longTag	M	[1..1]	
	addressingMode	M	[1..1]	
	ipAddress	M	[1..1]	
	port	M	[1..1]	
	ipProtocolVersion	M	[1..1]	
	networkID	M	[1..1]	
	longAddressByte1	M	[1..1]	
	longAddressByte2	M	[1..1]	
	longAddressByte3	M	[1..1]	
	longAddressByte4	M	[1..1]	
	longAddressByte5	M	[1..1]	
FdtHartExtension	STRUCT			Data in the root of FdtHartExtension is set by the parent DTM directly after attachment during topology management
	activeProtocolId	M	[1..1]	The initial value shall be 'HART_Basic'
	HartNetworkData	M	[1..1]	
	HartDeviceAddress	M	[1..1]	

10 Communication data types

10.1 General

The data types described in Clause 10 are used in the following services:

- connect service;
- disconnect service;
- transaction service.

The service arguments contain the address information and the communication data (explained in Table 7 and Table 8).

10.2 Protocol-specific Addressing Information

With the 'Extended_HART' protocols additional addressing information needs to be exchanged in order to establish a connection with the device. The additional addressing information is specific to the protocolId and version of the HART protocol.

For example HART 6 FSK device supports LongTag, and HART IP shall handle IP Address additionally.

During OnScanResponse(), the Communication Channel shall provide the additional addressing information for each device in the scan result, through the in-line schema extensions.

Frame Application can use this additional information to set the addressing information to the Child DTM through SetParameters().

10.3 Datatype definitions

To establish connection with a device, the DTM shall send protocol specific address information during ConnectRequest(). This information is used by the Communication Channel to address the device.

Child DTMs supporting an 'Extended_HART' protocol shall send additional addressing information as part of the ConnectRequest(), using the respective protocol-specific datatypes (see Table 7 and Table 8).

The Communication Channel supporting the 'Extended_HART' protocol can read the additional addressing information available in the ConnectRequest(), and use this information to address the device.

The data types described in 10.3 are defined for the following namespace.
Namespace: fdthart

Table 7 – Simple communication data types

Data type	Definition	Description
address1	USINT	Address information according to the IEC 61784 CPF 9 specification
address2	USINT	Address information according to the IEC 61784 CPF 9 specification
address3	USINT	Address information according to the IEC 61784 CPF 9 specification
addressingMode	enumeration (shortAddress | shortTag | longTag | longAddress)	Specifies which information will be used for creating the connection.
burstFrame	BOOL	Information whether the IEC 61784 CPF 9 response is a burst frame (message) or not
burstModeDetected	BOOL	Indicates whether the Communication Channel has detected that the device is already in burst mode. This is detected during a subscription request
commandNumber	USINT	Address information according to the IEC 61784 CPF 9 specification
communicationReference	UUID	Mandatory identifier for a communication link to a device This identifier is allocated by the communication component during the connect. The address information shall be used for all following communication calls
delayTime	UDINT	Minimum delay time in [ms] between two communication calls

Data type	Definition	Description
deviceStatus	USINT	Status information. This is the second status byte returned in command responses according to the IEC 61784 CPF 9 specification
deviceTypeId	USINT	Address information according to the IEC 61784 CPF 9 specification
ipAddress	STRING	This value is set to the IP address used to connect to the device when using an IP based physical layer.
ipProtocolVersion	enumeration (IPv4 | IPv6)	This value specifies the version of the IP protocol which is used.
longFrameRequired	BOOL	Address information according to the IEC 61784 CPF 9 specification
longAddressByte1	USINT	First byte of unique device identifier (long frame address). Composed from manufacturer id, master address bit and burst mode bit.
longAddressByte2	USINT	Second byte of unique device identifier (long frame address). 1 byte of device type code.
longAddressByte3	USINT	Third byte of unique device identifier (long frame address). First byte of unique device identifier.
longAddressByte4	USINT	Forth byte of unique device identifier (long frame address). Second byte of unique device identifier.
longAddressByte5	USINT	Fifth byte of unique device identifier (long frame address). Third byte of unique device identifier.
longTag	STRING	Value containing the long tag information that is used when connecting using addressingMode =longTag
manufacturerId	USINT	Address information according to the IEC 61784 CPF 9 specification (Table: VIII, MANUFACTURER IDENTIFICATION CODES)
networkID	INT	Stores the Network ID for a HART Wireless network (if applicable).
port	INT	This value is set to the port used to connect to the device when using an IP based physical layer.
preambleCount	USINT	At the connect request the attribute is optional and contains a hint for the communication component about the number of preambles, required by the device type. At the connect response the attribute is mandatory and contains the information about the currently used preambleCount
primaryMaster	BOOL	At the connect request the attribute is optional and contains a hint for a communication component that a DTM requires communication as primary or secondary master. At the connect response the attribute is mandatory and contains the information about the current state of the master
sequenceTime	UDINT	Period of time in [ms] for the whole sequence
shortAddress	USINT	Address information according to the IEC 61784 CPF 9 specification. This value is accessible via the attribute slaveAddress. SlaveAddress is part of the BusInformation structure. These values shall be set by the responsible component as described in clause Nested Communication of IEC 62453-2
shortTag	STRING	Value containing the 8 character PACKED_ASCII tag that is used when connecting using addressingMode =shortTag
value	USINT	Variable for status information
fdt:systemTag	STRING	System Tag of a DTM. It is strongly recommended to provide the attribute in the Request document.

Table 8 – Structured communication data types

Data type	Definition: Elementary data types	Usage	Multiplicity	Description
Abort	STRUCT			Describes the abort
	communicationReference	O	[0..1]	
CommandResponse	STRUCT			Status information. This is computed from the first status byte returned in command responses according to the IEC 61784 CPF 9 specification. If bit 7 of the first status byte is clear this value contains the value in the first status byte. If bit 7 is set this element is not returned in the status structure
	value	M	[1..1]	
CommunicationStatus	STRUCT			Status information. This is computed from the first status byte returned in command responses according to the IEC 61784 CPF 9 specification. If bit 7 of the first status byte is set this value contains the value in the first status byte (This is where we need to state whether it is the first status byte or bits 0-6 of the first status byte). If bit 7 is clear this element is not returned in the status structure
	value	M	[1..1]	
ConnectRequest	STRUCT			Describes the communication request for 'HART_Basic' protocol.
	fdt:tag	M	[1..1]	
	preambleCount	O	[0..1]	
	primaryMaster	O	[0..1]	
	longFrameRequired	O	[0..1]	
	fdt:systemTag	O	[0..1]	
	LongAddress	O	[0..1]	
	ShortAddress	M	[1..1]	
ExConnectRequest	STRUCT			Describes the communication request for 'Extended_HART' protocols.
	fdt:tag	M	[1..1]	
	shortAddress	O	[1..1]	
	addressingMode	M	[1..1]	
	ipAddress	O	[0..1]	
	port	O	[0..1]	
	preambleCount	O	[0..1]	
	primaryMaster	O	[0..1]	
	fdt:systemTag	O	[0..1]	
	LongAddress	O	[0..1]	
ConnectResponse	STRUCT			Describes the communication response
	fdt:tag	M	[1..1]	
	preambleCount	M	[1..1]	
	primaryMaster	M	[1..1]	
	communicationReference	M	[1..1]	
	LongAddress	O	[0..1]	
	ShortAddress	O	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DataExchange-Request	STRUCT			Describes the communication request
	commandNumber	M	[1..1]	
	communicationReference	M	[1..1]	
	fdt:CommunicationData	O	[0..1]	
DataExchange-Response	STRUCT			Describes the communication response
	commandNumber	M	[1..1]	
	communicationReference	M	[1..1]	
	burstFrame	O	[0..1]	
	fdt:CommunicationData	O	[0..1]	
	Status	M	[1..1]	
DisconnectRequest	STRUCT			Describes the communication request
	communicationReference	M	[1..1]	
DisconnectResponse	STRUCT			Describes the communication response
	communicationReference	M	[1..1]	
SubscribeRequest	STRUCT			Describes the subscription request for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
SubscribeResponse	STRUCT			Describes the subscription response request for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
	burstModeDetected	M	[1..1]	
	fdt:communicationError	O	[0..1]	
UnsubscribeRequest	STRUCT			Describes the request to release the subscription for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
UnsubscribeResponse	STRUCT			Describes the response request to release the subscription for device initiated data transfer (IEC 61784 CPF 9 burst mode)
	communicationReference	M	[1..1]	
	fdt:communicationError	O	[0..1]	

Data type	Definition			Description
	Elementary data types	**Usage**	**Multiplicity**	
LongAddress	STRUCT			Address information according to the IEC 61784 CPF 9 specification (only supported by devices based on HART revision > 5, see related documentation) In the IEC 61784 CPF 9 protocol Manufacturer ID and Device type ID are contained in the longaddress. If the channel delivers different values in fdthart:manufacturerId / fdthart:deviceTypeId and in the corresponding bytes in fdthart:LongAddress, the following rule applies: * the fdthart:LongAddress shall be used for communication and * the fdthart:manufacturerId and fdthart:deviceTypeId may be used only as information about the manufacturer and the type of device
	longAddressByte1	M	[1..1]	
	longAddressByte2	M	[1..1]	
	longAddressByte3	M	[1..1]	
	longAddressByte4	M	[1..1]	
	longAddressByte5	M	[1..1]	
SequenceBegin	STRUCT			Describes the sequence begin
	sequenceTime	O	[0..1]	
	delayTime	O	[0..1]	
	communicationReference	M	[1..1]	
SequenceEnd	STRUCT			Describes the sequence end
	communicationReference	M	[1..1]	
SequenceStart	STRUCT			Describes the sequence start
	communicationReference	M	[1..1]	
ShortAddress	STRUCT			Address information according to the IEC 61784 CPF 9 specification
	shortAddress	M	[1..1]	
Status	STRUCT			Status information according to the IEC 61784 CPF 9 specification
	deviceStatus	M	[1..1]	
	choice of	M	[1..1]	
	CommunicationStatus	S	[1..1]	
	CommandResponse	S	[1..1]	

The property ‘fdt:tag’, is part of the DtmDevice data type and contains the IEC 61784 CPF 9-specific value called TAG, which is used, for example within command #11, ‘READ UNIQUE IDENTIFIER ASSOCIATED WITH TAG’. This value shall be set by the responsible component as described in the Nested Communication publication IEC 62453-2.

11 Channel parameter data types

It is up to a DTM whether it provides any channels. If a DTM allows a Frame Application, other DTMs, or a controller to directly accessing the process values of its device via IEC 61784 CPF 9 protocol, then the DTM should provide FDT-Channel objects as described in Clause 11. Only the complete description of all channels belonging to a command allows proper access for external applications.

The description of channels, especially of the process values, allows the Frame Application to support the device in a more efficient way.

Used at ReadChannelData service and WriteChannelData service.

The information returned by the ReadChannelData service describes how to access an I/O value via a command (see Table 9 and Table 10).

