

TECHNICAL REPORT



**Field device tool (FDT) interface specification –
Part 52-150: Communication implementation for common language
infrastructure – IEC 61784 CPF 15**

IECNORM.COM : Click to view the full PDF of IEC TR 62453-52-150:2017



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2017 IEC, Geneva, Switzerland

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

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

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

IEC publications search - www.iec.ch/searchpub

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

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

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

IECNORM.COM : Click to view the full text of IEC 60352-150:2017

TECHNICAL REPORT



**Field device tool (FDT) interface specification –
Part 52-150: Communication implementation for common language
infrastructure – IEC 61784 CPF 15**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.05; 35.110

ISBN 978-2-8322-4336-7

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

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references	11
3 Terms, definitions, symbols, abbreviated terms and conventions	12
3.1 Terms and definitions.....	12
3.2 Symbols and abbreviated terms	12
3.3 Conventions.....	12
3.3.1 Datatype names and references to datatypes	12
3.3.2 Vocabulary for requirements	12
3.3.3 Use of UML	12
4 Bus category	12
5 Access to instance and device data	12
5.1 General.....	12
5.2 IO signals provided by DTM	13
5.3 Data interfaces	13
5.3.1 Common data set	13
5.3.2 Mapping of Modbus datatypes to FDT datatypes.....	13
5.3.3 SemanticInfo	13
6 Protocol specific behaviour.....	14
6.1 Modbus data and addressing model.....	14
6.2 Modbus-related information of a Device DTM.....	15
6.3 Broadcasting.....	15
6.4 Unconfirmed private Modbus request.....	16
7 Protocol specific usage of general datatypes	16
8 Protocol specific common datatype: ModbusDeviceAddress	17
9 Network management datatypes.....	18
9.1 General.....	18
9.2 Configuration.....	19
9.3 Process Data Items.....	19
9.4 Parameterization.....	19
10 Communication datatypes.....	19
10.1 General.....	19
10.2 ModbusConnectRequest	19
10.3 ModbusConnectResponse	20
10.4 ModbusDisconnectRequest.....	20
10.5 ModbusDisconnectResponse	21
10.6 ModbusAbortMessage.....	21
10.7 ModbusReadCoilsRequest	22
10.8 ModbusReadCoilsResponse	23
10.9 ModbusReadDiscreteInputsRequest	24
10.10 ModbusReadDiscreteInputsResponse	24
10.11 ModbusReadHoldingRegistersRequest	25
10.12 ModbusReadHoldingRegistersResponse.....	26
10.13 ModbusReadInputRegistersRequest	27
10.14 ModbusReadInputRegistersResponse.....	28

10.15	ModbusWriteSingleCoilRequest	29
10.16	ModbusWriteSingleCoilResponse	30
10.17	ModbusWriteSingleRegisterRequest	30
10.18	ModbusWriteSingleRegisterResponse	31
10.19	ModbusReadExceptionStatusRequest	32
10.20	ModbusReadExceptionStatusResponse	32
10.21	ModbusDiagnosticsRequest	33
10.22	ModbusDiagnosticsResponse	34
10.23	ModbusGetCommEventCounterRequest	35
10.24	ModbusGetCommEventCounterResponse	35
10.25	ModbusGetCommEventLogRequest	36
10.26	ModbusGetCommEventLogResponse	37
10.27	IModbusWriteMultipleCoilsRequest	38
10.28	ModbusWriteMultipleCoilsResponse	39
10.29	ModbusWriteMultipleRegistersRequest	39
10.30	ModbusWriteMultipleRegistersResponse	40
10.31	ModbusReportSlaveIDRequest	41
10.32	ModbusReportSlaveIDResponse	41
10.33	ModbusReadFileRecordRequest	42
10.34	ModbusReadFileSubRequest	43
10.35	ModbusReadFileRecordResponse	44
10.36	ModbusReadFileSubResponse	45
10.37	ModbusWriteFileRecordRequest	45
10.38	ModbusWriteFileSubRequest	46
10.39	ModbusWriteFileRecordResponse	47
10.40	ModbusMaskWriteRegisterRequest	47
10.41	ModbusMaskWriteRegisterResponse	48
10.42	ModbusReadWriteRegistersRequest	49
10.43	ModbusReadWriteRegistersResponse	50
10.44	ModbusReadFiFoQueueRequest	51
10.45	ModbusReadFiFoQueueResponse	52
10.46	ModbusEncapsulatedInterfaceTransportRequest	53
10.47	ModbusEncapsulatedInterfaceTransportResponse	53
10.48	ModbusReadDeviceIdentificationRequest	54
10.49	ModbusReadDeviceIdentificationResponse	55
10.50	ModbusPrivateRequest	58
10.51	ModbusPrivateResponse	58
10.52	ModbusUnconfirmedPrivateRequest	59
10.53	ModbusUnconfirmedPrivateResponse	60
10.54	ModbusExceptionResponse	61
11	Datatypes for process data information	62
11.1	General	62
11.2	ModbusIOSignalInfo	62
11.3	Mapping of Modbus datatypes to FDT datatypes	63
12	Device identification	63
12.1	General	63
12.2	ModbusDeviceScanInfo datatype	63
12.3	ModbusDeviceIdentInfo datatype	64
12.4	Mapping of Information Source	66

Bibliography.....	68
Figure 1 – Part 52-150 of the IEC 62453 series	10
Figure 2 – Modbus data and addressing model	15
Figure 3 – ModbusDeviceAddress.....	17
Figure 4 – ModbusNetworkData	18
Figure 5 – ModbusConnectRequest	19
Figure 6 – ModbusConnectResponse.....	20
Figure 7 – ModbusDisconnectRequest	21
Figure 8 – ModbusDisconnectResponse	21
Figure 9 – ModbusAbortMessage.....	22
Figure 10 – ModbusReadCoilsRequest	22
Figure 11 – ModbusReadCoilsResponse.....	23
Figure 12 – ModbusReadDiscreteInputsRequest.....	24
Figure 13 – ModbusReadDiscreteInputsResponse	25
Figure 14 – ModbusReadHoldingRegistersRequest.....	26
Figure 15 – ModbusReadHoldingRegistersResponse	27
Figure 16 – ModbusReadInputRegistersRequest.....	28
Figure 17 – ModbusReadInputRegistersResponse	28
Figure 18 – ModbusWriteSingleCoilRequest	29
Figure 19 – ModbusWriteSingleCoilResponse.....	30
Figure 20 – ModbusWriteSingleRegisterRequest	31
Figure 21 – ModbusWriteSingleRegisterResponse.....	31
Figure 22 – ModbusReadExceptionStatusRequest.....	32
Figure 23 – ModbusReadExceptionStatusResponse	33
Figure 24 – ModbusDiagnosticsRequest	34
Figure 25 – ModbusDiagnosticsResponse.....	34
Figure 26 – ModbusGetCommEventCounterRequest.....	35
Figure 27 – ModbusGetCommEventCounterResponse	36
Figure 28 – ModbusGetCommEventLogRequest	37
Figure 29 – ModbusGetCommEventLogResponse	37
Figure 30 – ModbusWriteMultipleCoilsRequest	38
Figure 31 – ModbusWriteMultipleCoilsResponse.....	39
Figure 32 – ModbusWriteMultipleRegistersRequest	40
Figure 33 – ModbusWriteMultipleRegistersResponse.....	40
Figure 34 – ModbusReportSlaveIDRequest.....	41
Figure 35 – ModbusReportSlaveIDResponse	42
Figure 36 – ModbusReadFileRecordRequest	43
Figure 37 – ModbusReadFileSubRequest	43
Figure 38 – ModbusReadFileRecordResponse.....	44
Figure 39 – ModbusReadFileSubResponse.....	45
Figure 40 – ModbusWriteFileRecordRequest	46
Figure 41 – ModbusWriteFileSubRequest	46

Figure 42 – ModbusWriteFileRecordResponse	47
Figure 43 – ModbusMaskWriteRegisterRequest	48
Figure 44 – ModbusMaskWriteRegisterResponse	49
Figure 45 – ModbusReadWriteRegistersRequest	50
Figure 46 – ModbusReadWriteRegistersResponse	51
Figure 47 – ModbusReadFiFoQueueRequest	52
Figure 48 – ModbusReadFiFoQueueResponse	52
Figure 49 – ModbusEncapsulatedInterfaceTransportRequest	53
Figure 50 – ModbusEncapsulatedInterfaceTransportResponse	54
Figure 51 – ModbusReadDeviceIdentificationRequest	55
Figure 52 – ModbusReadDeviceIdentificationResponse	56
Figure 53 – ModbusIdentificationObject	56
Figure 54 – ModbusPrivateRequest	58
Figure 55 – ModbusPrivateResponse	59
Figure 56 – ModbusUnconfirmedPrivateRequest	60
Figure 57 – ModbusUnconfirmedPrivateResponse	60
Figure 58 – ModbusExceptionResponse	61
Figure 59 – ModbusIOSignalInfo	62
Figure 60 – ModbusDeviceScanInfo	64
Figure 61 – ModbusDeviceIdentInfo	65
Table 1 – Mapping of datatypes	13
Table 2 – Usage of general datatypes	14
Table 3 – Usage of broadcasts in transaction requests	16
Table 4 – Protocol specific usage of general datatypes	17
Table 5 – ModbusDeviceAddress	18
Table 6 – ModbusDeviceSerialAddress	18
Table 7 – ModbusDeviceTcpAddress	18
Table 8 – Modbus Network Data	19
Table 9 – ModbusConnectRequest datatype	20
Table 10 – ModbusConnectResponse datatype	20
Table 11 – ModbusDisconnectRequest datatype	21
Table 12 – ModbusDisconnectResponse datatype	21
Table 13 – ModbusAbortMessage datatype	22
Table 14 – ModbusReadCoilsRequest datatype	23
Table 15 – ModbusReadCoilsResponse datatype	23
Table 16 – ModbusReadDiscreteInputsRequest datatype	24
Table 17 – ModbusReadDiscreteInputsResponse datatype	25
Table 18 – ModbusReadHoldingRegistersRequest datatype	26
Table 19 – ModbusReadHoldingRegistersResponse datatype	27
Table 20 – ModbusReadInputRegistersRequest datatype	28
Table 21 – ModbusReadInputRegistersResponse datatype	29
Table 22 – ModbusWriteSingleCoilRequest datatype	29

Table 23 – ModbusWriteSingleCoilResponse datatype.....	30
Table 24 – ModbusWriteSingleRegisterRequest datatype	31
Table 25 – ModbusWriteSingleRegisterResponse datatype.....	32
Table 26 – ModbusReadExceptionStatusRequest datatype	32
Table 27 – ModbusReadExceptionStatusResponse datatype	33
Table 28 – ModbusDiagnosticsRequest datatype	34
Table 29 – ModbusDiagnosticsResponse datatype	35
Table 30 – ModbusGetCommEventCounterRequest datatype	35
Table 31 – ModbusGetCommEventCounterResponse datatype	36
Table 32 – ModbusGetCommEventLogRequest datatype	37
Table 33 – ModbusGetCommEventLogResponse datatype	38
Table 34 – ModbusWriteMultipleCoilsRequest datatype	39
Table 35 – ModbusWriteMultipleCoilsResponse datatype	39
Table 36 – ModbusWriteMultipleRegistersRequest datatype	40
Table 37 – ModbusWriteMultipleRegistersResponse	41
Table 38 – ModbusReportSlaveIDRequest datatype	41
Table 39 – ModbusReportSlaveIDResponse	42
Table 40 – ModbusReadFileRecordRequest datatype	43
Table 41 – ModbusReadFileSubRequest datatype	44
Table 42 – ModbusReadFileRecordResponse	45
Table 43 – ModbusReadFileSubResponse	45
Table 44 – ModbusWriteFileRecordRequest datatype	46
Table 45 – ModbusWriteFileSubRequest datatype	47
Table 46 – ModbusWriteFileRecordResponse	47
Table 47 – ModbusMaskWriteRegisterRequest datatype	48
Table 48 – ModbusMaskWriteRegisterResponse	49
Table 49 – ModbusReadWriteRegistersRequest datatype	50
Table 50 – ModbusReadWriteRegistersResponse	51
Table 51 – ModbusReadFiFoQueueRequest datatype	52
Table 52 – ModbusReadFiFoQueueResponse	53
Table 53 – ModbusEncapsulatedInterfaceTransportRequest datatype.....	53
Table 54 – ModbusEncapsulatedInterfaceTransportResponse	54
Table 55 – ModbusReadDeviceIdentificationRequest datatype	55
Table 56 – ModbusReadDeviceIdentificationResponse	57
Table 57 – ModbusIdentificationObject	58
Table 58 – ModbusPrivateRequest datatype	58
Table 59 – ModbusPrivateResponse	59
Table 60 – ModbusUnconfirmedPrivateRequest datatype.....	60
Table 61 – ModbusUnconfirmedPrivateResponse	61
Table 62 – ModbusExceptionResponse.....	62
Table 63 – ModbusIOSignalInfo datatype.....	63
Table 64 – ModbusDeviceScanInfo datatype.....	64
Table 65 – ModbusDeviceIdentInfo datatype.....	65

Table 66 – Protocol specific mapping of scan information	66
Table 67 – Profile specific mapping of identification information	67

IECNORM.COM : Click to view the full PDF of IEC TR 62453-52-150:2017

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 52-150: Communication implementation for common language infrastructure – IEC 61784 CPF 15

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.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC TR 62453-52-150, which is a technical report, has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

Each part of the IEC 62453-52-xy series is intended to be read in conjunction with its corresponding part in the IEC 62453-3xy series. This document corresponds to IEC 63453-315.

