PD CLC/TR IEC 62453-42:2019

BSI Standards Publication

Field device tool (FDT) interface specification

Part 42: Object model integration profile – Common Language Infrastructure (IEC TR 62453-42:2016)

bsi.

National foreword

This Published Document is the UK implementation of CLC/TR IEC 62453-42:2019. It is identical to IEC TR 62453-42:2016. It supersedes PD IEC/TR 62453-42:2016, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee GEL/65/3, Industrial communications: process measurement and control, including fieldbus.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2019

ISBN 978 0 539 02868 3

ICS 25.040.40; 35.100.05; 35.110

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 May 2016.

Amendments/corrigenda issued since publication

Date	Text affected
31 May 2019	This corrigendum renumbers PD IEC/TR 62453-42:2016 as PD CLC/TR IEC 62453-42:2019

TECHNICAL REPORT

RAPPORT TECHNIQUE

TECHNISCHER BERICHT

CLC/TR IEC 62453-42

April 2019

ICS 35.100.05; 25.040.40; 35.110

English Version

Field device tool (FDT) interface specification - Part 42: Object model integration profile - Common Language Infrastructure (IEC TR 62453-42:2016)

Spécification des interfaces des outils des dispositifs de terrain (FDT) - Partie 42: Profil d'intégration des modèles d'objets - Infrastructure commune de langage (IEC TR 62453-42:2016)

Field Device Tool (FDT)-Schnittstellenspezifikation - Teil 42: Profil zur Integration des Objektmodells - Common Language Infrastructure (CLI) (IEC TR 62453-42:2016)

This Technical Report was approved by CENELEC on 2019-03-18.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

This document (CLC/TR IEC 62453-42:2019*)* consists of the text of the IEC TR 62453-42:2016 prepared by 65E: "Devices and integration in enterprise systems", of IEC technical committee 65: "Industrial process measurement, control and automation".

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights.

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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.

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

Publication	Year	Title	EN/HD	Year
IEC 62453-1	2016	Field device tool (FDT) interface specification - Part 1: Overview and guidance	EN 62453-1	2017
IEC 62453-2	2016	Field device tool (FDT) interface specification - Part 2: Concepts and detailed description	EN 62453-2	2017

This page deliberately left blank

CONTENTS

FOREWORD....19
INTRODUCTION....21
1 Scope....23
2 Normative references....23
3 Terms, definitions, abbreviations and conventions....23
3.1 Terms and definitions....23
3.2 Abbreviations....30
3.3 Conventions....30
4 Implementation concept....31
4.1 Technological orientation....31
4.2 Implementation of abstract FDT object model....31
4.3 FDT Frame Application (FA)....32
4.4 DTM Business Logic....33
4.4.1 General....33
4.4.2 Implementation of DTM, DTM Device Type, and Device Ident Info....34
4.4.3 Implementation of DTM device parameter access....35
4.4.4 Process Data Info....35
4.4.5 Diagnostic Data Info....36
4.4.6 Network Management Info....36
4.4.7 Function Info....37
4.4.8 Report Info....37
4.4.9 Document Reference Info....37
4.5 Implementation of DTM Functions....37
4.5.1 DTM User Interface....37
4.5.2 Function access control....38
4.5.3 Handling of standard UI elements in modeless DTM UI interfaces....38
4.5.4 Command functions....39
4.6 User management....39
4.6.1 General....39
4.6.2 Multi-user access....39
4.6.3 User levels....39
4.7 Implementation of FDT and system topology....42
4.7.1 General....42
4.7.2 Topology management....43
4.7.3 Data exchange between Frame Applications....45
4.8 Implementation of Modularity....45
4.9 Implementation of FDT communication....45
4.9.1 Handling of communication requests....45
4.9.2 Handling of communication errors....46
4.9.3 Handling of loss of connection....46
4.9.4 Point–to-point communication....46
4.9.5 Nested communication....47
4.9.6 Dynamic changes in network....47
4.10 Identification....48
4.10.1 DTM instance identification....48
4.10.2 Hardware identification....48
4.11 Implementation of DTM data persistence and synchronization....49