The data types described in Clause 11 are defined for the following namespace.
Namespace: hartchannel

Table 9 – Simple channel parameter data types

Data type	Definition	Description
byteLength	USINT	Number of static bytes in a Request or in a Reply
commandNumber	UDINT	Number of the command containing the channel value
frameApplicationTag	STRING	Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces
gatewayBusCategory	UUID	Unique identifier for a supported bus type according to the FDT specific CATID
protectedByChannelAssignment	BOOL	TRUE if the channel is set to read only by the Frame Application. Usually set to TRUE if a channel assignment exists
value	STRING	Current value of a channel for read or write

Table 10 – Structured channel parameter data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
CommandParameters	STRUCT			Static command parameter bytes in a Request or in a Reply
	fdt:binData	O	[0..1]	
	byteLength	M	[1..1]	
FDTChannel	STRUCT			Description of the channel
	fdt:tag	M	[1..1]	
	fdt:id	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	protectedByChannel-Assignment	M	[1..1]	
	fdt:dataType	M	[1..1]	
	byteLength	M	[1..1]	
	fdt:signalType	M	[1..1]	
	frameApplicationTag	O	[0..1]	
	appId:applicationId	O	[0..1]	
	fdt:SemanticInformation	O	[0..*]	
	fdt:BitEnumeratorEntries	O	[0..1]	
	fdt:EnumeratorEntries	O	[0..1]	
	fdt:Unit	O	[0..1]	
	ReadCommand	O	[0..1]	
	WriteCommand	O	[0..1]	
	fdt:Alarms	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:Deadband	O	[0..1]	
	fdt:SubstituteValue	O	[0..1]	
FDTChannelType	STRUCT			Description of the channel component in case of channels with gateway functionality
	fdt:VersionInformation	M	[1..1]	
	gatewayBusCategory	O	[0..1]	
ReadCommand	STRUCT			Description of the command to read the channel from a device
	commandNumber	M	[1..1]	
	Request	O	[0..1]	
	Reply	O	[0..1]	
	ResponseCodes	O	[0..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Reply	STRUCT			Description of the reply structure of a command according to the IEC 61784 CPF 9 specification
	collection of	M	[1..1]	
	fdt:ChannelReference		[0..*]	
	CommandParameters		[0..*]	
	ResponseCodes	O	[0..1]	
Request	STRUCT			Description of the request structure of a command according to the IEC 61784 CPF 9 specification
	collection of	M	[1..1]	
	fdt:ChannelReference		[0..*]	
	CommandParameters		[0..*]	
ResponseCodes	STRUCT			Collection of specific response codes according to the IEC 61784 CPF 9 specification (known as COMMAND-SPECIFIC RESPONSE CODES)
	fdt:EnumeratorEntry	M	[1..*]	
WriteCommand	STRUCT			Description of the command to write the channel to a device
	commandNumber	M	[1..1]	
	Request	O	[0..1]	
	Reply	O	[0..1]	
	ResponseCodes	O	[0..1]	

12 Device identification

12.1 Protocol-specific handling of data type STRING

IEC 61784 CPF 9 char array rules:

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

12.2 Address Range for Scan

The Frame Application can specify the bus address range to the Communication Channel for scanning. The supported scan range is specific to the protocol. Table 11 describes how the BusAddressRange and ScanMode attributes can be used for different HART protocols.

Table 11 – Address range for device identification

Protocol	Comments
HART	The Frame Application can specify the short address range for scan.
HART_FSK	Only if the addressingMode of the Communication DTM is shortAddress, this is applicable.
HART_Wireless	The Frame Application can use the ScanMode, to specify to scan all addresses or to request the Communication DTM to open GUI
HART_RS485	Only if the addressingMode of the Communication DTM is shortAddress, this is applicable.
HART_Infrared	The Frame Application can use the ScanMode, to specify to scan all addresses or to request the Communication DTM to open GUI
HART_IP	The Frame Application can use the ScanMode, to specify to scan all addresses or to request the Communication DTM to open GUI to specify select IP addresses.

12.3 Support for Extended Manufacturer and Device Type Code

HART 7 devices support extended manufacturer id and device type codes. With all 'Extended_HART' protocols, the extended manufacturer id and device type codes are supported.

Parent DTMs supporting the 'Extended_HART' protocols shall use the in-line schema extension during the OnScanResponse(). Through the in-line schema the Parent DTM can provide additional information, e.g. extended manufacturer id, device type code, Device Id, HART long tag and short tag.

A Frame Application can use this additional information for assigning the DTM based on the extended manufacturer id and device type code. The extended FDT 1.2 scan result document format is specified in Table 25.

12.4 Device type identification data types for protocol 'HART_Basic'

The data types described in 12.4 are reused as defined by 12.6 and 12.9.

The IEC 61784 CPF 9 device type identification data types provide general data types with a protocol-specific semantic (see Table 12 and Table 13) as well as data types without such a mapping (see Table 14 and Table 15).

The data types described in 12.4 are defined for following namespace.
Namespace: hartident

Table 12 – Identification data types with protocol-specific mapping for protocol 'HART_Basic'

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
shortAddress	IdAddress	Poll possible address range (HART5: [0-15], HART 6: [0-63]) by calling Cmd #0. If Cmd #0 response is available, a physical device is connected to this address. Cmd #0 response does not contain short address value whether the short or long format is used. If master using short address for polling receives a response, it can assume that short address of device is the same as used in the polling request. In addition to this, polling address can be read from HART 6 device with cmd #7	Polling Address	Unsigned 8	USINT	See [3] Chapter 6.8 Command 7 Read Loop Configuration
busProtocol	IdBusProtocol	CommChannel shall pass "HART" in this attribute	HART	Enumeration: "HART" for HART5 and HART6	enumeration (HART)	
universalCommandRevisionLevel	IdBusProtocolVersion	Command #0 Byte 4	HART Revision	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
manufacturerIdentificationCode	IdManufacturer	HART7: Command #0 Byte 17+18. HART<7: Command #0 Byte 1.	Manufacturer Identification Code	8 bit unsigned integer Example: Endress+Hauser: 17 (0x11)	UINT (dec)	See [4] Chapter 5.8 Manufacturer Identification Codes
deviceTypeID	IdTypeID	HART 7: Command #0 Byte 1+2. HART<7: Command #0 Byte 2.	Device Type Code	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
softwareRevision	IdSoftwareRevision	Command #0 Byte 6	Software Revision	8 bit unsigned integer	USINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
hardwareRevision	IdHardwareRevision	Command #0 Byte 7 (Most significant 5 bits).	Hardware Revision	5 bit unsigned integer	INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
physicalSignalingCode	-	Command #0 Byte 7 (Least significant 3 bits).	Physical Signaling Code	3 bit unsigned enumerator	INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
tag	IdTag	Command #13 Bytes 0 – 5	Tag	6 Bytes or Packed ASCII characters	STRING	See [3] Chapter 6.13 Command 13 Read Tag, Descriptor, Date
deviceID	IdSerialNumber	Command #0 Bytes 9 – 11	Device Identification Number	Unsigned 24	UDINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
N/A	IdDTMSupportLevel	Not applicable for scan / physical device. Attribute to be used only in DTMDeviceType identification. Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	DTM Support Level	-	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	-

Table 13 – Identification data types with semantics for protocol ‘HART_Basic’

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	XML-FDT format (display format)	Specification reference
deviceCommandRevisionLevel	-	Command #0 Byte 5	Device Revision Level	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
deviceFlag	-	Command #0 Byte 8	Flags	Bit value according Flag Assignment table. 8 bit – unsigned int	USINT (hex)	See [4] Chapter 5.11 Table Flag Assignments
manufacturerSpecificExtension		Can be used by DTM for vendor specific device identification information, for example by combining a number of device parameter values into one string value. This can be used to identify a specific device variant			STRING	

Table 14 – Simple identification data types for protocol 'HART_Basic' with protocol independent semantics

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	enumeration genericSupport profileSupport blockspecificProfileSupport specificSupport
match	STRING	Used by Device DTM to define a regular expression which shall match to scanned physical define identification information
nomatch	STRING	Used by Device DTM to define a regular expression which shall not match to scanned physical define identification information. Used by Device DTM to indicate if identification information may not match

Table 15 – Structured identification data types for protocol 'HART_Basic' with protocol independent semantics

Elements	Definition			Description
	Elementary data types	Usage	Multiplicity	
RegExpr	STRUCT			Includes regular expression string – either for match or for nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.5 Common device type identification data types for 'Extended_HART' protocols

The data types described in 12.5 are reused as defined by 12.6 and 12.9.

The IEC 61784 CPF 9 device type identification data types provide general data types with a protocol-specific semantic (see Table 16 and Table 17) as well as data types without such a mapping (see Table 18 and Table 19).

The data types described in 12.5 are defined for following namespace.
Namespace: hartident2

Table 16 – Identification data types for 'Extended_HART' protocols with protocol-specific mapping

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
shortAddress	IdAddress	Poll possible address range (HART5: [0-15], HART >=6: [0-63]) by calling Cmd #0. If Cmd #0 response is available, a physical device is connected to this address. Cmd #0 response does not contain short address value whether the short or long format is used. If master using short address for polling receives a response, it can assume that short address of device is the same as used in the polling request. In addition to this, polling address can be read from HART 6 device with cmd #7	Polling Address	Unsigned 8	USINT	See [3] Chapter 6.8 Command 7 Read Loop Configuration
busProtocol	IdBusProtocol	Communication Channel shall pass "HART" in this attribute	HART	Enumeration: "HART" for HART5 and HART6	enumeration (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP)	
universalCommandRevisionLevel	IdBusProtocolVersion	Command #0 Byte 4	HART Revision	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
manufacturerIdentificationCode	IdManufacturer	For HART 7: 16 Bit Command #0 Byte 17+18 – Manufacturer Identification Code For HART 5 and 6: 8 bit Command #0 Byte 1- Manufacturer Identification Code	Manufacturer Identification Code	16 bit unsigned integer Example: Endress+Hauser: 17 (0x11)	UINT (dec)	See [3] Chapter 5.8 Manufacturer Identification Codes See [4] Manufacturer Identification Codes
deviceTypeID	IdTypeID	For HART 7: 16 Bit Command #0 Byte 1+2 – Manufacturer Identification Code For HART 5 and 6: 8 bit Command #0 Byte 2 – Manufacturers Device Type code	Device Type Code	16 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
softwareRevision	IdSoftwareRevision	Command #0 Byte 6	Software Revision	8 bit unsigned integer	USINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
hardwareRevision	IdHardwareRevision	Command #0 Byte 7 (Most significant 5 bits)	Hardware Revision	5 bit unsigned integer	INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	FDT data type (display format)	Specification reference
physicalSignalingCode	-	Command #0 Byte 7 (Least significant 3 bits).	Physical Signaling Code	3 bit unsigned enumerator	INT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
tag	IdTag	For HART >=6: Command #20 For HART <6: Command #13 Bytes 0 – 5	Tag	6 Bytes of 8 Packed ASCII characters	STRING	See [4] Chapter 6.20 – Command 20 Read Long Tag See [3] Chapter 6.13 Command 13 Read Tag, Descriptor, Date
longTag		For HART >=6: Command #20 For HART <6: Command #12 Bytes 0 – 23		32 Bytes char or 24 Bytes or 32 Packed ASCII characters	STRING	See [4] Chapter 6.20 – Command 20 Read Long Tag See [4] Chapter 6.12 – Command 12 Read Message
shortTag		Command #13 Bytes 0 – 5.		6 Bytes of 8 Packed ASCII characters	STRING	See [4] Chapter 6.13 – Command 13 Read Tag, Descriptor, Date
deviceID	IdSerialNumber	Command #0 Bytes 9 – 11	Device Identification Number	Unsigned 24	UDINT	See [3] Chapter 6.1 Command 0 Read Unique Identifier
ipAddress		host name or IP address conformant to IPv4 or IPv6 standard (additionally including the port number, if required) of a HART UDP or HART TCP device.			STRING	
port		port of a HART TCP or UDP device		64 bit unsigned	USINT (dec)	
N/A	IdDTMSupportLevel	Not applicable for scan / physical device. Attribute to be used only in DTMDeviceType identification. Enumeration: GenericDTM, ProfileDTM, BlockSpecificProfileDTM	DTM Support Level	-	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	-

Table 17 – Identification data types for 'Extended_HART' protocols without protocol independent semantics

IEC 61784 CPF 9 Attribute	Semantic element name	Data request in physical device	Protocol specific name	IEC 61784 CPF 9 data format	XML-FDT format (display format)	Specification reference
deviceCommandRevisionLevel	-	Command #0 Byte 5	Device Revision Level	8 bit unsigned integer	USINT (dec)	See [3] Chapter 6.1 Command 0 Read Unique Identifier
deviceFlag	-	Command #0 Byte 8	Flags	Bit value according Flag Assignment table. 8 bit – unsigned int	USINT (hex)	See [4] Chapter 5.11 Table Flag Assignments
manufacturerSpecificExtension		Can be used by DTM for vendor specific device identification information, for example by combining a number of device parameter values into one string value. This can be used to identify a specific device variant.			STRING	

Table 18 – Simple identification data types for ‘Extended_HART’ protocols with protocol independent semantics

Data type	Definition	Description
idDTMSupportLevel	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	enumeration genericSupport profileSupport blockspecificProfileSupport specificSupport
match	STRING	Used by Device DTM to define a regular expression which shall match to scanned physical define identification information
nomatch	STRING	Used by Device DTM to define a regular expression which shall not match to scanned physical define identification information. Used by Device DTM to indicate if identification information may not match
schemaVersion	STRING	Version number that is used by a Frame Application to identify an updated schema. The value for schemas redefined with this document shall be set to "1.3"
addressingMode	enumeration (shortAddress | shortTag | longTag)	With this attribute the Parent DTM defines which address property shall be used for the connection

Table 19 – Structured identification data types for ‘Extended_HART’ protocols with protocol independent semantics

Elements	Definition			Description
	Elementary data types	Usage	Multiplicity	
RegExpr	STRUCT			Includes regular expression string – either for match or for nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

12.6 Topology scan data types

This data type is used at Scan service response.