The text of this technical report is based on the following documents:

Enquiry draft	Report on voting
65E/440/DTR	65E/514/RVC

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

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

The 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 "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

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

A bilingual version of this publication may be issued at a later date.

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

INTRODUCTION

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

With the integration of fieldbuses 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 Device Type Manager (DTM), 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 this part of the IEC 62453-52-xy series is aligned in the structure of the IEC 62453 series.

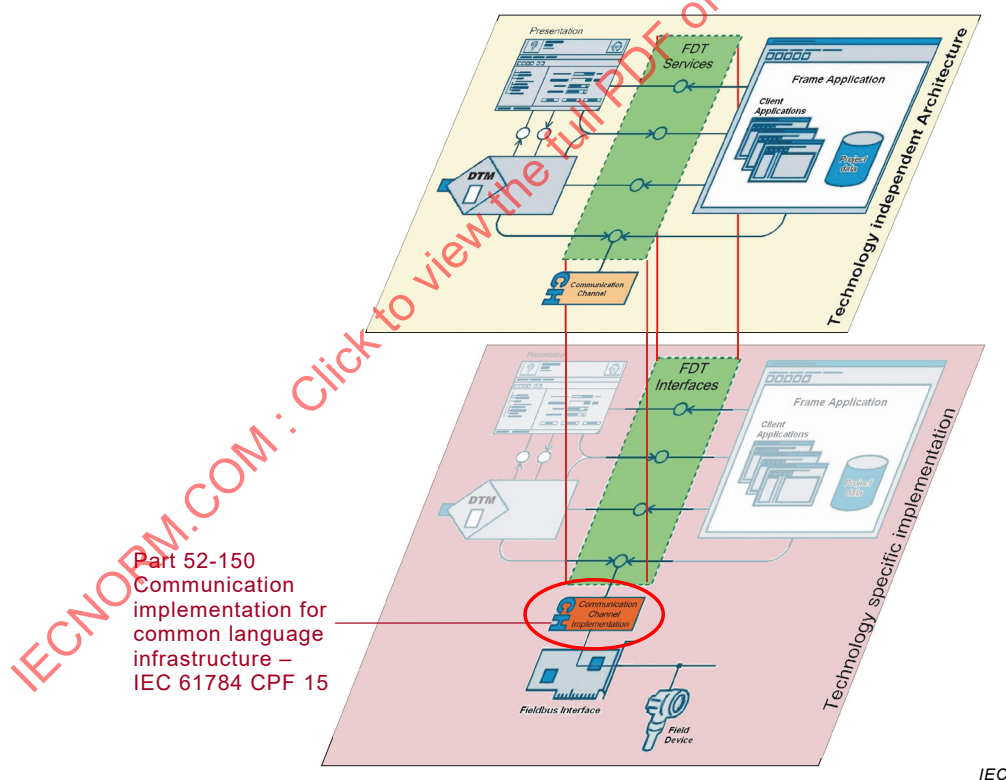


Figure 1 – Part 52-150 of the IEC 62453 series

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 52-150: Communication implementation for common language infrastructure – IEC 61784 CPF 15

1 Scope

This part of the IEC 62453-52-xy series, which is a Technical Report, provides information for integrating the Modbus®¹ technology into the CLI-based implementation of FDT interface specification (IEC TR 62453-42).

This part of IEC 62453 specifies the implementation of communication and other services based on IEC 62453-315.

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 61131-3:2013, *Programmable controllers – Part 3: Programming languages*

IEC 61158-5-15:2010, *Industrial communication networks – Fieldbus specifications – Part 5-15: Application layer service definition – Type 15 elements*

IEC 61784-1:2014, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 62453-1:2016, *Field device tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2016, *Field device tool (FDT) interface specification – Part 2: Concepts and detailed description*

IEC TR 62453-42:2016, *Field device tool (FDT) interface specification – Part 42: Object model integration profile – Common language infrastructure*

IEC 62453-315:2009, *Field device tool (FDT) interface specification – Part 315: Communication profile integration – IEC 61784 CPF 15*

IEC 62453-315:2009/AMD1:2016

¹ Modbus is the trademark of Schneider Automation Inc. It is registered in the United States of America. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trademark Modbus. Use of the trademark Modbus requires permission from Schneider Automation Inc.

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, IEC 62453-2, IEC TR 62453-42 and IEC 62453-315 apply.

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

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

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviations given in IEC 62453-1, IEC 62453-2, IEC 62453-315, IEC TR 62453-42 and the following apply.

IO	Input/Output
----	--------------

3.3 Conventions

3.3.1 Datatype names and references to datatypes

The conventions for naming and referencing of datatypes are explained in IEC 62453-2:2016, 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 Annex A of IEC 62453-1:2016.

4 Bus category

IEC 61784 CPF 15 protocol is identified in the attribute busCategory of the BusCategory element by the identifiers, as specified in IEC 62453-315.

5 Access to instance and device data

5.1 General

The minimum set of provided data shall be:

- All process values available for the device shall be modeled as ProcessData including the ranges and scaling if applicable;
- All network configuration related parameters shall be exposed in NetworkData (see Clause 9).

5.2 IO signals provided by DTM

A DTM shall provide IO signal information for the device using the IProcessData interface. The IO signals describe datatype and address parameters of process data as detailed in 11.2.

5.3 Data interfaces

5.3.1 Common data set

For Modbus no minimum set of parameters or common data set is defined which shall be provided by a DTM. Exposure of device parameters is at the vendor's discretion. If a DTM provides process values, the process variables shall be modelled as process data objects.

5.3.2 Mapping of Modbus datatypes to FDT datatypes

Modbus uses datatypes as specified in [2]² for the transmission on the fieldbus. The FDT interfaces IDeviceData and IInstanceData use .NET datatypes, while PLC applications use datatypes defined in IEC 61131-3. This subclause defines the mapping of parameter datatypes, whereas mapping of process datatypes is defined in 11.3.

The mapping of parameter datatypes is described in Table 1.

Table 1 – Mapping of datatypes

Modbus datatype	FDT datatype	IEC 61131 datatype
Discrete Inputs	BitArray	ARRAY [] OF BOOL
Coil	bool	BOOL
Coils	BitArray	ARRAY [] OF BOOL
Input Registers	ushort[]	ARRAY [] OF WORD
Holding Register	ushort	WORD
Holding Registers	ushort[]	ARRAY [] OF WORD

The FDT data types do not change any byte or bit order of the data. The data interpretation, e.g. if a float value is transferred in a holding register, is under the responsibility of the DTM respectively the user of the data. The user of the data shall know how the data is represented by the hardware device.

5.3.3 SemanticInfo

The usage of the SemanticInfo datatype is shown in Table 2. The identifier in SemanticId shall be unique and always reference the same element. This means the semantic information shall be the same whenever the same data is referenced. By using this attribute e.g. a Frame Application is able to get the information regarding the meaning and usage of a single data structure.

² Figures in square brackets refer to the Bibliography.

Table 2 – Usage of general datatypes

Attribute	Description for use in Modbus
SemanticInfo.ParameterReadAddress / SemanticInfo.ParameterWriteAddress	The value of ParameterReadAddress and ParameterWriteAddress is a string conforming to a pattern as follows: FunctionCode.StartAddress.BitOffset.BitLength where FunctionCode = Modbus function code to access the data StartAddress = starting address BitOffset = start bit within the starting address BitLength = length in bits All values are unsigned integers.
SemanticInfo.ApplicationDomain	The value of ApplicationDomain is as follows: MODBUS
SemanticInfo.SemanticId	The value of SemanticId is vendor specific.

6 Protocol specific behaviour

6.1 Modbus data and addressing model

The Modbus data model is very simple and described in [2]. Figure 2 shows an extract of the data model description.

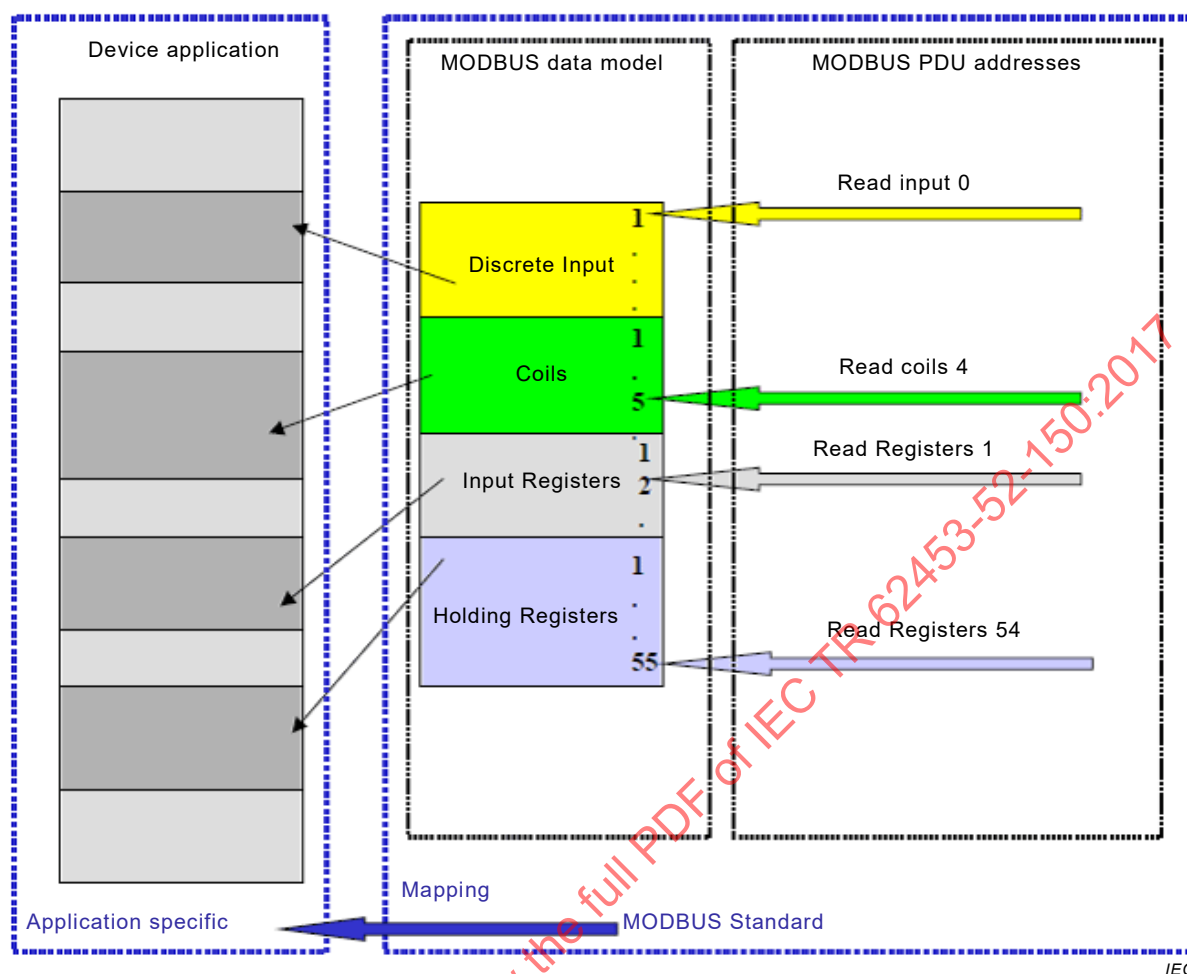


Figure 2 – Modbus data and addressing model

6.2 Modbus-related information of a Device DTM

The information used by a Modbus client (IO scanner) to set up the Modbus network properly and allow cyclic communication between control system and Modbus server devices is provided by a DTM in Process data items.