The data types describe one entry in the list of scanned devices (see Table 20).

The data types described in 12.6 are defined for the following namespace.
Namespace: fdthartdevice

Table 20 – Structured device type identification data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
HARTDevice	STRUCT			Definition of a IEC 61784 CPF 9 device concerning the scan response
	fdthart:LongAddress	O	[0..1]	
	fdthart:manufacturerId	O	[0..1]	
	fdthart:deviceTypeId	O	[0..1]	
	fdt:subDeviceType	O	[0..1]	
	fdt:tag	M	[1..1]	
	fdthart:shortAddress	O	[0..1]	

12.7 Scan identification data types for protocol ‘HART_Basic’

Subclause 12.7 defines data types that are used to provide protocol-specific scanning (see Table 21 and Table 22).

The data types described in 12.7 are used at following services: scan service.

The data types described in 12.7 are defined for the following namespace.
Namespace: hartscan

Table 21 – Simple scan identification data types for protocol ‘HART_Basic’

Data type	Definition	Description
resultState	enumeration (provisional | final | error)	Identifies if the result is one of the provisional results or the final result of the split scan results
configuredState	enumeration (configuredAndPhysicallyAvailable | | configuredAndNotPhysicallyAvailable | availableButNotConfigured | notApplicable)	A communication master shall indicate in this attribute, if the scan response is related to a detected physical device which is configured or unconfigured

Table 22 – Structured scan identification data types for protocol 'HART_Basic'

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdAddress	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix "Id" for better identification
	hartident:shortAddress	M	[1..1]	
IdBusProtocol	STRUCT			
	hartident:busProtocol	M	[1..1]	
IdBusProtocolVersion	STRUCT			
	hartident:universalCommandRevisionLevel	M	[1..1]	
IdManufacturer	STRUCT			
	hartident:manufacturerIdentificationCode	M	[1..1]	
IdTypeID	STRUCT			
	hartident:deviceTypeID	M	[1..1]	
IdSoftwareRevision	STRUCT			
	hartident:softwareRevision	M	[1..1]	
IdHardwareRevision	STRUCT			
	hartident:hardwareRevision	M	[1..1]	
IdTag	STRUCT			
	hartident:tag	M	[1..1]	
IdSerialNumber	STRUCT			
	hartident:deviceID	M	[1..1]	
DeviceCommandRevision Level	STRUCT			All elements without semantic prefix "Id" are transformed by XSL to name value pairs. These elements contain exactly one attribute defined in IEC 62453-2, each including one value of the physical device
	hartident:deviceCommandRevisionLevel	M	[1..1]	
DeviceFlag	STRUCT			
	hartident:deviceFlag	M	[1..1]	
ManufacturerSpecific-Extension	STRUCT			
	hartident:manufacturerSpecificExtension	M	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ScanIdentification	STRUCT			All IEC 61784 CPF 9 scan identification elements for one scanned physical device
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdTag	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	DeviceCommandRevisionLevel	M	[1..1]	
	DeviceFlag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentifications	STRUCT			Collection of ScanIdentification elements
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	

12.8 Scan identification data types for ‘Extended_HART’ protocols

Subclause 12.8 defines data types that are used to provide protocol-specific scanning (see Table 23 and Table 24).

The data types described in 12.8 are used at following services: scan service.

The data types described in 12.8 are defined for the following namespace.
Namespace: hartscan2

Table 23 – Simple scan identification data types for ‘Extended_HART’ protocols

Data type	Definition	Description
resultState	enumeration (provisional | final | error)	Identifies if the result is one of the provisional results or the final result of the split scan results
configuredState	enumeration (configuredAndPhysicallyAvailable | | configuredAndNotPhysicallyAvailable | availableButNotConfigured | notApplicable)	A communication master shall indicate in this attribute, if the scan response is related to a detected physical device which is configured or unconfigured

Table 24 – Structured scan identification data types for 'Extended_HART' protocols

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdAddress	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix "Id" for better identification
	hartident:shortAddress	M	[1..1]	
	hartident:shortTag	M	[1..1]	
	hartident:longTag	M	[1..1]	
	hartident:addressingMode	M	[1..1]	
	hartident:ipAddress	O	[1..1]	
	hartident:port	O	[1..1]	
	hartident:ipVersion	O	[1..1]	
	hartident:networkId	O	[1..1]	
	hartident:longAddressByte1	M	[1..1]	
	hartident:longAddressByte2	M	[1..1]	
	hartident:longAddressByte3	M	[1..1]	
	hartident:longAddressByte4	M	[1..1]	
	hartident:longAddressByte5	M	[1..1]	
IdBusProtocol	STRUCT			All elements contain exactly one attribute each including the value of the scanned physical device. All elements with semantic meaning have a prefix "Id" for better identification
	hartident:busProtocol	M	[1..1]	
IdBusProtocolVersion	STRUCT			
	hartident:universalCommandRevisionLevel	M	[1..1]	
IdManufacturer	STRUCT			
	hartident:manufacturerIdentificationCode	M	[1..1]	
IdTypeID	STRUCT			
	hartident:deviceTypeID	M	[1..1]	
IdSoftwareRevision	STRUCT			
	hartident:softwareRevision	M	[1..1]	
IdHardwareRevision	STRUCT			
	hartident:hardwareRevision	M	[1..1]	
IdTag	STRUCT			
	hartident:tag	M	[1..1]	
IdSerialNumber	STRUCT			
	hartident:deviceID	M	[1..1]	
DeviceCommandRevision Level	STRUCT			All elements without semantic prefix "Id" are transformed by XSL to name value pairs. These elements contain exactly one attribute defined in IEC 62453-2, each including one value of the physical device
	hartident:deviceCommandRevisionLevel	M	[1..1]	
DeviceFlag	STRUCT			
	hartident:deviceFlag	M	[1..1]	
ManufacturerSpecific-Extension	STRUCT			
	hartident:manufacturerSpecificExtension	M	[1..1]	

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ScanIdentification	STRUCT			All IEC 61784 CPF 9 scan identification elements for one scanned physical device
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdAddress	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdTag	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	DeviceCommandRevisionLevel	M	[1..1]	
	DeviceFlag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..1]	
ScanIdentifications	STRUCT			Collection of ScanIdentificati on elements
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	

12.9 Device type identification data types – provided by DTM

Subclause 12.9 defines data types that are used to provide protocol-specific information for device types (see Table 25).

The data types described in 12.9 are used in the following services:

- GetIdentificationInformation service.

The data types described in 12.9 are defined for the following namespace.
Namespace: hartdevtype

Table 25 – Structured device type identification data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdBusProtocol	STRUCT			All elements contain exactly one attribute, each including the value of the scanned physical device. All elements with semantic meaning have a prefix "Id" for better identification
	hartident:busProtocol	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdBusProtocolVersion	STRUCT			
	hartident:universalCommandRevisionLevel	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdManufacturer	STRUCT			
	hartident:manufacturerIdentificationCode	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdTypeID	STRUCT			
	hartident:deviceTypeID	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdSoftwareRevision	STRUCT			
	hartident:softwareRevision	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
IdHardwareRevision	STRUCT			
	hartident:hardwareRevision	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
DeviceCommand-RevisionLevel	STRUCT			All elements without semantic prefix "Id" are transformed by XSL to name value pairs. These elements contain exactly one attribute defined in IEC 62453-2 each including one value of the physical device
	hartident:deviceCommandRevisionLevel	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
DeviceFlag	STRUCT			
	hartident:deviceFlag	O	[0..1]	
	hartident:RegExpr	O	[0..*]	
ManufacturerSpecific-Extension	STRUCT			
	hartident:manufacturerSpecificExtension	M	[1..1]	
DeviceIdentification	STRUCT			IEC 61784 CPF 9 specific identification elements
	hartident:idDTMSupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	DeviceCommandRevisionLevel	M	[1..1]	
	DeviceFlag	M	[1..1]	
	ManufacturerSpecificExtension	O	[0..*]	
DeviceIdentifications	STRUCT			List of device identifications
	fdt:protocolId	M	[1..1]	
	DeviceIdentification	M	[1..*]	

Bibliography

[1] HCF_SPEC-99 Revision 9.0*, HCF Command Summary Specification*, July 2007

[2] HCF_SPEC-085 Revision 2.0, *HCF Network Management Specification*, June 2012

[3] HCF SPEC-127 Revision 7.1, *Universal Command Specification*, May 2008

[4] HCF SPEC-183 Revision 16.0*, Common Tables Specification*, 19 December 2005

[5] IEC TR 62453-42, *Field device tool (FDT) interface specification – Part 42: Object model integration profile – Common Language Infrastructure*

[6] ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*

____

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

____

SPÉCIFICATION DES INTERFACES DES OUTILS DES DISPOSITIFS DE TERRAIN (FDT) –

Partie 309: Intégration des profils de communication – CPF 9 de l'IEC 61784

AVANT-PROPOS

1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments du présent document de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

L'IEC 60317-84 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels. Il s'agit d'une norme internationale.

Cette troisième édition annule et remplace la seconde édition parue en 2016. Cette édition constitue une révision technique.

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

- corrections relatives à l'accès aux informations de l'équipement concerné et;
- corrections relatives à la description de la prise en charge des différentes versions de protocole.

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

Projet	Rapport de vote
65E/907/FDIS	65E/936/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/publications/.

Chaque partie de la série IEC 62453-3xy doit être utilisée conjointement avec l'IEC 62453- 2.

Une liste de toutes les parties de la série IEC 62453, publiées sous le titre général *Spécification des interfaces des outils des dispositifs de terrain (FDT),* peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

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

IMPORTANT – Le logo 'colour inside' qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La présente partie de l'IEC 62453 constitue une spécification d'interface pour les développeurs des composants des outils des dispositifs de terrain (FDT[1] - *field device tool*) afin de prendre en charge le contrôle de fonction et l'accès aux données dans une architecture client/serveur. La spécification résulte d'un processus d'analyse et de conception destiné à réaliser des interfaces normalisées et permettre ainsi à de nombreux fournisseurs de développer des serveurs et des clients dans le cadre d'une interaction ininterrompue répondant à leur besoin.

L'intégration de bus de terrain dans les systèmes de commande nécessite d'effectuer quelques tâches supplémentaires. Outre les outils spécifiques à un bus de terrain et aux dispositifs, l'intégration de ces outils dans des outils d'ingénierie ou de planification à l'échelle d'un système de plus haut niveau s'avère nécessaire. La définition claire des interfaces d'ingénierie faciles à utiliser pour tous les outils concernés revêt une grande importance, en particulier, pour une utilisation dans des systèmes de commande importants et hétérogènes, généralement dans le domaine de l'industrie de transformation.