A DTM of a Modbus device shall deliver process data items related information to get integrated into an FDT-based engineering system.

This specification makes no assumptions whether a modular Modbus device is modeled in a Device DTM, Composite DTM or Gateway DTM. All types of DTMs shall provide all mandatory information defined in the following subclauses.

6.3 Broadcasting

In broadcast mode a DTM can send a Modbus request to all devices connected to the bus. This mode is only supported for devices which are connected via Modbus Serial Line. The connection can either be a direct connection or a connection via a gateway. The broadcast mode shall be initiated by a ConnectRequest with the slave address of the target device set to 0 (SlaveAddress=0).

Because in broadcast mode no response will be returned by the device, the broadcast mode shall be only used with the transaction requests listed in Table 3.

Table 3 – Usage of broadcasts in transaction requests

Broadcast Transaction Request	Restrictions
ModbusWriteSingleRequest	None
ModbusWriteSingleRegisterRequest	None
ModbusDiagnosticsRequest	This transaction request shall be used in broadcast mode only with the following sub-functions: 0x01: Restart Communication Option 0x03: Change ASCII Input Delimiter 0x04: Force Listen Only Mode 0x0A: Clear Counters and Diagnostic Register 0x14: Clear Overrun Counter and Flag
ModbusWriteMultipleCoilsRequest	None
ModbusWriteMultipleRegisterRequest	None
ModbusWriteFileRecordRequest	None
ModbusMaskWriteRegisterRequest	None
ModbusPrivateRequest	Shall be only used with private services if no response is required from the device.

Although in broadcast mode no response will be returned by the target devices, a transaction response which corresponds to the transaction request, shall be generated. This transaction response shall be generated in order to inform the DTM that the broadcast request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it shall provide this generated transaction response. If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, shall provide this transaction response.

6.4 Unconfirmed private Modbus request

This transaction request can be used to send any unconfirmed private Modbus request. Although no response will be returned by the target device, a transaction response, which corresponds to the transaction request, shall be generated. This transaction response shall be generated in order to inform the DTM that the unconfirmed request was sent on the bus. If a Modbus Communication DTM is used to establish the communication, it shall provide this generated transaction response. If no Modbus Communication DTM is used to establish the communication, the Modbus Gateway DTM, which provides the link to the higher communication level different from Modbus, shall provide this transaction response.

7 Protocol specific usage of general datatypes

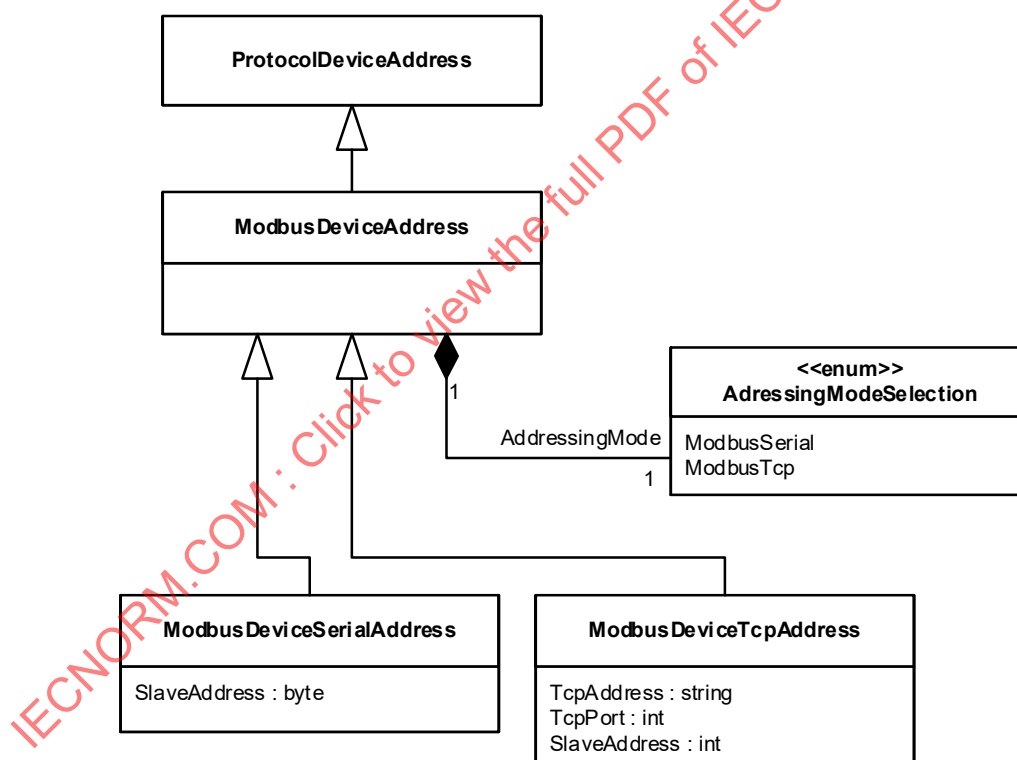
Table 4 shows how general datatypes are used with IEC 61784 CPF 15 devices.

Table 4 – Protocol specific usage of general datatypes

Attribute	Description for use
ProtocolId	See Clause 4
PhysicalLayer	Not applicable
ApplicationDomain / SemanticId	See 5.3.3
Address	See Clause 8
ManufacturerId	Modbus Vendor Name
DeviceTypeId	Modbus Product Code
HardwareRevision	Not applicable
SoftwareRevision	Modbus MajorMinorRevision
ProtocolIdentificationProfile	Not applicable

8 Protocol specific common datatype: ModbusDeviceAddress

The only protocol-specific common datatype is ModbusDeviceAddress. The protocol specific device address relates to network management and communication.



IEC

Figure 3 – ModbusDeviceAddress

Table 5, Table 6 and Table 7 show how address information is used with the different Modbus protocols.

Table 5 – ModbusDeviceAddress

Member Name	Type	Description
AddressingModeSelection	enum	Specifies whether a Modbus Serial or a Modbus TCP address is used

Table 6 – ModbusDeviceSerialAddress

Member Name	Type	Description
SlaveAddress	byte	Address of the Modbus Serial device

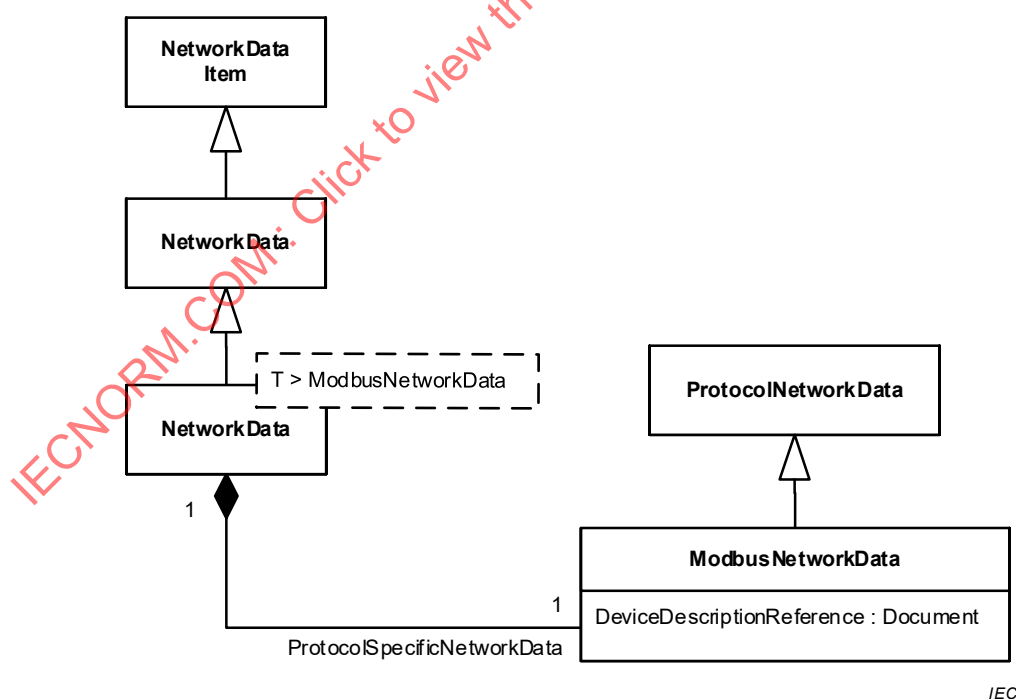
Table 7 – ModbusDeviceTcpAddress

Member Name	Type	Description
TcpAddress	string	String representation of the IP address of a Modbus TCP device
TcpPort	int	Port of the Modbus TCP connection
SlaveAddress	int	Slave address of a (virtual) Modbus device behind a Modbus TCP/ Modbus Serial Line gateway

9 Network management datatypes

9.1 General

The data needed for management of the network are exposed by the Device DTM in the INetworkData interface (see Figure 4).



Used in:
INetworkData.GetNetworkDataInfo()

Figure 4 – ModbusNetworkData

The properties of ModbusNetworkData are described in Table 8.

Table 8 – Modbus Network Data

Member Name	Type	Description
DeviceDescriptionReference	Document	Path to file which contains the Modbus device description if available

9.2 Configuration

Modbus networks do not require a special configuration. The focus for Modbus DTMs is on the process data items and parameterization of devices. See 9.4 and 11.2.

9.3 Process Data Items

For a detailed description about Process Data Items, please refer to Clause 11.

9.4 Parameterization

For the parameterization of a Modbus device the DTM shall use the communication services and communication datatypes described in Clause 10.

Because Modbus does not define a specific or common way how to parameterize a device, the parameterization is vendor specific.

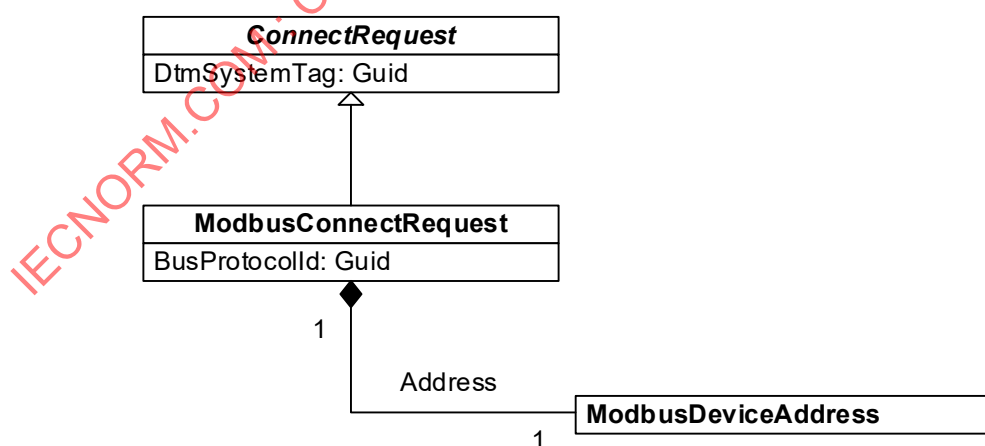
10 Communication datatypes

10.1 General

The datatypes contain the address information and the communication data required to execute the respective request or to transport the response information.

10.2 ModbusConnectRequest

This is the Modbus specific implementation of the abstract class ConnectRequest (see Figure 5).