Un composant logiciel spécifique à un équipement, appelé gestionnaire de type d'équipement (DTM - *device type manager*) est fourni par le fabricant de dispositifs de terrain avec son équipement. Le DTM est intégré dans des outils d'ingénierie par l'intermédiaire des interfaces FDT définies dans la présente spécification. L'approche d'intégration s'applique en général à tous les types de bus de terrain et satisfait ainsi aux exigences relatives à l'intégration de différents types d'équipements dans des systèmes de commande hétérogènes.

La Figure 1 représente l'alignement de l'IEC 62453-309 dans la structure de la série IEC 62453.

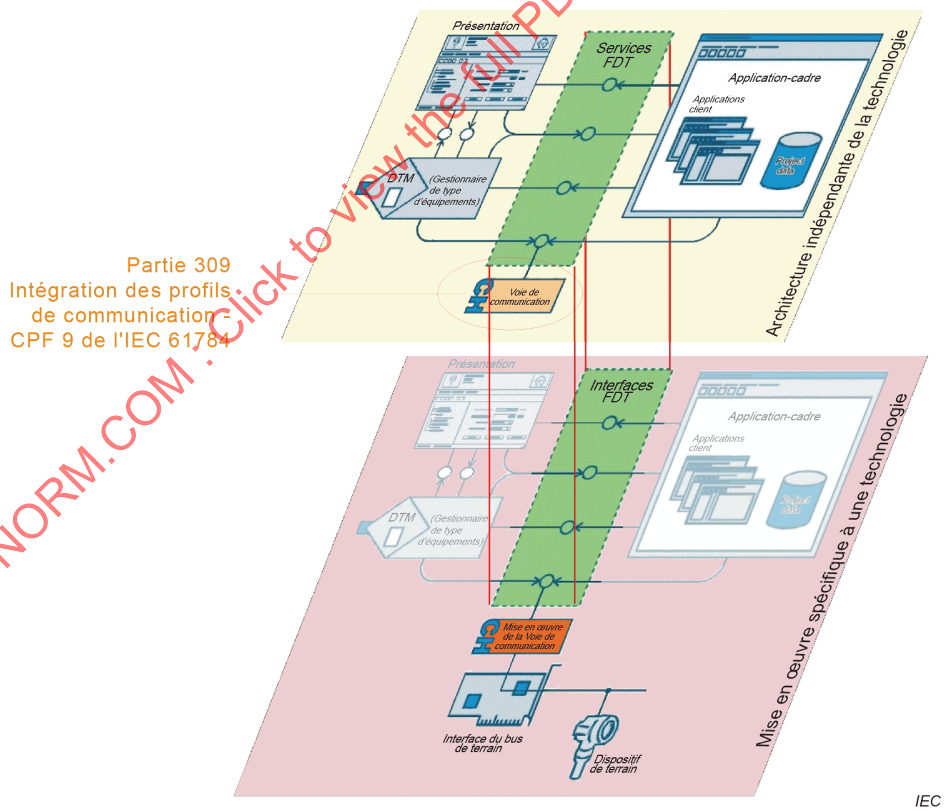


Figure 1 – Partie 309 de la série IEC 62453

[1] FDT® est l'appellation commerciale des produits fournis par FDT Group AISBL. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit cité. Des produits équivalents peuvent être utilisés à condition qu'ils génèrent les mêmes résultats.

SPÉCIFICATION DES INTERFACES DES OUTILS DES DISPOSITIFS DE TERRAIN (FDT) –

Partie 309: Intégration des profils de communication – CPF 9 de l'IEC 61784

1 Domaine d'application

La Famille de Profils de Communication 9 (communément appelée HART®[2]) définit les profils de communication fondés sur l'IEC 61158-5-20 et l'IEC 61158-6-20. Le profil de base CP 9/1 est défini dans l'IEC 61784-1.

La présente partie de l'IEC 62453 fournit des informations sur l'intégration de la technologie HART® dans la norme des outils des dispositifs de terrain (FDT) (IEC 62453-2).

La présente partie de l'IEC 62453 spécifie les services de communication et autres services.

Le présent document ne contient pas la spécification des outils FDT, ni ne la modifie.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61158-5-20, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-20: Définition des services de la couche application – Éléments de type 20*

IEC 61158-6-20, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-20: Spécification du protocole de la couche application – Éléments de type 20*

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

IEC 62453-1:–[3], *Spécification des interfaces des outils des dispositifs de terrain (FDT) – Partie 1: Vue d'ensemble et guide*

IEC 62453-2:–[3], *Spécification des interfaces des outils des dispositifs de terrain (FDT) – Partie 2: Concepts et description détaillée*

2 HART® et WirelessHART® sont les appellations commerciales des produits fournis par FieldComm Group. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit cité. Des produits équivalents peuvent être utilisés à condition qu'ils génèrent les mêmes résultats.

3 En cours d'élaboration. Stades respectifs au moment de la publication: IEC/CCDV 62453-1:2022 et IEC/RFDIS 62453-2:2022.

3 Termes, définitions, symboles, termes abrégés et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 62453-1 et l'IEC 62453-2, ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

3.1.1
mode salve
mode dans lequel le dispositif de terrain génère des télégrammes de réponse sans télégramme de demande de la part du maître

3.2 Termes abrégés

Pour les besoins du présent document, les abréviations données dans l'IEC 62453-1 et l'IEC 62453-2, ainsi que les suivantes s'appliquent.

BACK	Burst ACKnowledge (Accusé de réception en salves)
C8PSK	Coherent 8-way Phase Shift Keying (Modulation par déplacement de phase à 8 voies cohérente), couche de communication HART telle que définie dans la HCF_SPEC-60, Révision 1.0
DR	Delayed Response (Réponse différée)
EDD	Electronic Device Description (Description d'équipement électronique)
FSK	Frequency Shift Keying (Modulation par déplacement de fréquence), couche de communication HART telle que définie dans la HCF_SPEC-54, Révision 8.1
HART	Highway Addressable Remote Transducer (Transducteur distant d'autoroute adressable)

3.3 Conventions

3.3.1 Dénominations des types de données et références aux types de données

Les conventions pour la dénomination et le référencement des types de données sont décrites à l'Article A.1 de l'IEC 62453-2:–.

3.3.2 Vocabulaire relatif aux exigences

Les expressions suivantes sont utilisées pour spécifier des exigences.

Utilisation de "doit" ou "obligatoire"	Aucune exception tolérée.
Utilisation de "il convient de" ou "recommandé"	Forte recommandation. Il peut être légitime, dans des cas particuliers exceptionnels, de s'écarter du comportement décrit.
Utilisation de "peut" ou "facultatif"	La fonction ou le comportement peut être donné, selon des conditions définies.

3.3.3 Utilisation de la notation UML

Les figures du présent document utilisent la notation UML telle que définie à l'Annexe A de l'IEC 62453-1:–.

4 Catégorie de bus

Le protocole CPF 9 de l'IEC 61784 est identifié dans l'élément protocolId du type structuré de données 'fdt:BusCategory' par les identificateurs uniques suivants (voir le Tableau 1):

Tableau 1 – Identificateurs du protocole

Valeur d'identificateur	ProtocolId	Chaîne à l'affichage	Description
036D1498-387B-11D4-86E1-00E0987270B9	HART_Basic	'HART'	Prise en charge du protocole CPF 9 de l'IEC 61784 sur communication FSK avec fonctionnalité de base (déprécié)
98503B8F-0FFB-4EB7-BB67-F4D6BD16DB8D	HART_FSK	'HART FSK'	Prise en charge du protocole HART sur communication FSK avec fonctionnalité complète.
74D29D22-F752-40EF-A747-ACA72C791155	HART_Wireless	'HART Wireless' (HART sans fil)	Prise en charge du protocole WirelessHART
58001A08-C178-4A59-A76B-9EF9111CB83D	HART_RS485	'HART RS485'	Prise en charge du protocole HART sur communication RS485
EF708CB7-A2A1-42AF-890C-15CEB680CC12	HART_Infrared	'HART Infrared' (HART infrarouge)	Prise en charge du protocole HART sur communication Infrared (infrarouge)
D122D172-F0C7-4B03-965B-512CD4C0871E	HART_IP	'HART IP'	Prise en charge de HART sur protocole IP

Le protocole 'HART_Basic' est maintenu pour la rétrocompatibilité seulement (par exemple, pour l'interaction avec les DTM conformes à l'IEC 62453-309:2009). Les autres identificateurs de protocole fournissent une meilleure prise en charge pour la planification des topologies de réseaux et pour l'établissement de connexions entre le DTM et l'équipement correspondant. Pour les DTM conformes au présent document, la prise en charge de l'un des autres protocoles est obligatoire.

Dans le présent document, les autres protocoles (HART_FSK, HART_Wireless, HART_RS485, HART_Infrared, HART_IP) sont référencés comme étant des protocoles 'Extended_HART'. (Par exemple, pour les définitions qui s'appliquent à tous les protocoles à l'exception de 'HART_Basic'.)

Le Tableau 2 définit quelle PhysicalLayer peut être utilisée conjointement avec la BusCategory définie dans le Tableau 1.

Tableau 2 – Définition de PhysicalLayer

Valeur de l'identificateur de PhysicalLayer	Valeur du nom de PhysicalLayer	Description
BAB2091A-C0A7-4614-B9DE-FCC2709DCF5D	Couche physique HART FSK	Prise en charge de la couche physique HART FSK
B9F1A250-AC94-4487-8F25-A8F3F8F89DC5	Couche physique WirelessHART	Prise en charge de la couche physique WirelessHART
036D1591-387B-11D4-86E1-00E0987270B9	Couche physique HART RS-485	Prise en charge des équipements HART utilisant la communication RS-485
AE4119EF-B9FD-429c-B244-134DB182296A	Couche physique HART infrared (HART infrarouge)	Prise en charge des équipements HART utilisant la communication infrarouge
307dd808-c010-11db-90e7-0002b3ecdcbe	10BASET	Couches physiques fondées sur HART Ethernet
307dd809-c010-11db-90e7-0002b3ecdcbe	10BASETXHD	
307dd80a-c010-11db-90e7-0002b3ecdcbe	10BASETXFD	
307dd80b-c010-11db-90e7-0002b3ecdcbe	10BASEFLHD	
307dd80c-c010-11db-90e7-0002b3ecdcbe	10BASEFLFD	
307dd80d-c010-11db-90e7-0002b3ecdcbe	10BASEFXHD	
307dd80e-c010-11db-90e7-0002b3ecdcbe	10BASEFXFD	
307dd80f-c010-11db-90e7-0002b3ecdcbe	100BASETXHD	
307dd810-c010-11db-90e7-0002b3ecdcbe	100BASETXFD	
307dd811-c010-11db-90e7-0002b3ecdcbe	100BASEFXHD	
307dd812-c010-11db-90e7-0002b3ecdcbe	100BASEFXFD	
307dd813-c010-11db-90e7-0002b3ecdcbe	100BASELX10	
307dd814-c010-11db-90e7-0002b3ecdcbe	100BASEPX10	
307dd815-c010-11db-90e7-0002b3ecdcbe	1000BASEXHD	
307dd816-c010-11db-90e7-0002b3ecdcbe	1000BASEXFD	
307dd817-c010-11db-90e7-0002b3ecdcbe	1000BASELXHD	
307dd818-c010-11db-90e7-0002b3ecdcbe	1000BASELXFD	
307dd819-c010-11db-90e7-0002b3ecdcbe	1000BASESXHD	
307dd81a-c010-11db-90e7-0002b3ecdcbe	1000BASESXFD	
307dd81b-c010-11db-90e7-0002b3ecdcbe	1000BASETHD	
307dd81c-c010-11db-90e7-0002b3ecdcbe	1000BASETFD	
307dd81d-c010-11db-90e7-0002b3ecdcbe	10GigBASEFX	

La BusCategory constitue l'information significative pour la planification de la topologie. La PhysicalLayer (qui est fournie dans le type de données BusInformation) ne doit être utilisée que pour les informations complémentaires.

La propriété DataLinkLayer n'est pas applicable pour HART et doit être mise à "null".

5 Accès aux données d'instance et d'équipement

5.1 Généralités

La sémantique définie du protocole HART permet une vaste plage pour l'identification de variables d'équipements et de paramètres d'équipements. La plupart de ces informations sémantiques sont définies dans les bibliothèques normalisées d'importation des EDD.

L'Article 5 décrit comment les informations sémantiques définies dans le protocole HART doivent être utilisées pour exporter les données d'équipements, les données d'instance et les données de processus.

5.2 Objets Voies de processus (Process Channels) fournis par le DTM

L'ensemble minimal de données fournies doit être:

les quatre premières valeurs fournies relatives à un processus (PV, SV, ...) – si disponibles – sont modélisées comme références de la voie. La voie référencée doit inclure les plages et les changements d'échelle.

Un équipement HART communique les données de processus soit par ses voies analogiques, soit par le biais d'informations numériques (par demande ou par mode salve, par exemple). Les voies analogiques étant toujours liées à une variable dynamique, telle que spécifiée dans la référence [1][4] Chapitre 8, la description d'une voie analogique doit donc être accessible en utilisant la variable dynamique correspondante (par exemple, les attributs de la variable dynamique PV décrivent toujours la première voie analogique).

HART établit une distinction entre trois méthodes d'accès aux signaux numériques:

1) Accès à une valeur analogique et aux variables dynamiques assignées (Command #3)

   Les signaux E/S peuvent être assignés à l'une des quatre variables dynamiques PV, SV, TV et QV. La commande #3 permet de lire la valeur analogique et les variables dynamiques sans connaissances spécifiques de l'équipement.

2) Accès par indice aux variables d'équipements (Commande #33)

   Toutes les valeurs des variables d'équipements et leurs unités peuvent être lues en utilisant les informations relatives au code de variable d'équipement dans la commande #33.

3) Accès par indice à la classification et au statut de variable d'équipements (Commande #9)

   La commande #9 fournit plus d'informations que la commande #33. Outre la valeur et l'unité, une classification et le statut des variables peuvent également être déterminés.

L'auteur de la commande détermine au moyen de la spécification HART les commandes qui sont utilisées.

5.3 Services du DTM pour accéder aux données d'instance et aux données d'équipement

Les services InstanceDataInformation et DeviceDataInformation doivent fournir un accès au minimum à l'ensemble des paramètres des commandes Universal (Universelle) et Common Practice (Pratique courante) (dans la mesure où l'équipement prend en charge la fonction).

De plus, les Octets 0 et 1 de la réponse pour chacune des commandes doivent être présentés.

Les services InstanceDataInformation et DeviceDataInformation peuvent également fournir un accès aux paramètres spécifiques à un équipement (par exemple, information de diagnostic).

6 Comportement spécifique à un protocole

6.1 Vue d'ensemble

Il n'existe qu'une seule séquence spécifique à un protocole définie pour la CPF 9 de l'IEC 61784:abonnement au mode salve.

4 Les chiffres entre crochets se réfèrent à la Bibliographie.

Cette séquence explique la manière dont la séquence "Transfert de données initié par l'équipement", définie dans l'IEC 62453-2, s'applique dans le contexte des télégrammes en salves tels que définis dans la CPF 9 de l'IEC 61784.

En outre, l'Article 6 fournit des informations concernant:

- l'utilisation des informations d'adressage de l'équipement;
- la prise en charge des codes de commande étendus;
- le traitement des défaillances de communication;
- le traitement des réponses différées; et
- la gestion des topologies physiques.

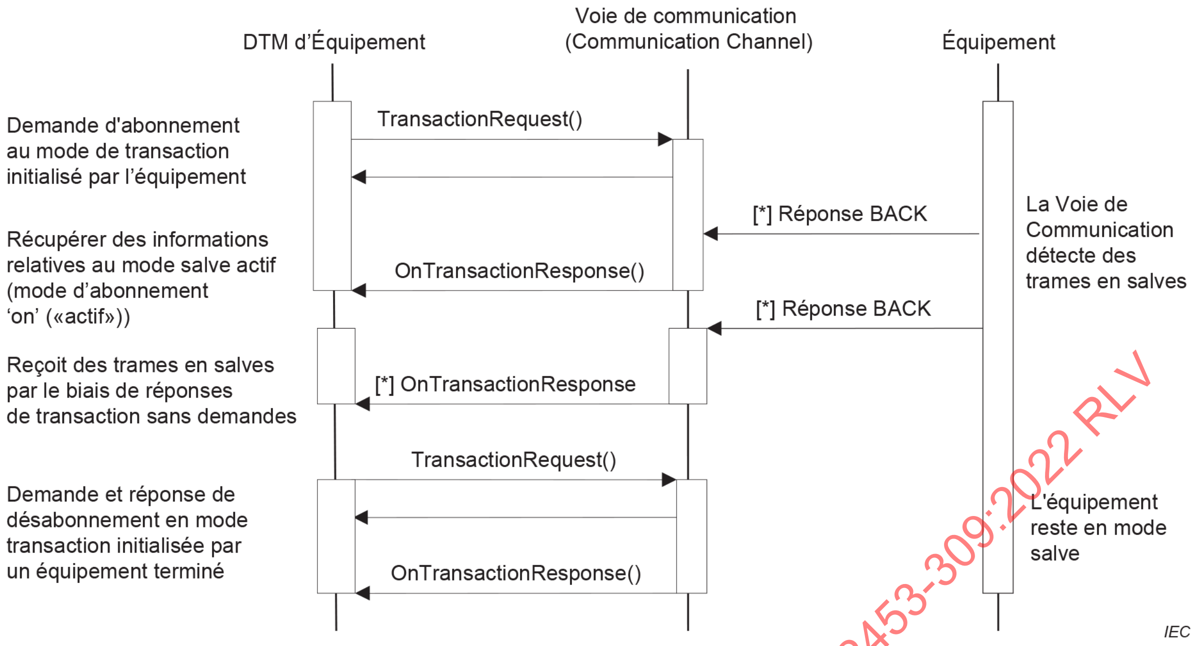
6.2 Abonnement au mode salve

Un abonnement au transfert de données initié par un équipement peut être demandé en envoyant une demande de transaction avec le contenu SubscribeRequest (voir la Figure 2). La Voie de Communication (Communication Channel) peut détecter si l'équipement est déjà en mode salve.

NOTE Dans HART 5, la détection ne peut s'effectuer que lorsque les trames en salves proviennent de l'équipement. Dans HART 6, le mode salve peut être détecté à l'aide de la commande #105.

La Voie de Communication répond à une SubscribeRequest par un contenu SubscribeResponse. Si des trames en salves sont reçues, l'équipement est en mode salve et la valeur burstModeDetected est mise sur TRUE (VRAI). Cette disposition signifie que le DTM d'Équipement commence à recevoir des messages en salves par le biais du mécanisme de réponse de transaction. Lorsqu'aucun message en salve n'a été reçu, la valeur burstModeDetected est mise sur FALSE (FAUX). Il incombe au DTM d'Équipement de régler ce dernier en mode salve. Le DTM d'Équipement peut ensuite appeler une demande de transaction, une nouvelle fois avec le contenu SubscribeRequest, afin de recevoir des messages en salves.

Pour se désabonner, le DTM d'Équipement envoie une demande de transaction avec une UnsubcribeRequest. La Voie de Communication répond par une UnSubscribeResponse avec la valeur burstModeDetected mise sur FALSE. Le DTM d'Équipement ne reçoit pas d'autres informations en salves par le biais du mécanisme de réponse de transaction. La Voie de Communication ne désactive pas le mode salve de l'équipement. Le DTM d'Équipement peut activer ou désactiver le mode salve en utilisant les demandes de transaction normale (commande #109). Cette disposition est indépendante de l'abonnement.



NOTE BACK signifie Burst ACKnowledge (accusé de réception en salves).

Figure 2 – Abonnement au mode salve

6.3 Utilisation des informations d'adressage de l'équipement

HART est un protocole maître/esclave sans connexion. Les demandes de transaction sont toujours traitées en utilisant des informations relatives à l'adresse d'équipement unique (un entier de 5 octets), désignée "adresse longue".

L'adressage d'équipement dans HART est donc principalement axé sur la détermination de cette adresse longue.

Actuellement, il existe trois modes de détermination possibles de l'adresse longue.

1) adresse courte

   L'adresse courte est un nombre compris entre 0 et 63 (0 à 15 uniquement pour la version 5 de HART). Dans le contexte d'une connexion directe à l'équipement, l'adresse courte est unique et permet de lire l'adresse longue en utilisant la commande #0.

2) marqueur court

   La commande #11 permet de demander les informations relatives à l'adresse longue pour un équipement avec un marqueur court spécifique. De telles demandes sont notamment utilisées pour les installations avec un grand nombre d'équipements HART connectés. Tous les équipements de multiplexeurs HART et autres structures de communication HART doivent prendre en charge cette commande.

3) marqueur long

   Le marqueur long a été introduit à partir de la version 6 de HART. Avec ses 32 caractères, il en compte 24 de plus que le marqueur court, ce qui permet de proposer un plus grand nombre d'options d'étiquetage de l'équipement. Pour les équipements d'une version HART antérieure à la version 6, un message est utilisé à la place du marqueur long. La commande #21 permet de demander les informations relatives à l'adresse longue pour un équipement avec un marqueur long spécifique. La commande #21 est habituellement prise en charge par des équipements fortement modulaires ou des passerelles (gateways).

Un DTM d'Équipement est chargé de fournir et de stocker toutes les informations utilisées pour résoudre l'adresse longue d'un équipement connecté. Il doit donc tenir à jour les données pour les trois méthodes de résolution d'adresse. Le DTM chargé de se connecter au matériel de communication doit sélectionner la méthode et fournir à l'utilisateur les moyens de saisir les informations relatives à l'adresse.

Outre la question de l'adressage, il existe également différentes approches pour l'identification du fabricant et du type d'équipement selon la version de HART prise en charge. Les versions HART jusqu'à HART 6 utilisent des valeurs d'un seul octet. Les versions HART à partir de HART 7 (et plus récentes) utilisent une valeur de deux octets. Les valeurs de deux octets sont également stockées dans les types de données décrits en 9.4.

Un DTM de Communication utilise les informations d'adressage fournies par le DTM d'Équipement pour résoudre l'adresse longue de la manière décrite ci-dessus.

6.4 Numéros de commande étendue

Le numéro de commande HART est défini comme étant un entier non signé d'un seul octet. À partir de la version 6 de HART, un format de numéro de commande étendue, utilisant deux octets au lieu d'un, a été défini pour permettre plus de 255 numéros de commande. Ce format utilise le numéro de commande 31, qui était auparavant réservé, pour indiquer que la demande utilise le format de numéro de commande étendue.

Selon la spécification définie en 8.1.2 de la référence [2], les commandes étendues sont mises en œuvre avec la commande #31 en utilisant le numéro de commande étendue comme étant les deux premiers octets dans la section de demande et de réponse.

Dans les outils FDT, toutes les commandes avec des numéros de commande étendue doivent être mises en œuvre à l'aide de la commande #31.

6.5 Traitement des défaillances et temporisations de communication

HART utilise un traitement d'erreurs de communication spécifique à un équipement. Le protocole définit une section dans la trame de réponse qui peut comporter les informations relatives aux défaillances de communication.