IEC

Used in:

ICommunication.BeginConnect()

Figure 5 – ModbusConnectRequest

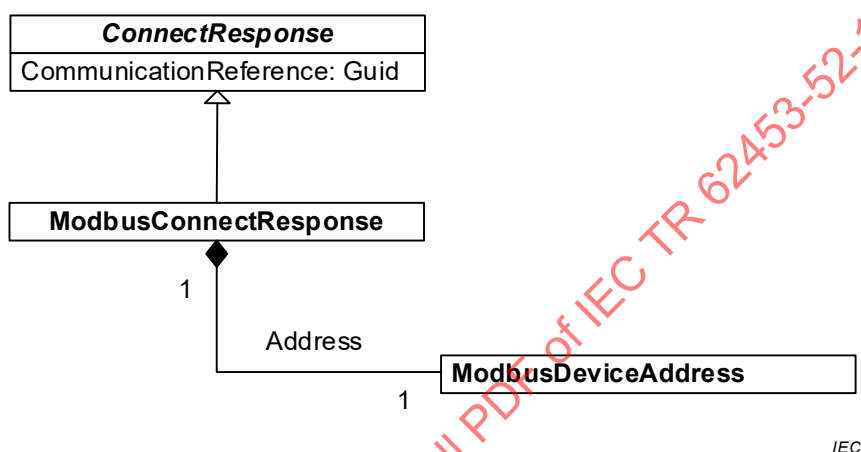
The properties of the ModbusConnectRequest datatype are described in Table 9.

Table 9 – ModbusConnectRequest datatype

Property	Description
Address	Address information of the Modbus device
BusProtocolId	Protocol identifier of the used protocol (Serial or TCP) of the connect request
DtmSystemTag	Unique identification of the DTM in the frame application

10.3 ModbusConnectResponse

This is the Modbus specific implementation of the abstract class ConnectResponse (see Figure 6).



IEC

Used in:

ICommunication.EndConnect()

Figure 6 – ModbusConnectResponse

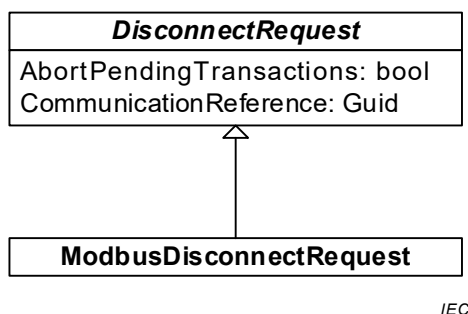
The properties of the ModbusConnectResponse datatype are described in Table 10.

Table 10 – ModbusConnectResponse datatype

Property	Description
Address	Address information of the Modbus device
CommunicationReference	Identifier for a communication link to a device

10.4 ModbusDisconnectRequest

This is the Modbus specific implementation of the abstract class DisconnectRequest (see Figure 7).

**Used in:**

ICommunication.BeginDisconnect()

Figure 7 – ModbusDisconnectRequest

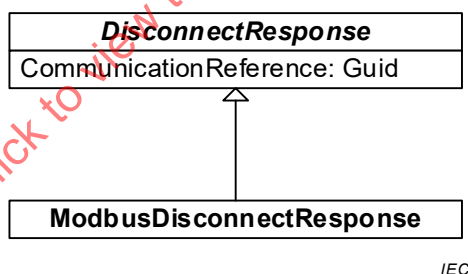
The properties of the ModbusDisconnectRequest datatype are described in Table 11.

Table 11 – ModbusDisconnectRequest datatype

Property	Description
AbortPendingTransactions	Indicates whether pending transactions shall be aborted.
CommunicationReference	Identifier for a communication link to a device.

10.5 ModbusDisconnectResponse

This is the Modbus specific implementation of the abstract class DisconnectResponse (see Figure 8).

**Used in:**

ICommunication.EndDisconnect()

Figure 8 – ModbusDisconnectResponse

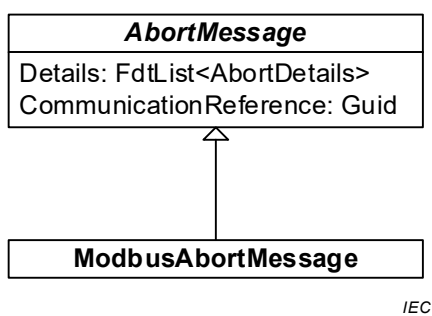
The properties of the ModbusDisconnectResponse datatype are described in Table 12.

Table 12 – ModbusDisconnectResponse datatype

Property	Description
CommunicationReference	Identifier for a communication link to a device.

10.6 ModbusAbortMessage

This is the Modbus specific implementation of the abstract AbortMessage class (see Figure 9).



Used in:
 ICommunication.EndDisconnect()

Figure 9 – ModbusAbortMessage

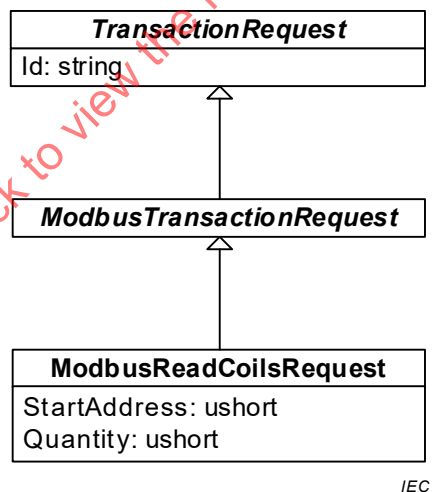
The properties of the ModbusAbortMessage datatype are described in Table 13.

Table 13 – ModbusAbortMessage datatype

Property	Description
CommunicationReference	Identifier for a communication link to a device.
Details	Details about the cause and source of the abort.

10.7 ModbusReadCoilsRequest

This subclause describes the request for the Modbus service Read Coils (see Figure 10).



Used in:
 ICommunication.BeginCommunicationRequest()

Figure 10 – ModbusReadCoilsRequest

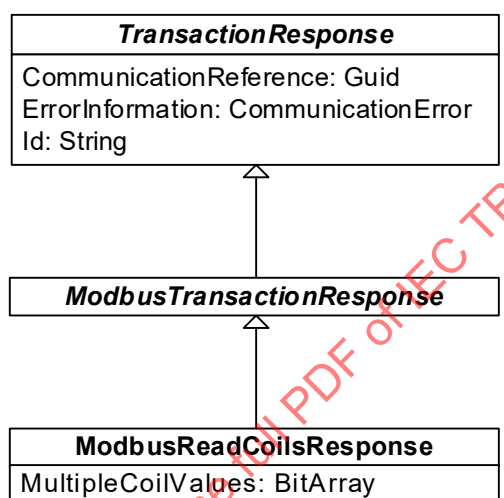
The properties of the ModbusReadCoilsRequest datatype are described in Table 14.

Table 14 – ModbusReadCoilsRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
Quantity	Number of coils to be read	Quantity of coils
StartAddress	Address of the first coil to be read	Address of first coil

10.8 ModbusReadCoilsResponse

This subclause describes the response for the Modbus service Read Coils (see Figure 11).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 11 – ModbusReadCoilsResponse

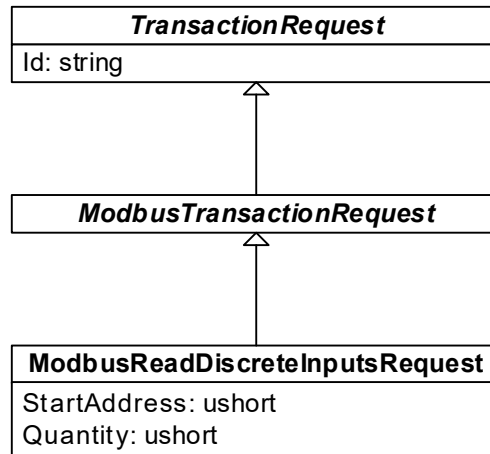
The properties of the ModbusReadCoilsResponse datatype are described in Table 15.

Table 15 – ModbusReadCoilsResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
MultipleCoilValues	Bit-array with each coil state coded in one character: - "0" = FALSE or "OFF" - "1" = TRUE or "ON"	Data

10.9 ModbusReadDiscreteInputsRequest

This subclause describes the request for the Modbus service Read Discrete Inputs (see Figure 12).



Used in:

ICommunication.BeginCommunicationRequest()

Figure 12 – ModbusReadDiscreteInputsRequest

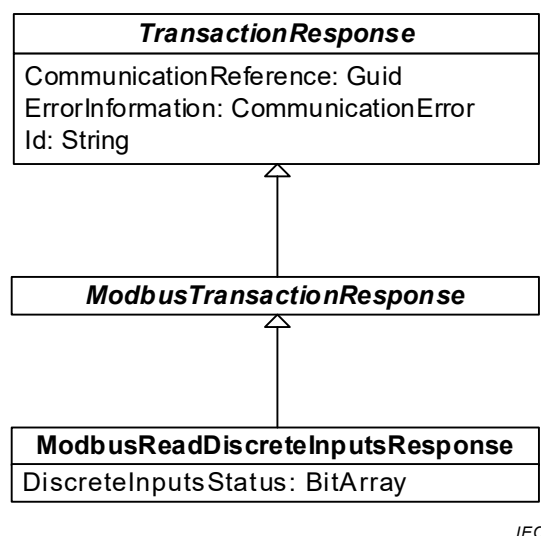
The properties of the ModbusReadDiscreteInputsRequest datatype are described in Table 16.

Table 16 – ModbusReadDiscreteInputsRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request	
Quantity	Number of discrete inputs to be read	Quantity of discretes
StartAddress	Address of the first discrete input to be read	Address of first discrete

10.10 ModbusReadDiscreteInputsResponse

This subclause describes the response for the Modbus service Read Discrete Inputs (see Figure 13).

**Used in:**

ICommunication.EndCommunicationRequest()

Figure 13 – ModbusReadDiscreteInputsResponse

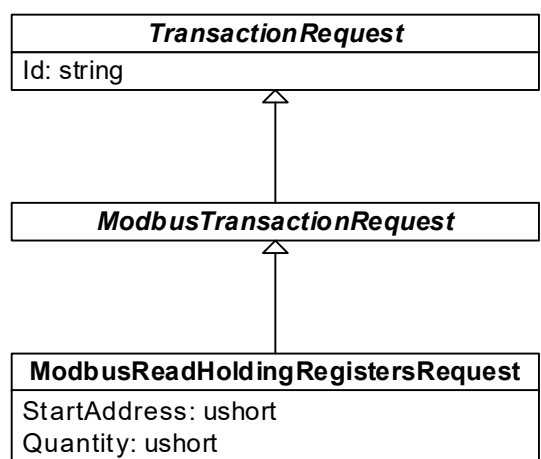
The properties of the ModbusDiscreteInputsResponse datatype are described in Table 17.

Table 17 – ModbusReadDiscreteInputsResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
DiscreteInputsStatus	Bit-array with each coil state coded in one character: "0" = FALSE or "OFF" "1" = TRUE or "ON"	Data
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.11 ModbusReadHoldingRegistersRequest

This subclause describes the request for the Modbus service Read Holding Registers (see Figure 14).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 14 – ModbusReadHoldingRegistersRequest

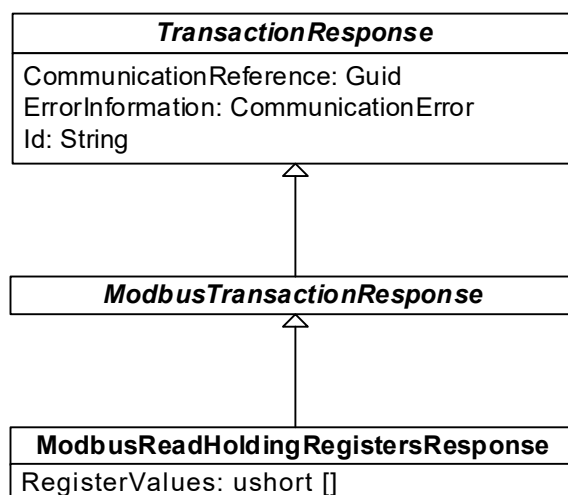
The properties of the ModbusReadHoldingRegistersRequest datatype are described in Table 18.

Table 18 – ModbusReadHoldingRegistersRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
Quantity	Number of holding registers to be read	Quantity of holding registers to read
StartAddress	Address of the first holding register to be read	Address of first holding register to read

10.12 ModbusReadHoldingRegistersResponse

This subclause describes the response for the Modbus service Read Holding Registers (see Figure 15).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 15 – ModbusReadHoldingRegistersResponse

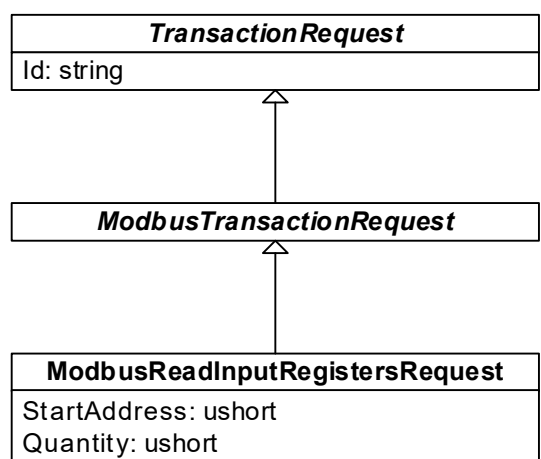
The properties of the ModbusReadHoldingRegistersResponse datatype are described in Table 19.

Table 19 – ModbusReadHoldingRegistersResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
RegisterValues	Read holding register values.	Data

10.13 ModbusReadInputRegistersRequest

This subclause describes the request for the Modbus service Read Input Registers (see Figure 16).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 16 – ModbusReadInputRegistersRequest

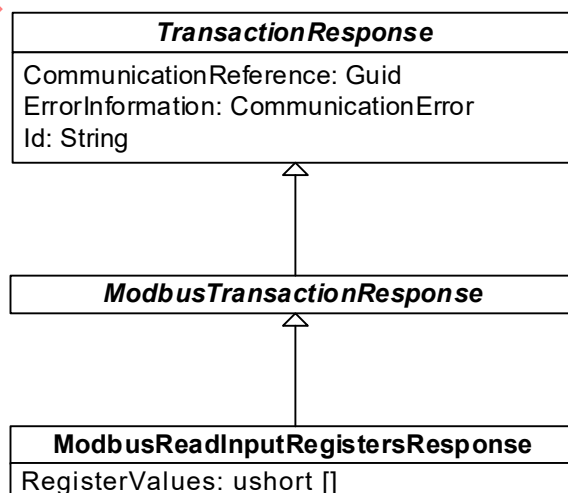
The properties of the ModbusReadInputRegistersRequest datatype are described in Table 20.

Table 20 – ModbusReadInputRegistersRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request	
Quantity	Number of input registers to be read	Quantity of input registers
StartAddress	Address of the first input register to be read	Address of first input register

10.14 ModbusReadInputRegistersResponse

This subclause describes the response for the Modbus service Read Input Registers (see Figure 17).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 17 – ModbusReadInputRegistersResponse

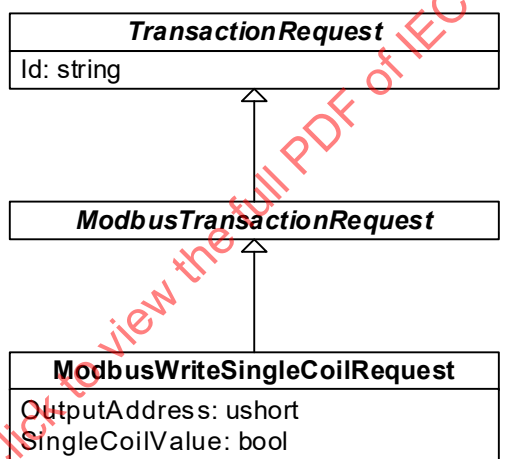
The properties of the ModbusReadInputRegistersResponse datatype are described in Table 21.

Table 21 – ModbusReadInputRegistersResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
RegisterValues	Read input register values.	Data

10.15 ModbusWriteSingleCoilRequest

This subclause describes the request for the Modbus service Write Single Coil (see Figure 18).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 18 – ModbusWriteSingleCoilRequest

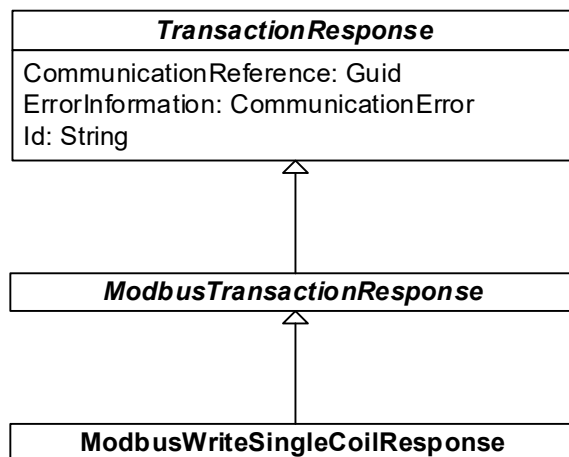
The properties of the ModbusWriteSingleCoilRequest datatype are described in Table 22.

Table 22 – ModbusWriteSingleCoilRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
OutputAddress	Address of the coil to be forced	Address of first coil
SingleCoilValue	Coil state to be forced, with: "0"=FALSE or "OFF" "1"=TRUE or "ON"	Data single coil

10.16 ModbusWriteSingleCoilResponse

This subclause describes the response for the Modbus service Write Single Coil (see Figure 19).



Used in:

ICommunication.EndCommunicationRequest()

Figure 19 – ModbusWriteSingleCoilResponse

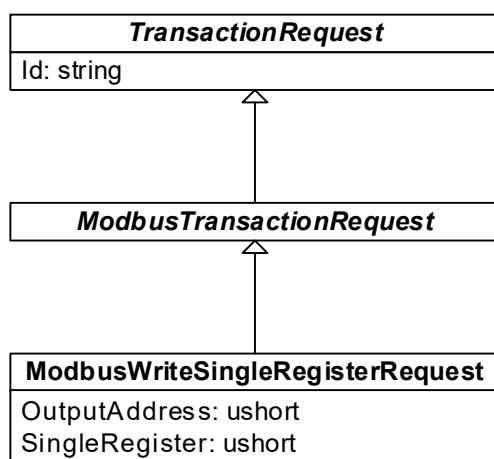
The properties of the ModbusWriteSingleCoilResponse datatype are described in Table 23.

Table 23 – ModbusWriteSingleCoilResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.17 ModbusWriteSingleRegisterRequest

This subclause describes the request for the Modbus service Write Single Register (see Figure 20).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 20 – ModbusWriteSingleRegisterRequest

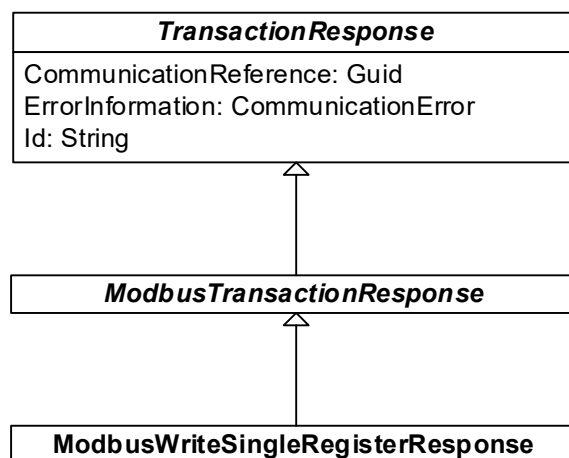
The properties of the ModbusWriteSingleRegisterRequest datatype are described in Table 24.

Table 24 – ModbusWriteSingleRegisterRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request	
OutputAddress	Address of the holding register to be written	Address of first holding register to write
SingleRegister	Value to be written to the register	Data

10.18 ModbusWriteSingleRegisterResponse

This subclause describes the response for the Modbus service Write Single Register (see Figure 21).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 21 – ModbusWriteSingleRegisterResponse

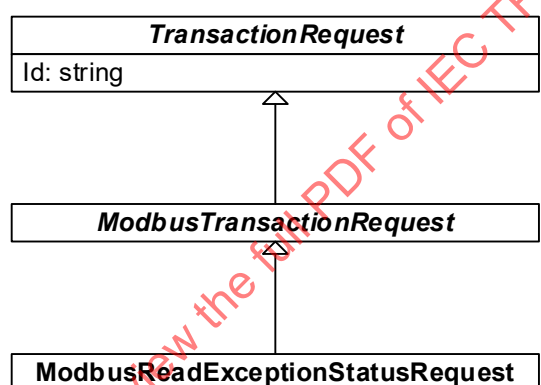
The properties of the ModbusWriteSingleRegisterResponse datatype are described in Table 25.

Table 25 – ModbusWriteSingleRegisterResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.19 ModbusReadExceptionStatusRequest

This subclause describes the request for the Modbus service Read Exception Status (see Figure 22).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 22 – ModbusReadExceptionStatusRequest

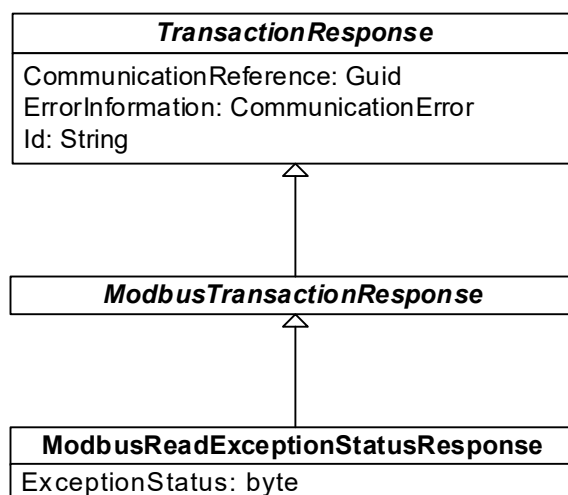
The properties of the ModbusReadExceptionStatusRequest datatype are described in Table 26.

Table 26 – ModbusReadExceptionStatusRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	

10.20 ModbusReadExceptionStatusResponse

This subclause describes the response for the Modbus service Read Exception Status (see Figure 23).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 23 – ModbusReadExceptionStatusResponse

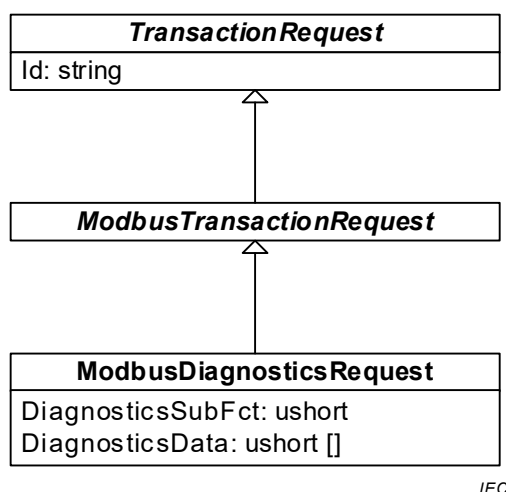
The properties of the ModbusReadExceptionStatusResponse datatype are described in Table 27.

Table 27 – ModbusReadExceptionStatusResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
ExceptionStatus	Exception status of Modbus Serial Line device.	Output data
Id	[Optional] Identifier for a single Transaction Request.	

10.21 ModbusDiagnosticsRequest

This subclause describes the request for the Modbus service Diagnostics (see Figure 24).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 24 – ModbusDiagnosticsRequest

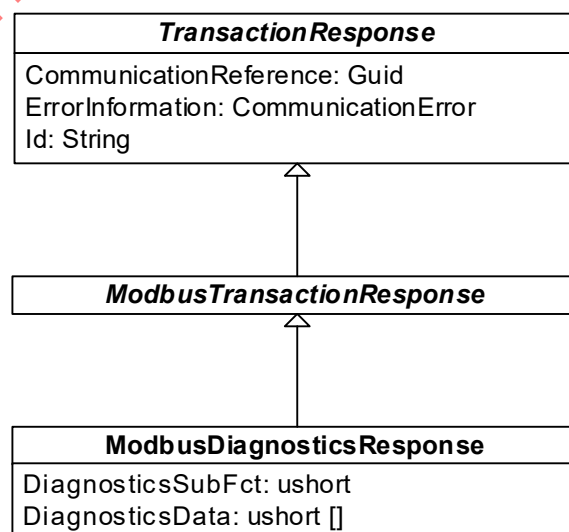
The properties of the ModbusDiagnosticsRequest datatype are described in Table 28.

Table 28 – ModbusDiagnosticsRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
DiagnosticsData	Required data, depending on the sub-function type	Data
DiagnosticsSubFct	Sub-function code	Sub-function
Id	[Optional] Identifier for a single Transaction Request	

10.22 ModbusDiagnosticsResponse

This subclause describes the response for the Modbus service Diagnostics (see Figure 25).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 25 – ModbusDiagnosticsResponse

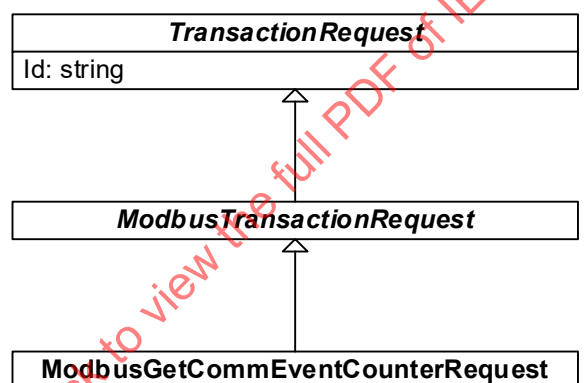
The properties of the ModbusDiagnosticsResponse datatype are described in Table 29.

Table 29 – ModbusDiagnosticsResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
DiagnosticsData	Required data, depending on the sub-function type.	Data
DiagnosticsSubFct	Sub-function code	Sub-function
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.23 ModbusGetCommEventCounterRequest

This subclause describes the request for the Modbus service Get Comm Event Counter (see Figure 26).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 26 – ModbusGetCommEventCounterRequest

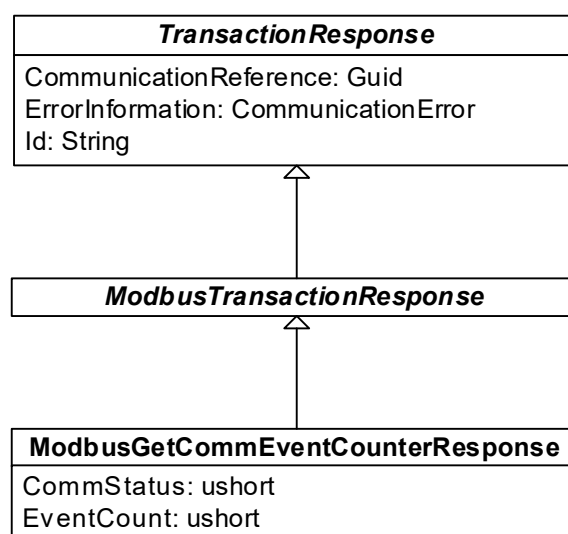
The properties of the ModbusGetCommEventCounterRequest datatype are described in Table 30.

Table 30 – ModbusGetCommEventCounterRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	

10.24 ModbusGetCommEventCounterResponse

This subclause describes the response for the Modbus service Get Comm Event Counter (see Figure 27).



Used in:

ICommunication.EndCommunicationRequest()

Figure 27 – ModbusGetCommEventCounterResponse

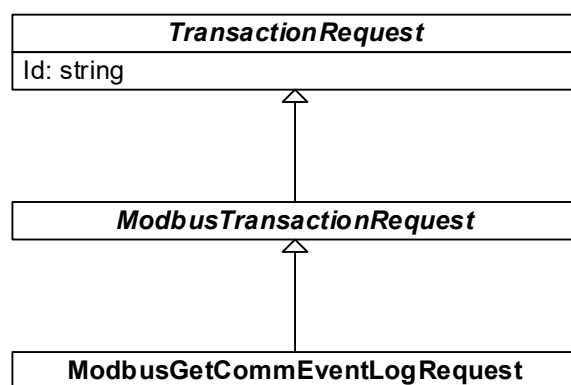
The properties of the ModbusGetCommEventCounterResponse datatype are described in Table 31.

Table 31 – ModbusGetCommEventCounterResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
CommStatus	Two-byte status information. The status information will be 0xFFFF if a previously-issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000.	Status
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
EventCount	Number of events counted by the device.	Event Count
Id	[Optional] Identifier for a single Transaction Request.	

10.25 ModbusGetCommEventLogRequest

This subclause describes the request for the Modbus service Get Comm Event Log (see Figure 28).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 28 – ModbusGetCommEventLogRequest

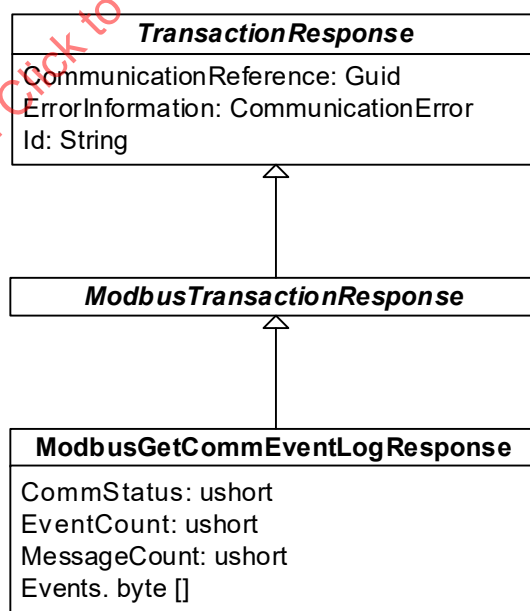
The properties of the ModbusGetCommEventLogRequest datatype are described in Table 32.

Table 32 – ModbusGetCommEventLogRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	

10.26 ModbusGetCommEventLogResponse

This subclause describes the response for the Modbus service Get Comm Event Log (see Figure 29).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 29 – ModbusGetCommEventLogResponse

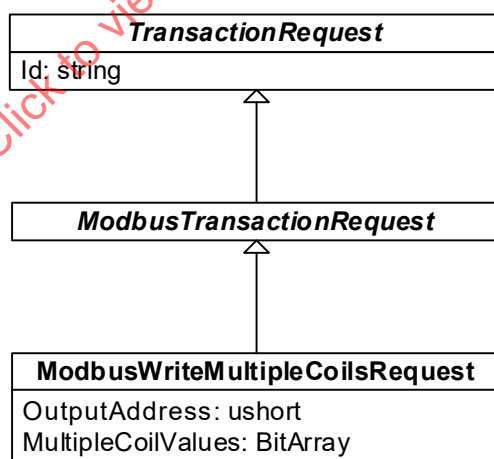
The properties of the ModbusGetCommEventLogResponse datatype are described in Table 33.

Table 33 – ModbusGetCommEventLogResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
CommStatus	Two-byte status information. The status information will be 0xFFFF if a previously-issued program command is still being processed by the remote device (busy condition), otherwise it will be 0x0000.	Status
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Events	Each byte corresponds to the status of one MODBUS send or receive operation for the remote device.	Events
EventCount	Number of events counted by the device.	Event Count
Id	[Optional] Identifier for a single Transaction Request.	
MessageCount	Quantity of messages processed by the remote device since its last restart, clear counters operation, or power-up.	Message Count

10.27 IModbusWriteMultipleCoilsRequest

This subclause describes the request for the Modbus service Write Multiple Coils (see Figure 30).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 30 – ModbusWriteMultipleCoilsRequest

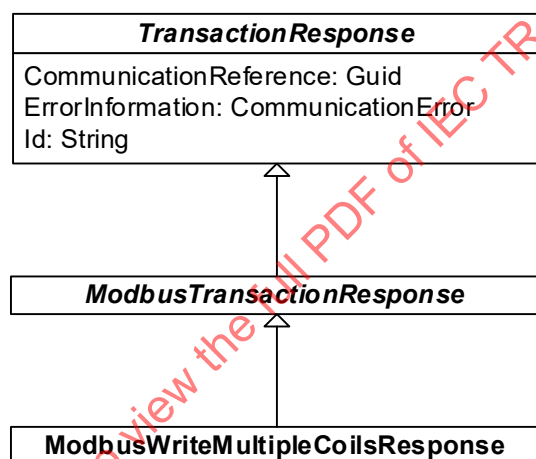
The properties of the ModbusWriteMultipleCoilsRequest datatype are described in Table 34.

Table 34 – ModbusWriteMultipleCoilsRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
MultipleCoilValues	Bit-array with each coil state coded in one bit, where the first bit in the array represents the first coil to be written: "0"=FALSE or "OFF" "1"=TRUE or "ON"	Data
OutputAddress	Address of the first coil to be forced.	Address of first coil

10.28 ModbusWriteMultipleCoilsResponse

This subclause describes the response for the Modbus service Write Multiple Coils (see Figure 31).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 31 – ModbusWriteMultipleCoilsResponse

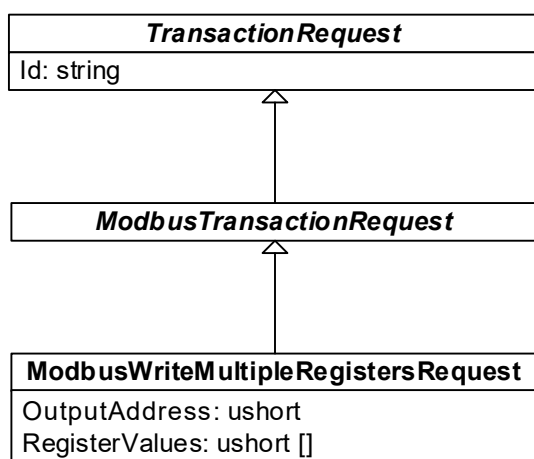
The properties of the ModbusWriteMultipleCoilsResponse datatype are described in Table 35.

Table 35 – ModbusWriteMultipleCoilsResponse datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.29 ModbusWriteMultipleRegistersRequest

This subclause describes the request for the Modbus service Write Multiple Registers (see Figure 32).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 32 – ModbusWriteMultipleRegistersRequest

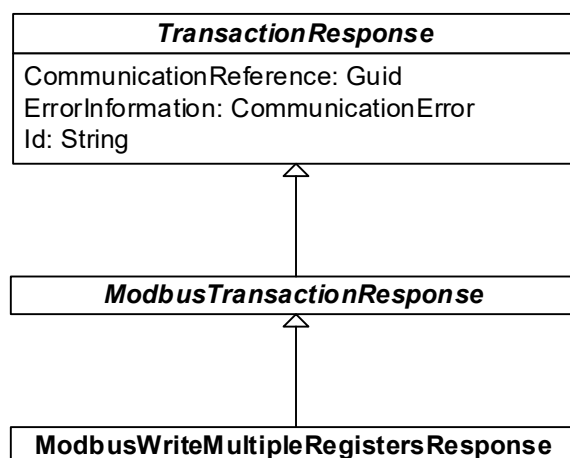
The properties of the ModbusWriteMultipleRegistersRequest datatype are described in Table 36.

Table 36 – ModbusWriteMultipleRegistersRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
OutputAddress	Address of the first coil to be forced.	Address of first coil
RegisterValues	Register values to be written.	Data

10.30 ModbusWriteMultipleRegistersResponse

This subclause describes the response for the Modbus service Write Multiple Registers (see Figure 33).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 33 – ModbusWriteMultipleRegistersResponse

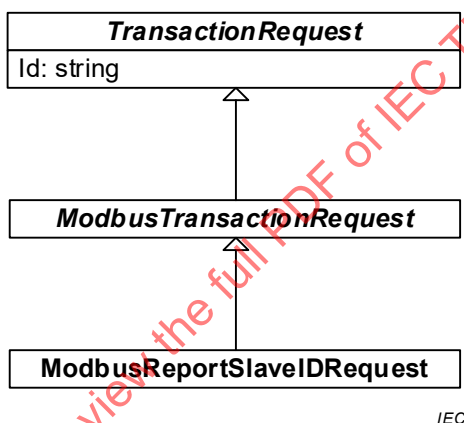
The properties of the ModbusWriteMultipleRegistersResponse datatype are described in Table 37.

Table 37 – ModbusWriteMultipleRegistersResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.31 ModbusReportSlaveIDRequest

This subclause describes the request for the Modbus service Report Slave ID (see Figure 34).



Used in:

ICommunication.BeginCommunicationRequest()

Figure 34 – ModbusReportSlaveIDRequest

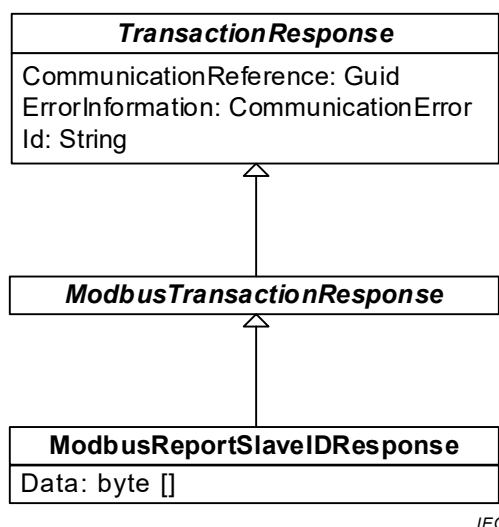
The properties of the ModbusReportSlaveIDRequest datatype are described in Table 38.

Table 38 – ModbusReportSlaveIDRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	

10.32 ModbusReportSlaveIDResponse

This subclause describes the response for the Modbus service Report Slave ID (see Figure 35).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 35 – ModbusReportSlaveIDResponse

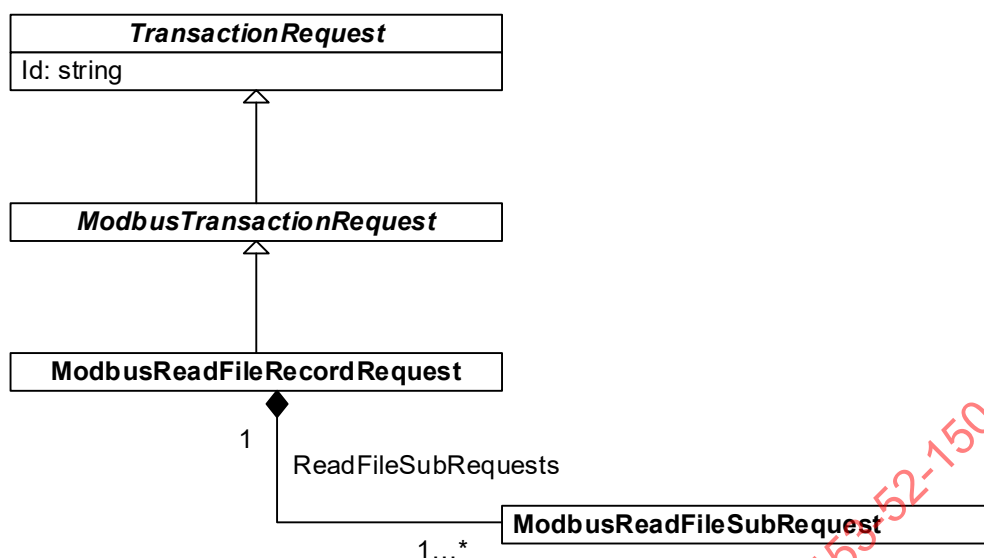
The properties of the ModbusReportSlaveIDResponse datatype are described in Table 39.

Table 39 – ModbusReportSlaveIDResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
Data	This attribute contains the: – Slave ID; – the Run Indicator Status (0x00 or 0xFF); – and the additional device specific data; in the same format and order as defined in the MODBUS Application Protocol Specification.	Slave ID, Run Indicator Status Additional data
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.33 ModbusReadFileRecordRequest

This subclause describes the request for the Modbus service Read File Record (see Figure 36).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 36 – ModbusReadFileRecordRequest

The properties of the ModbusReadFileRecordRequest datatype are described in Table 40.

Table 40 – ModbusReadFileRecordRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
ModbusReadFileSubRequest	Sub-request equivalent to the Read File Record sub-request element defined in IEC 61158-5-15.	

10.34 ModbusReadFileSubRequest

This subclause describes the sub-request for the Modbus service Read File Record (see Figure 37).

ModbusReadFileSubRequest
ReferenceType: byte FileNumber: ushort RecordNumber: ushort Quantity: ushort

IEC

Used in

ICommunication.BeginCommunicationRequest()

Figure 37 – ModbusReadFileSubRequest

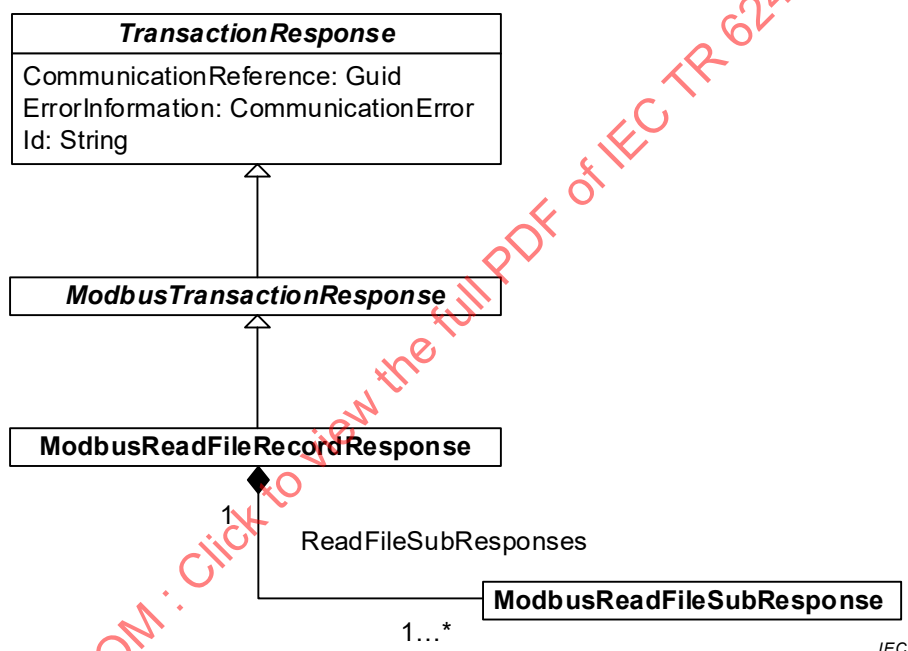
The properties of the ModbusReadFileSubRequest datatype are described in Table 41.

Table 41 – ModbusReadFileSubRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
FileNumber	File number	File number
Id	[Optional] Identifier for a single Transaction Request.	
Quantity	Length of the record to be read as quantity of registers	Record length
RecordNumber	Starting record number within the file.	Record number
ReferenceType	Reference type	Reference type

10.35 ModbusReadFileRecordResponse

This subclause describes the response for the Modbus service Read File Record (see Figure 38).



Used in:

ICommunication.EndCommunicationRequest()

Figure 38 – ModbusReadFileRecordResponse

The properties of the ModbusReadFileRecordResponse datatype are described in Table 42.

Table 42 – ModbusReadFileRecordResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
ReadFileSubResponse	Sub-response equivalent to the Read File Record sub-response element defined in IEC 61158-5-15.	

10.36 ModbusReadFileSubResponse

This subclause describes the sub-response for the Modbus service Read File Record (see Figure 39).

ModbusReadFileSubResponse
RecordData: ushort[]

IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 39 – ModbusReadFileSubResponse

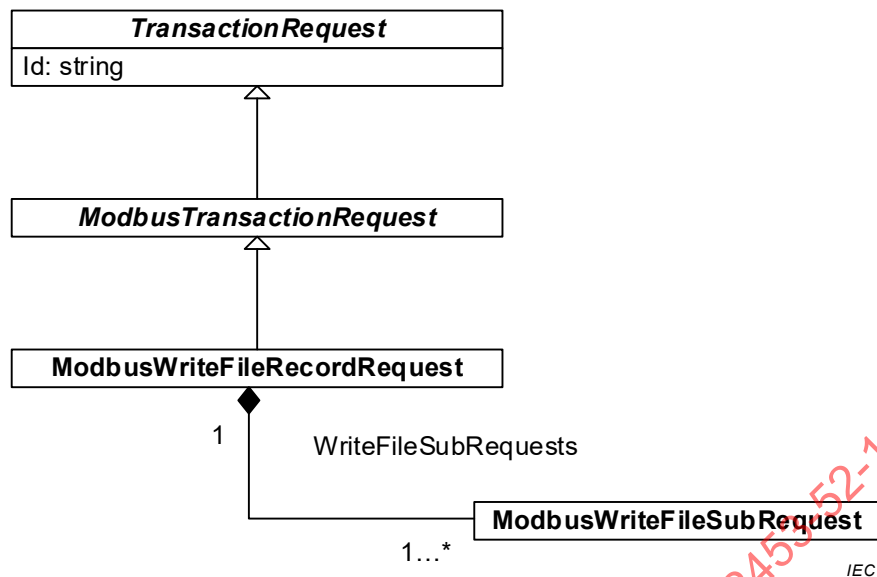
The properties of the ModbusReadFileSubResponse datatype are described in Table 43.

Table 43 – ModbusReadFileSubResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
RecordData	Record data	Record data array

10.37 ModbusWriteFileRecordRequest

This subclause describes the request for the Modbus service Write File Record (see Figure 40).



Used in:
 ICommunication.BeginCommunicationRequest()

Figure 40 – ModbusWriteFileRecordRequest

The properties of the ModbusWriteFileRecordRequest datatype are described in Table 44.

Table 44 – ModbusWriteFileRecordRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
ModbusWriteFileSubRequest	Reference type	Sub-request equivalent to the Write File Record sub-request element defined in IEC 61158-5-15

10.38 ModbusWriteFileSubRequest

This subclause describes the sub-request for the Modbus service Write File Record (see Figure 41).

ModbusWriteFileSubRequest
ReferenceType: byte FileNumber: ushort RecordNumber: ushort RecordData: ushort[]

Used in:
 ICommunication.BeginCommunicationRequest()

Figure 41 – ModbusWriteFileSubRequest

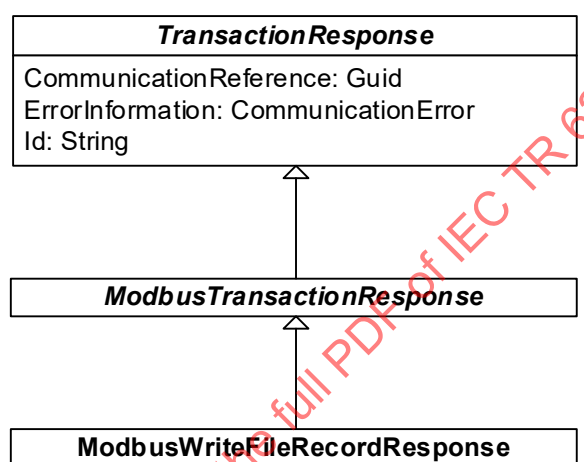
The properties of the ModbusWriteFileSubRequest datatype are described in Table 45.

Table 45 – ModbusWriteFileSubRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
FileNumber	File number	File number
RecordData	Record data	Record data array
RecordNumber	Starting record number within the file	Record number
ReferenceType	Reference type	Reference type

10.39 ModbusWriteFileRecordResponse

This subclause describes the response for the Modbus service Write File Record (see Figure 42).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 42 – ModbusWriteFileRecordResponse

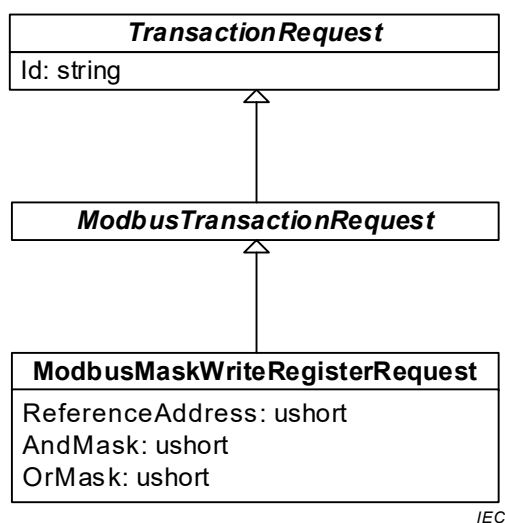
The properties of the ModbusWriteFileRecordResponse datatype are described in Table 46.

Table 46 – ModbusWriteFileRecordResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.40 ModbusMaskWriteRegisterRequest

This subclause describes the request for the Modbus service Mask Write Register (see Figure 43).



Used in:

ICommunication.BeginCommunicationRequest()

Figure 43 – ModbusMaskWriteRegisterRequest

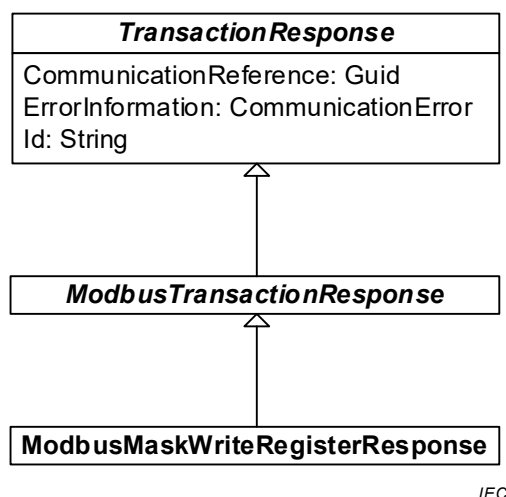
The properties of the ModbusMaskWriteRegisterRequest datatype are described in Table 47.

Table 47 – ModbusMaskWriteRegisterRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
AndMask	AND mask	AND Mask
Id	[Optional] Identifier for a single Transaction Request.	
OrMask	OR mask	OR Mask
ReferenceAddress	Address of the holding register the mask shall be applied to.	Address of first holding register to write

10.41 ModbusMaskWriteRegisterResponse

This subclause describes the response for the Modbus service Mask Write Register (see Figure 44).

**Used in:**

ICommunication.EndCommunicationRequest()

Figure 44 – ModbusMaskWriteRegisterResponse

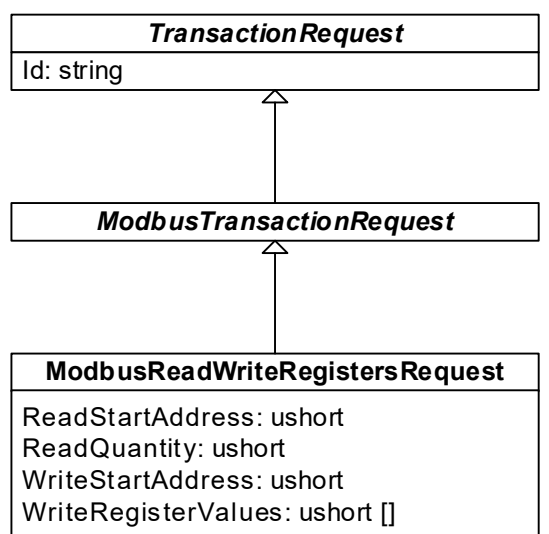
The properties of the ModbusMaskWriteRegisterResponse datatype are described in Table 48.

Table 48 – ModbusMaskWriteRegisterResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	

10.42 ModbusReadWriteRegistersRequest

This subclause describes the request for the Modbus service Read/Write Multiple Registers (see Figure 45).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 45 – ModbusReadWriteRegistersRequest

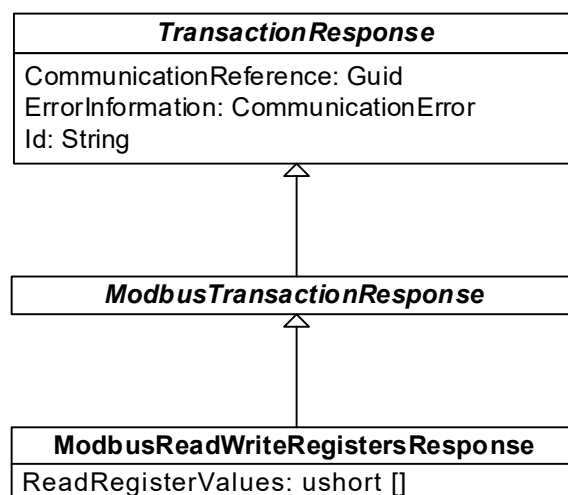
The properties of the ModbusReadWriteRegistersRequest datatype are described in Table 49.

Table 49 – ModbusReadWriteRegistersRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request	
ReadQuantity	Number of holding registers to be read	Quantity of holding registers to read
ReadStartAddress	Address of the first holding register to be read	Address of first holding register to read
WriteRegisterValues	Register values to be written	Data
WriteStartAddress	Address of the first holding register to be written	Address of first holding register to write

10.43 ModbusReadWriteRegistersResponse

This subclause describes the response for the Modbus service Read/Write Multiple Registers (see Figure 46).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 46 – ModbusReadWriteRegistersResponse

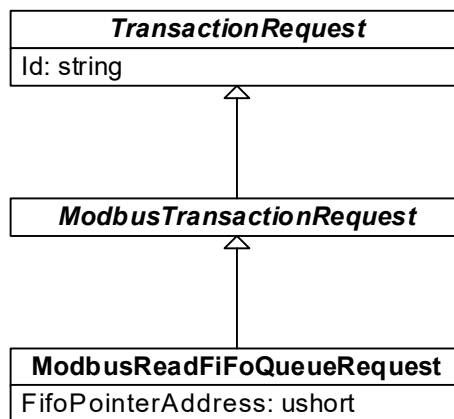
The properties of the ModbusReadWriteRegistersResponse datatype are described in Table 50.

Table 50 – ModbusReadWriteRegistersResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
ReadRegisterValues	Read holding register values.	Data

10.44 ModbusReadFiFoQueueRequest

This subclause describes the request for the Modbus service Read FiFo Queue (see Figure 47).



IEC

Used in:

ICommunication.BeginCommunicationRequest()

Figure 47 – ModbusReadFiFoQueueRequest

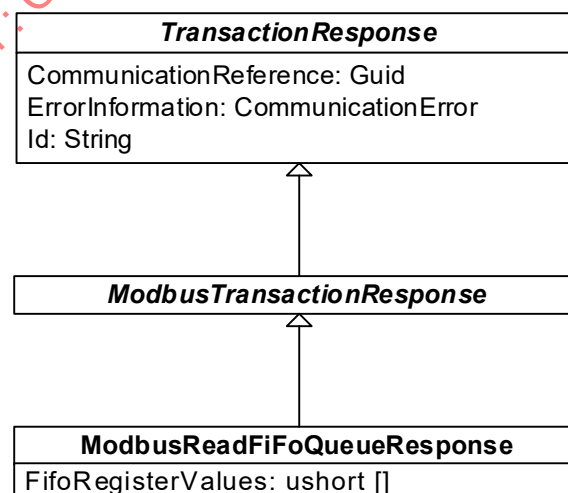
The properties of the ModbusReadFiFoQueueRequest datatype are described in Table 51.

Table 51 – ModbusReadFiFoQueueRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
FifoPointerAddress	Address of the first register to be read in a First-In-First-Out (FIFO) queue of registers.	Address of FIFO queue

10.45 ModbusReadFiFoQueueResponse

This subclause describes the response for the Modbus service Read FIFO Queue (see Figure 48).



IEC

Used in:

ICommunication.EndCommunicationRequest()

Figure 48 – ModbusReadFiFoQueueResponse

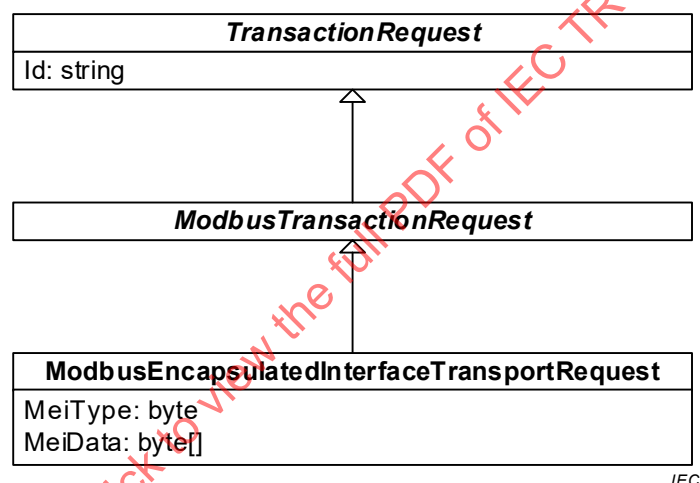
The properties of the ModbusReadFiFoQueueResponse datatype are described in Table 52.

Table 52 – ModbusReadFiFoQueueResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
FifoRegisterValues	Register values read from the FIFO queue Data.	

10.46 ModbusEncapsulatedInterfaceTransportRequest

This subclause describes the request for the Modbus service Encapsulated Interface Transport (see Figure 49).



Used in:

ICommunication.BeginCommunicationRequest()

Figure 49 – ModbusEncapsulatedInterfaceTransportRequest

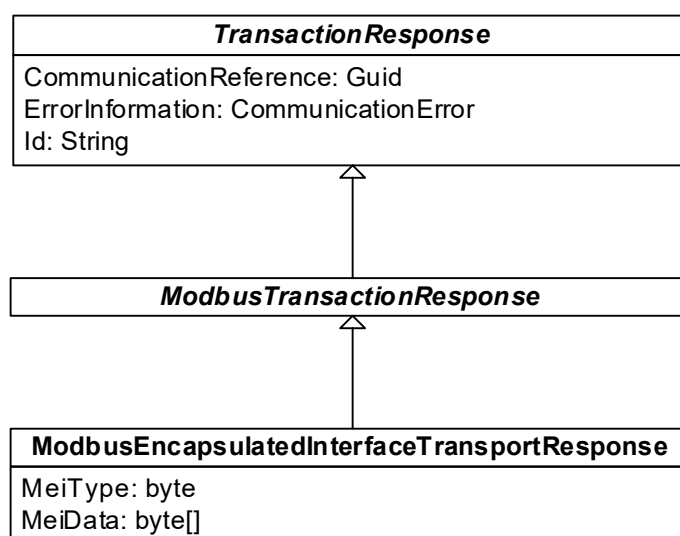
The properties of the ModbusEncapsulatedInterfaceTransportRequest datatype are described in Table 53.

Table 53 – ModbusEncapsulatedInterfaceTransportRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
MeiData	Modbus encapsulated interface type specific data.	MEI type specific data
MeiType	Modbus encapsulated interface type (MEI type).	MEI type

10.47 ModbusEncapsulatedInterfaceTransportResponse

This subclause describes the response for the Modbus service Encapsulated Interface Transport (see Figure 50).



Used in:

ICommunication.EndCommunicationRequest()

Figure 50 – ModbusEncapsulatedInterfaceTransportResponse

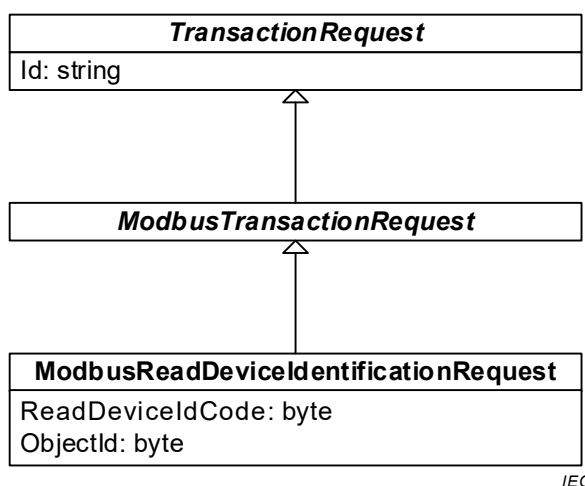
The properties of the **ModbusEncapsulatedInterfaceTransportResponse** datatype are described in Table 54.

Table 54 – ModbusEncapsulatedInterfaceTransportResponse

Property	Description	Equivalent IEC 61158-5-15 service parameter
CommunicationReference	Identifier for a communication link to a device.	CommunicationReference
ErrorInformation	[Optional] Description of a fieldbus protocol independent error occurred during communication.	
Id	[Optional] Identifier for a single Transaction Request.	
MeiData	Modbus encapsulated interface type specific data.	MEI type specific data
MeiType	Modbus encapsulated interface type (MEI type).	MEI type

10.48 ModbusReadDeviceIdentificationRequest

This subclause describes the request for the Modbus service Read Device Identification (see Figure 51).

**Used in:**

ICommunication.BeginCommunicationRequest()

Figure 51 – ModbusReadDeviceIdentificationRequest

The properties of the ModbusReadDeviceIdentificationRequest datatype are described in Table 55.

Table 55 – ModbusReadDeviceIdentificationRequest datatype

Property	Description	Equivalent IEC 61158-5-15 service parameter
Id	[Optional] Identifier for a single Transaction Request.	
ReadDeviceIDCode	The parameter "Read Device ID code" allows to define four access types: 01: request to get the basic device identification (stream access) 02: request to get the regular device identification (stream access) 03: request to get the extended device identification (stream access) 04: request to get one specific identification object (individual access)	Read device ID code
ObjectID	Identification of the first object to obtain.	Requested object ID

10.49 ModbusReadDeviceIdentificationResponse

This subclause describes the response for the Modbus service Read Device Identification (see Figure 52 and Figure 53).