Si, au cours de l'exécution d'une demande de communication envoyée à une Voie de Communication, une erreur de communication se produit sur les couches physiques HART (ce qui inclut également les temporisations), aucun message Arrêt prématuré (Abort) ne doit être envoyé au DTM Enfant. La demande de transaction doit toutefois recevoir une réponse comportant un ensemble de données qui décrit l'erreur de communication telle que définie dans HART [1].

Dans le cas d'une telle défaillance de communication, le DTM d'Équipement est chargé de traiter l'erreur pour pallier la défaillance de communication.

La Voie de Communication doit envoyer un signal Arrêt prématuré au DTM d'Équipement uniquement dans le cas d'une rupture de communication établie par une connexion (par exemple, une connexion Ethernet à un modem HART).

6.6 Traitement des réponses différées

HART définit des contraintes de temps strictes pour les réponses à une demande dans le cadre d'une transaction HART. Si un équipement est incapable de satisfaire aux contraintes de temps, il peut lancer une séquence de réponses différées (DR - *delayed response*). Afin de prendre en charge le traitement des DR dans le cadre d'une communication imbriquée, le Paragraphe 6.6 définit le traitement dans le cadre des outils FDT.

Le DTM qui représente l'équipement est chargé de traiter les réponses DR issues de ce même équipement. Le DTM de Communication et les DTM de Passerelle (s'ils sont utilisés) doivent veiller à ce que les réponses DR soient correctement communiquées au DTM concerné. La Figure 3 donne un exemple d'un tel traitement de réponses différées.

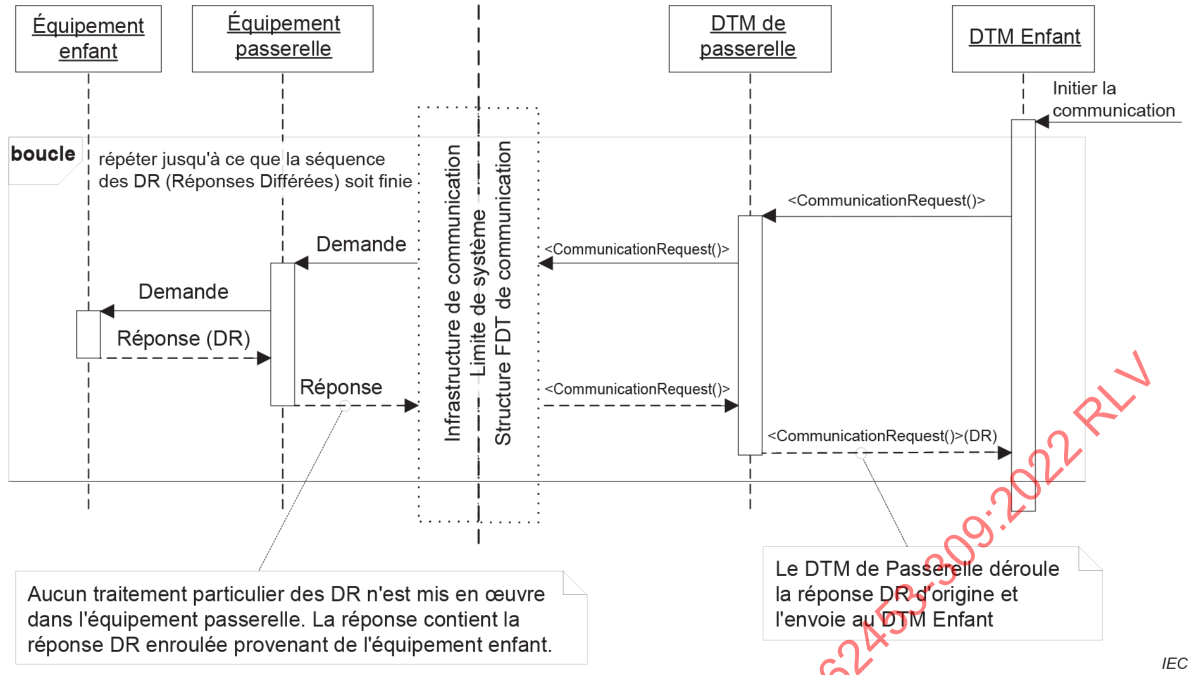


Figure 3 – Traitement des réponses différées (scénario 1)

Il est également possible que deux partenaires d'une séquence de DR soient les deux équipements à la fois. Par exemple, un équipement passerelle (par exemple, Passerelle WirelessHART) peut exécuter une séquence de réponses différées avec un équipement enfant (par exemple, un adaptateur WirelessHART). Dans ce cas, l'équipement passerelle est chargé de traiter la DR de l'équipement enfant. Les réponses différées n'atteignent pas le DTM Enfant respectif. Si l'équipement passerelle est incapable de traiter directement les DR, il peut les envoyer au DTM de Passerelle. Dans ce cas, les DR doivent être traitées par le DTM de Passerelle respectif. Habituellement, le concept de communication imbriquée représente l'interaction entre les équipements. Dans le cas présent, cette interaction n'est pas possible et la mise en œuvre doit suivre la séquence représentée à la Figure 4.

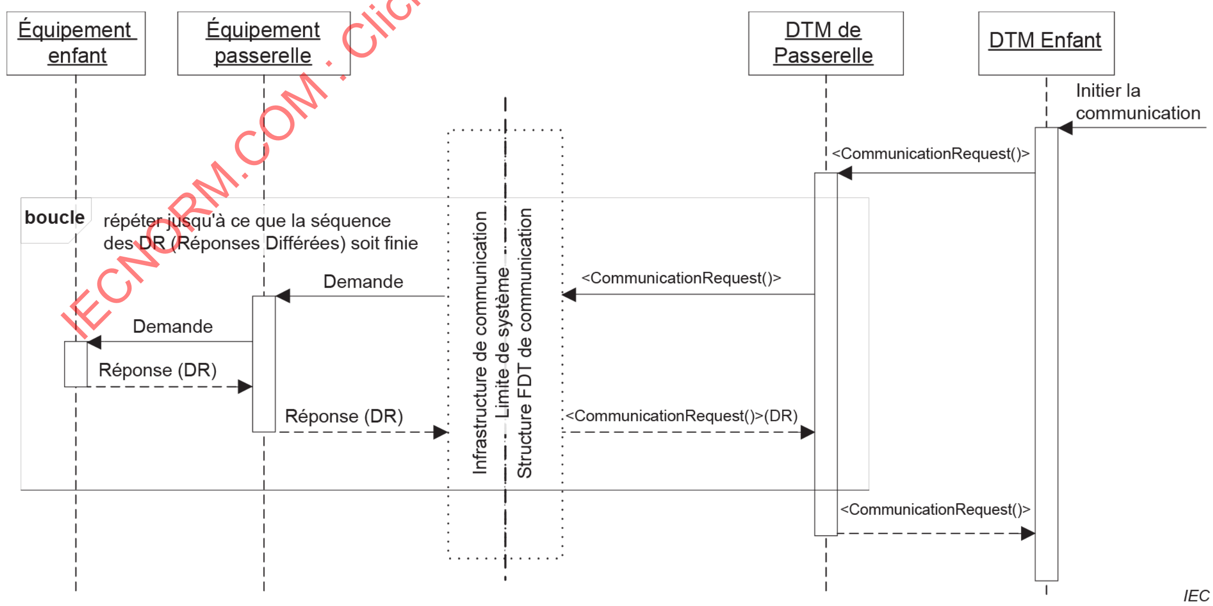


Figure 4 – Traitement des réponses différées (scénario 2)

Une durée longue d'une séquence de DR peut perturber l'utilisation de l'Application-Cadre FDT et bloquer l'interaction utilisateur. Il n'existe pas de définition de durée de temporisation pour les séquences de DR et ni le DTM lui-même ni tout autre DTM dans la chaîne de communication imbriquée ne sont capables de déclencher une temporisation susceptible de récupérer le système. La durée de temporisation dans ce cas dépend de l'application et doit être configurée par l'utilisateur. Lorsque la durée d'une séquence de DR n'est pas raisonnable, l'objectif doit consister à impliquer l'utilisateur. Si une séquence de DR est utilisée dans une interface utilisateur, des mécanismes de temporisation configurables doivent être mis en œuvre.

Pour traiter les réponses DR avec une interopérabilité fiable, les règles suivantes doivent être respectées:

- le DTM d'un équipement susceptible d'envoyer des réponses DR doit traiter les réponses DR de l'équipement;
- les réponses DR qui ne sont pas traitées par d'autres équipements doivent être transmises au DTM qui représente l'équipement qui envoie les réponses;
- un DTM doit savoir qu'il ne reçoit pas de réponses DR provenant de l'équipement, lorsque les réponses DR sont traitées par l'équipement parent;
- un DTM qui traite les réponses DR doit mettre en œuvre une gestion des temporisations configurables par l'utilisateur qui doit lui permettre d'établir une temporisation.

6.7 Topologies avec des protocoles HART mixtes

6.7.1 Généralités

Les DTM HART utilisant des protocoles ‘Extended_HART’ peuvent également prendre en charge le protocole ‘HART_Basic’, afin d'assurer la compatibilité avec les DTM HART existants.

Les protocoles ‘Extended_HART’ ont été définis pour mieux différencier les différents types de communication HART. L'utilisation simultanée des protocoles ‘Extended_HART’ et ‘HART_Basic’ nécessite des processus bien définis pour garantir l'interopérabilité.

6.7.2 Comportement des DTM prenant uniquement en charge le protocole ‘Extended_HART’

L'Application-Cadre (référence) valide la topologie. Si la Voie de Communication reçoit un appel à ValidateAddChild(), elle doit vérifier si le type d'équipement donné exige un protocolId approprié.

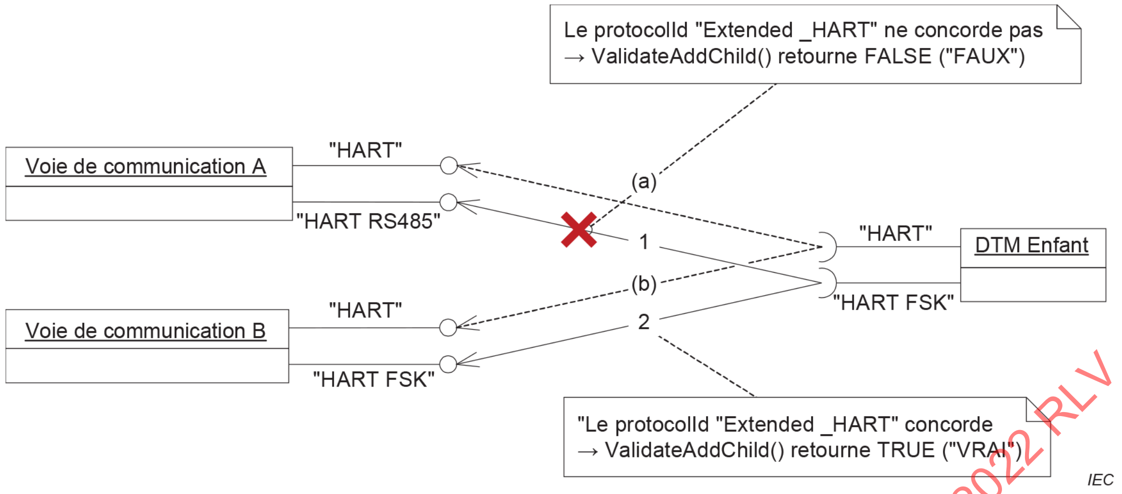
Le comportement d'un tel DTM dans un appel de ValidateAddChild est le suivant:

- si une concordance ne peut pas être établie, l'appel de ValidateAddChild () doit recevoir la réponse "FALSE";
- si une concordance a été établie, l'appel de ValidateAddChild () doit recevoir la réponse "TRUE". Au cours de l'appel à OnAddChild(), le DTM Parent définit l'attribut 'activeProtolID' du DTM Enfant sur le protocolId actuel.

6.7.3 Comportement des DTM prenant en charge le protocole ‘Extended_HART’ et le protocole ‘HART_Basic’

Lors de la création des topologies, une Application-Cadre doit vérifier la compatibilité de communication d'un DTM Enfant et d'un DTM Parent en comparant les listes de protocoles pris en charge et exigés. Les protocoles ‘Extended_HART’ permettent de réaliser une validation plus efficace des topologies. Cependant, si les deux DTM prennent également en charge le protocole ‘HART_Basic’, ceci peut donner lieu à des topologies non valables.

Lorsque, par exemple, une Voie de Communication, qui prend en charge ‘HART_RS485’ et ‘HART_Basic’, et un DTM d'Équipement, qui exige ‘HART_FSK’ et ‘HART_Basic’, sont connectés (voir la Voie de Communication A à la Figure 5), une Application-Cadre autorise la connexion de ces DTM en raison du protocolId ‘HART_Basic’ concordant. Il s'agit en fait d'une topologie qui n'est pas valable.



L'Application-Cadre autorise la liaison du DTM Enfant aux deux Voies de Communication, car au moins (a) et (b) sont possibles. Mais la discordance en (1) permet à la 'Voie de Communication A' de détecter la discordance et de refuser la liaison du 'DTM Enfant'.

Figure 5 – Comportement des DTM prenant en charge le protocole 'Extended_HART'et le protocole 'HART_Basic'

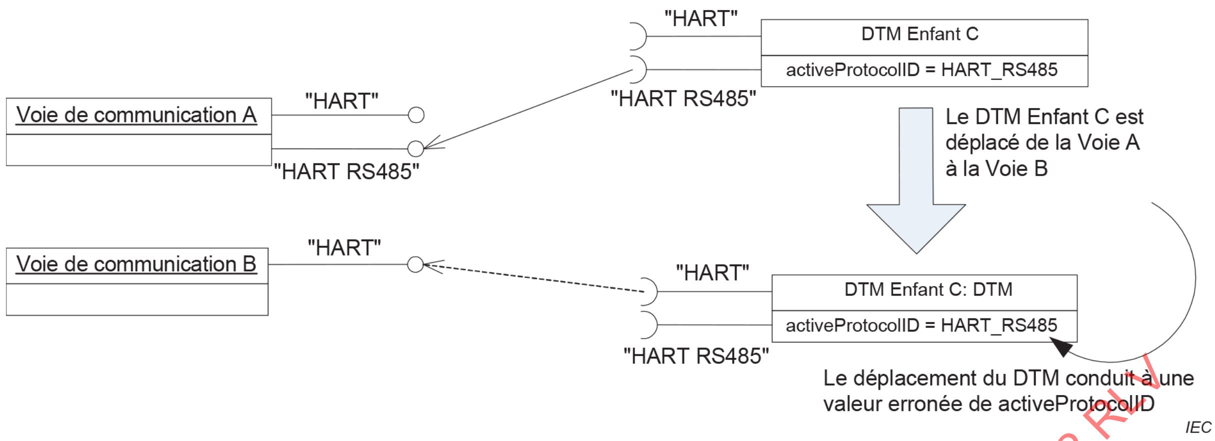
Pour éviter une telle situation, une Voie de Communication prenant en charge des protocoles 'Extended_HART' et le protocole 'HART_Basic' doit vérifier pendant un appel de ValidateAddChild() si un DTM connecté exige également le protocole 'Extended_HART' et le protocole 'HART_Basic'. La Voie de Communication doit également vérifier s'il existe un protocole 'Extended_HART' concordant.

- Si aucune concordance ne peut être établie dans les catégories de bus 'Extended_HART', la Voie de Communication doit répondre par "FALSE" à l'appel de ValidateAddChild().
- Si une concordance peut être établie, l'appel de ValidateAddChild() doit recevoir la réponse "TRUE". Au cours de l'appel de OnAddChild(), le Voie de Communication définit l'attribut "activeProtocolID" dans le DTM Enfant.

6.7.4 Comportement des DTM qui exigent le protocole 'Extended_HART' ou le protocole 'HART_Basic'

Avec l'attribut 'activeProtocolID', un DTM est informé du type de connexion actuel dans la topologie. Cette procédure peut cependant échouer lorsque le DTM est connecté à une Voie de Communication qui ne met pas en œuvre la gestion de 'activeProtocolID'.

Établir l'hypothèse selon laquelle, par exemple, un DTM Enfant était connecté à la Voie de Communication en utilisant 'HART_RS485' avec pour résultat de définir l'attribut 'activeProtocolID' sur 'HART_RS485' (voir la Figure 6). L'Application-Cadre déplace maintenant le DTM Enfant vers une Voie de Communication qui ne prend en charge que le protocole 'HART_Basic'. Lorsque le DTM Enfant tente alors d'établir une connexion 'HART_RS485', cela peut conduire à une erreur.



Le déplacement du DTM C de la Voie A vers le Voie B conduit à une valeur incohérente de l'attribut "activeProtocolID", car le DTM B ne prend pas en charge le protocole "Enhanced_HART" et ne définit donc pas l'attribut activeProtocolID.
→ Avant que le DTM C n'exécute une demande de connexion, il doit valider si le DTM Parent établit ou non la communication relative à la valeur actuelle de l'attribut activeProtoclID.

Figure 6 – Comportement des DTM qui exigent le protocole ‘Extended_HART’ ou le protocole ‘HART_Basic’

Par conséquent, il est nécessaire qu'un DTM qui exige à la fois le protocole ‘Extended_HART’ et le protocole ‘HART_Basic’ vérifie les capacités de la Voie de Communication avant d'établir une connexion.

6.8 Communication imbriquée comportant plusieurs passerelles

Le protocole HART prend en charge des topologies dans le réseau physique qui permettent d'avoir plusieurs passerelles dans une chaîne de communication. Des équipements HART câblés connectés à un adaptateur sans fil en communication avec une passerelle sans fil constituent un exemple de cette topologie (voir 6.9).

Le concept général de communication imbriquée veut qu'un équipement reçoive les données de commande qui ont été générées par son DTM respectif et que le DTM reçoive les données de réponse de son équipement respectif. La communication imbriquée exige également que le DTM Enfant soit toujours l'expéditeur actif et qu'il ne soit donc pas autorisé à transmettre sans encapsulation ou transformation une communication envoyée par son DTM Enfant.

La commande #77 (envoi à sous-équipement) permet au protocole HART de définir un mécanisme normalisé d'encapsulation pour transmettre la communication par une topologie de réseau. Chaque demande envoyée à un sous-équipement doit être encapsulée dans une demande de commande #77 avant de la transmettre à l'équipement passerelle. Lorsqu'une réponse à une commande #77 est retournée, le DTM de Passerelle doit décompacter cette commande et envoyer les données de réponse qu'elle contient au DTM Enfant respectif.

Une commande #77 peut être restructurée en une autre structure de commande selon la mise en œuvre dans la passerelle. Dans ce cas, le DTM de Passerelle est chargé de transformer les demandes de commande #77 entrantes provenant du DTM Enfant en commandes spécifiques à une passerelle. Il est également chargé de restructurer les réponses obtenues pour leur rendre leur structure de réponses sur la demande de commande #77 reçue à l'origine.

6.9 Structures des communications et des réseaux dans WirelessHART

6.9.1 Généralités

WirelessHART définit un protocole riche et sécurisé entre des équipements utilisant la technologie sans fil à 2,4 GHz. Les systèmes hôtes ne sont pas censés interagir directement avec le réseau WirelessHART. Un système hôte peut, en utilisant un équipement passerelle

WirelessHART, communiquer avec tout équipement WirelessHART par le biais des transactions maître/esclave de HART, sans nécessiter de connaissances spécifiques du protocole WirelessHART.

Le protocole HART spécifie trois types normalisés d'équipements WirelessHART:

1) Équipement passerelle WirelessHART:

   Cet équipement connecte un réseau WirelessHART à l'environnement extérieur par le biais du protocole HART ou d'autres protocoles qui permettent un transfert de données avec des débits en bauds élevés. Il est possible qu'un réseau WirelessHART comporte plusieurs équipements passerelles WirelessHART actifs. Les équipements passerelles WirelessHART sont chargés de gérer le répertoire du réseau et de transmettre les informations à destination et en provenance des équipements WirelessHART.

   Pour une meilleure lisibilité en 6.9, un équipement passerelle WirelessHART est simplement dénommé Passerelle.

2) Dispositif de terrain WirelessHART:

   Le Dispositif de Terrain WirelessHART est un équipement qui peut participer à un réseau WirelessHART.

   Pour une meilleure lisibilité en 6.9, un Dispositif de Terrain WirelessHART est simplement dénommé Dispositif de Terrain.

3) Équipement adaptateur WirelessHART:

   L'équipement adaptateur WirelessHART est un Dispositif de Terrain WirelessHART spécialisé qui permet de connecter des sous-équipements HART FSK et/ou de 4 mA à 20 mA au réseau WirelessHART.

   Pour une meilleure lisibilité en 6.9, un équipement adaptateur WirelessHART est simplement dénommé Adaptateur ("Adapter") et des équipements connectés à un Adaptateur sont simplement dénommés Sous-Équipements ("Sub-Devices").

Le Paragraphe 6.9 traite principalement des spécificités de WirelessHART et définit les règles de mise en œuvre au sein des outils FDT qui sont exigées pour la communication imbriquée.

6.9.2 Topologie de réseau

Les adaptateurs sont des équipements spéciaux qui connectent d'autres couches physiques HART (habituellement HART FSK) au réseau WirelessHART comme le représente la Figure 7.

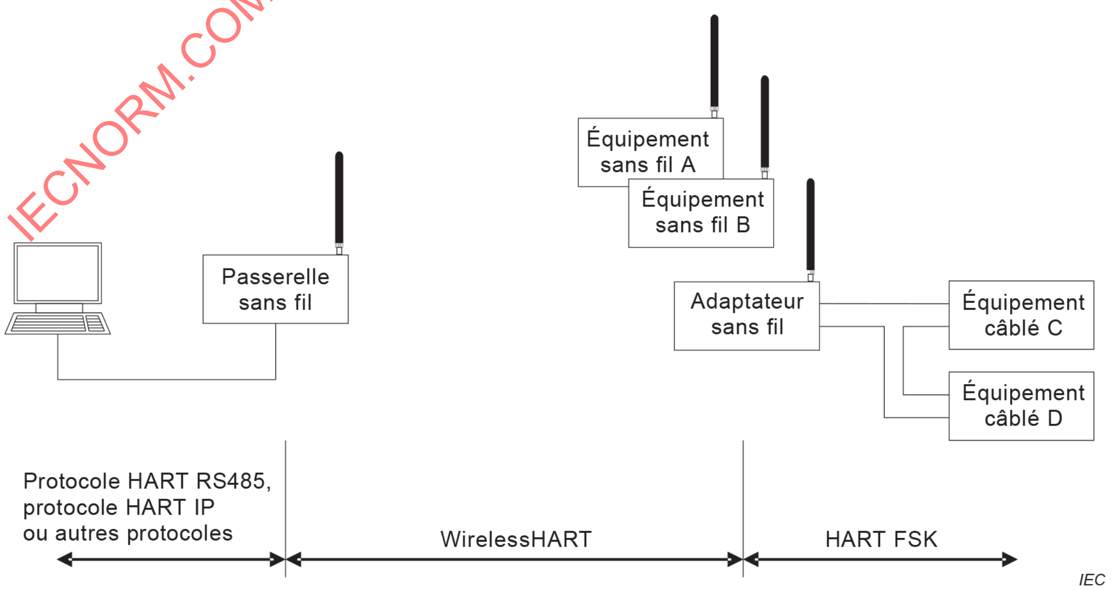


Figure 7 – Hôte connecté à un équipement passerelle WirelessHART

Du point de vue de la communication imbriquée dans les outils FDT, la Passerelle et l'Adaptateur sont tous deux des équipements passerelles qui doivent être présentés comme tels dans la topologie de réseau d'une Application-Cadre FDT. La topologie FDT résultante est représentée à la Figure 8.

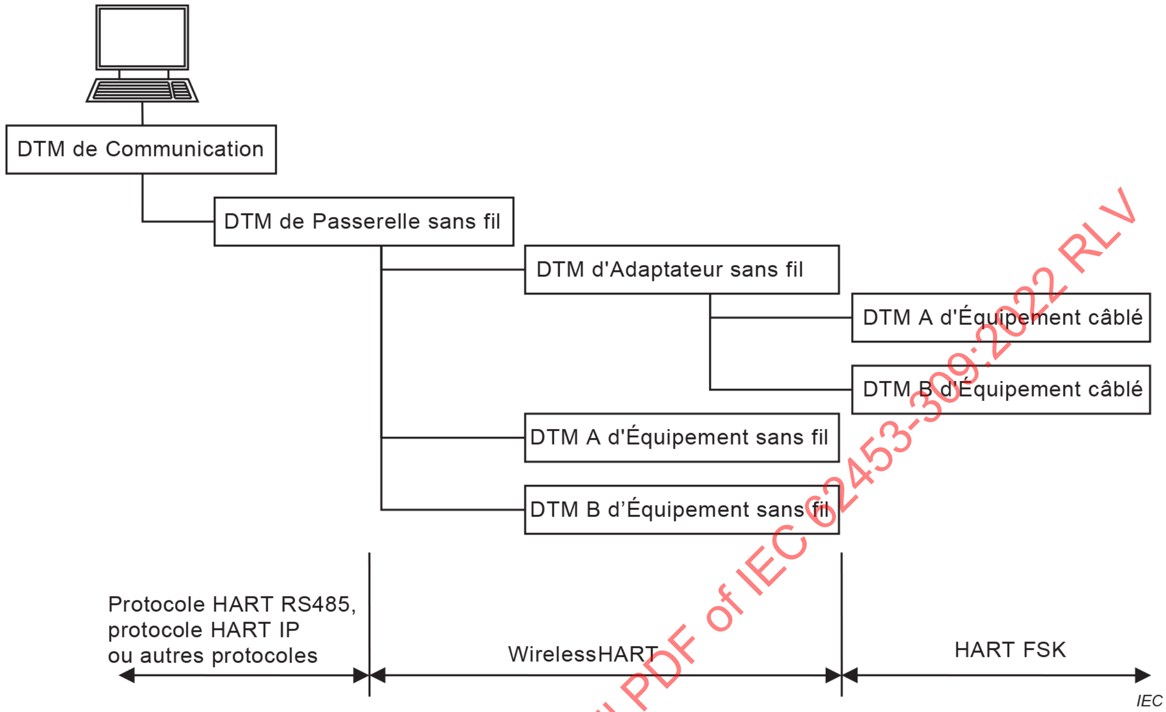


Figure 8 – Topologie FDT d'un réseau WirelessHART

Un Adaptateur interagit avec la boucle HART FSK comme un équipement HART ordinaire. Il agit comme un Maître HART, mais peut aussi être traité avec des transactions HART à partir d'un autre Maître. Une Application-Cadre FDT peut, notamment dans les cas d'utilisation de services, être connectée à la boucle HART FSK pour accéder directement à l'Adaptateur. Dans ce cas, l'Adaptateur est connecté à l'Application-Cadre FDT comme un équipement habituel dans un scénario multipoint et multimaître HART FSK tel que représenté à la Figure 9.

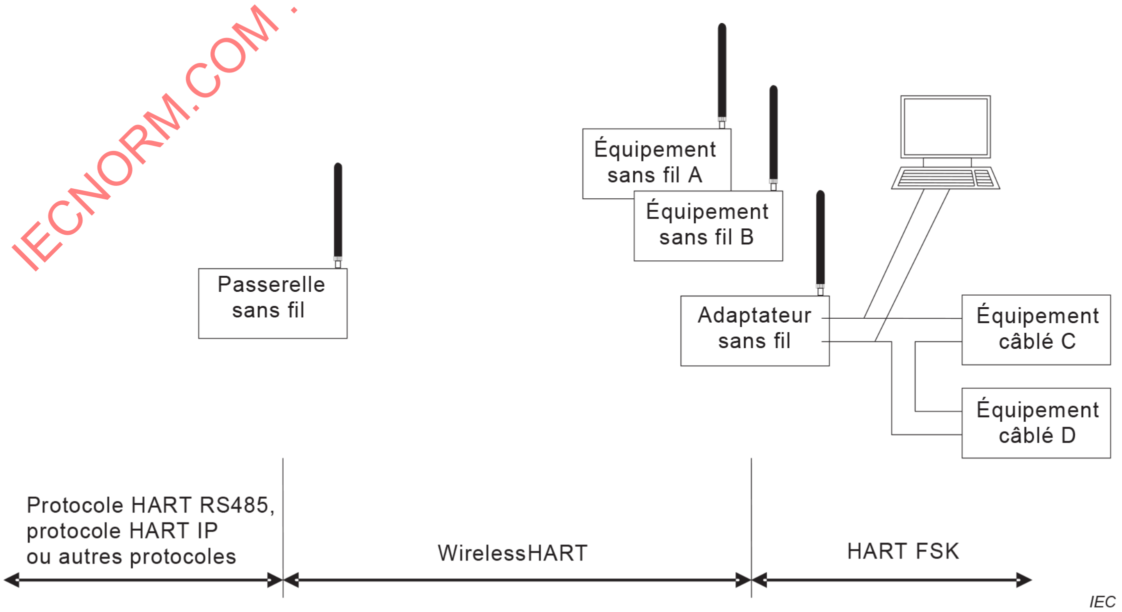


Figure 9 – Hôte connecté à HART FSK

Dans ce cas d'utilisation, la topologie de réseau dans l'Application-Cadre FDT doit être structurée comme cela est représenté à la Figure 10.

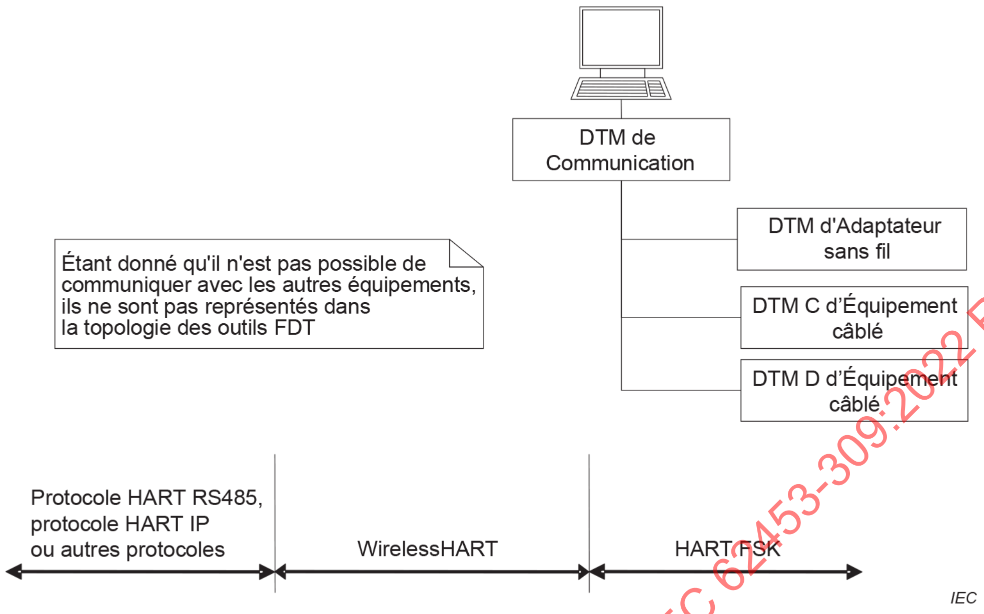


Figure 10 – Topologie FDT dans le cas d'une connexion directe à un équipement adaptateur WirelessHART

La Figure 8 et la Figure 10 indiquent qu'un DTM d'Adaptateur doit mettre en œuvre des fonctions de passerelle lorsqu'il est utilisé dans un environnement FDT WirelessHART, et doit, d'autre part, se comporter comme un équipement simple, lorsqu'il est utilisé dans un environnement HART FSK. Dans le scénario FDT 2, un DTM est toujours informé des modifications du type de communication. En utilisant les informations relatives au type de connexion actuel, le DTM d'Adaptateur doit mettre en œuvre le comportement structurel spécifique respectif.

En tant que synthèse des éléments indiqués en 6.9.2, les règles suivantes doivent être mises en œuvre:

Si l'Adaptateur est connecté au moyen du protocole WirelessHART, il doit:

- interagir comme un DTM de Passerelle;
- gérer la communication vers les sous-équipements connectés (tels que spécifiés en 6.7) qui sont reliés dans la topologie sous forme de DTM Enfants;
- traiter les transactions de DR comme cela est décrit en 6.8.

Si l'Adaptateur est connecté au moyen du protocole HART FSK, il doit:

- refuser la liaison des DTM Enfants;
- refuser la connexion à des DTM Enfants.

Si une instance d'un DTM d'Adaptateur est déplacée d'une Voie de Communication WirelessHART vers une Voie de Communication HART FSK, elle doit:

- laisser intactes toutes les instances des DTM Enfants;
- autoriser l'éloignement des DTM Enfants par rapport à son nœud.

7 Utilisation spécifique à un protocole des types de données généraux

Le Tableau 3 présente le mode d'utilisation des types de données généraux, définis dans l'IEC 62453-2 sous l'espace de noms 'fdt', avec les équipements HART.

Tableau 3 – Utilisation spécifique à un protocole des types de données généraux

Type de données	Description d'utilisation
fdt:address	La propriété 'address' ("adresse") n'est pas obligatoire pour les paramètres présentés dans les DTM. Cependant, en cas d'utilisation de cette propriété, la chaîne doit être construite selon les règles de semanticId, ce qui signifie que la propriété 'semanticId' est toujours identique à la propriété 'address'
fdt:protocolId	Voir l'Article 4
fdt:deviceTypeId	La propriété "fdt:DtmDeviceType.deviceTypeId" doit comporter le DeviceTypeID de l'équipement physique pris en charge selon le catalogue de produits en ligne de FieldComm Group's
fdt:manufacturerId	Saisir le nom du fabricant selon la liste de FieldComm Group's
fdt:semanticId fdt:applicationDomain	L'attribut applicationDomain est: FDT_HART Le sematicId pour le paramètre relatif au protocole est directement lié à la spécification du protocole. La définition des commandes constitue la base de semanticId. Le semanticId pour un paramètre suit la définition suivante: CMDxxBy et CMD31EXTENDEDxxBy pour les commandes étendues de la famille d'équipements HART 6. Les semanticId pour les octets de réponse 0 et 1 définis dans la spécification de la CPF 9 de l'IEC 61784 sont les suivants: CMDxxRESPONSE_BYTE_0 CMDxxRESPONSE_BYTE_1 xx: représente le numéro de commande, en ayant récupéré le paramètre par le biais du protocole de la CPF 9 de l'IEC 61784 ou le numéro de commande de la famille d'équipements y: octet de départ dans la définition de la commande xx, y sont exprimés en format décimal sans '0' de tête
subDeviceType	Saisir la valeur spécifique au fabricant

8 Types communs de données spécifiques à un protocole

Non applicable.

9 Types de données de gestion de réseau

9.1 Généralités

Les types de données spécifiés en 9.1 sont utilisés dans les services suivants:

- service NetworkManagementInfoRead;
- service NetworkManagementInfoWrite.

9.2 Mode d'adressage

Le mode d'adressage dépend du type du protocole HART utilisé. De même, des informations d'adressage complémentaires peuvent être nécessaires pour certains types de protocoles