4.11.1 Persistence overview 49
4.11.2 Relations of DTMDataSet 50
4.11.3 DTMDataSet structure 51
4.11.4 Types of persistent DTM data 52
4.11.5 Data synchronization 52
4.12 Implementation of access to device data and IO information 53
4.12.1 Exposing device data and IO information 53
4.12.2 Data access control 54
4.12.3 Routed IO information 56
4.12.4 Comparison of DTM and device data 56
4.12.5 Support for multirole devices 57
4.13 Clone of DTM instances 58
4.13.1 General 58
4.13.2 Replicating a part of topology with Parent DTM and a subset of its Child DTMs 58
4.13.3 Cloning of a DTM without its children 58
4.13.4 Delayed cloning 58
4.14 Lifecycle concepts 59
4.15 Audit trail 59
4.15.1 General 59
4.15.2 Audit trail events 59
5 Technical concepts 60
5.1 General 60
5.2 Support of .NET Common Language Runtime versions 62
5.2.1 General 62
5.2.2 Rules for FDT .NET assemblies 62
5.2.3 DTM rules 62
5.2.4 Frame Application rules 62
5.2.5 FDT CLR extension concept 63
5.3 Support for 32-bit and 64-bit target platforms 63
5.4 Object activation and deactivation 64
5.4.1 General 64
5.4.2 Assembly loading and object creation 64
5.4.3 Assembly dependencies 65
5.4.4 Shared assemblies 65
5.4.5 Object deactivation and unloading 66
5.5 Datatypes 67
5.5.1 General 67
5.5.2 Serialization / deserialization 67
5.5.3 Support of XML 68
5.5.4 Optional elements 68
5.5.5 Verify 68
5.5.6 Clone 68
5.5.7 Equals 69
5.5.8 Lists 69
5.5.9 Nullable 70
5.5.10 Enumeration 70
5.5.11 Protocol-specific datatypes 70
5.5.12 Custom datatypes 72

5.6 General object interaction 73
5.6.1 General 73
5.6.2 Decoupling of FDT Objects 73
5.6.3 Parameter interchange with .NET datatypes 74
5.6.4 Interaction patterns 74
5.6.5 Properties 74
5.6.6 Synchronous methods 74
5.6.7 Asynchronous methods 75
5.6.8 Events pattern 81
5.6.9 Exception handling 82
5.7 Threading 86
5.7.1 Introduction 86
5.7.2 Threading rules 87
5.8 Localization support 88
5.8.1 General 88
5.8.2 Access to localized resources and culture-dependent functions 89
5.8.3 Handling of cultures 89
5.8.4 Switching the User Interface language 90
5.9 DTM User Interface implementation 90
5.9.1 General 90
5.9.2 Resizing 90
5.9.3 Private dialogs 92
5.10 DTM User Interface hosting 92
5.10.1 General 92
5.10.2 Hosting DTM WPF controls 92
5.10.3 Hosting DTM WinForms controls 93
5.11 Static Function implementation 94
5.12 Persistence 96
5.12.1 Overview 96
5.12.2 Data format 97
5.12.3 Adding / reading / writing / deleting of data 97
5.12.4 Searching for data 99
5.13 Comparison of DTM and device data 100
5.13.1 Comparison of datasets using IDeviceData / IInstanceData 100
5.13.2 Comparison of datasets using IComparison 101
5.14 Tracing 101
5.15 Report generation 101
5.15.1 General 101
5.15.2 Report types 102
5.15.3 DTM report data format 102
5.15.4 Report data exchange 103
5.16 Security 103
5.16.1 General 103
5.16.2 Strong naming of assemblies 103
5.16.3 Identification of origin 104
5.16.4 Code access security 104
5.16.5 Validation of FDT compliance certification 104
6 FDT Objects and interfaces 106
6.1 General 106

6.2 Frame Application 107
6.3 DTM Business Logic 109
6.3.1 DTM BL interfaces 109
6.3.2 State machines related to DTM BL 114
6.3.3 State machine of instance data 120
6.4 DTM User Interface 123
6.5 Communication Channel 124
6.6 Availability of interface methods 125
7 FDT datatypes 126
7.1 General 126
7.2 Datatypes – Base 127
7.3 General datatypes 127
7.4 Datatypes – DtmInfo / TypeInfo 128
7.5 Datatypes – DeviceIdentInfo 130
7.6 Datatypes for installation and deployment 135
7.6.1 Datatypes – SetupManifest 135
7.6.2 Datatypes – DtmManifest 136
7.6.3 Datatypes – DtmUiManifest 137
7.7 Datatypes – Communication 137
7.8 Datatypes – BusCategory 143
7.9 Datatypes – Device / Instance Data 143
7.9.1 General 143
7.9.2 Datatypes used in reading and writing DeviceData 150
7.10 Datatypes for export and import 152
7.10.1 Datatypes – TopologyImportExport 152
7.10.2 Datatypes – ImportExportDataset 153
7.11 Datatypes for process data description 154
7.11.1 Datatypes – ProcessDataInfo 154
7.11.2 Datatypes – Process Image 159
7.12 Datatypes – Address information 160
7.13 Datatypes – NetworkDataInfo 164
7.14 Datatypes – DTM functions 166
7.15 Datatypes – DTM messages 168
7.16 Datatypes for delegation of DTM UI dialog actions 170
7.17 Datatypes – CommunicationChannelInfo 170
7.18 Datatypes – HardwareIdentification and scanning 172
7.18.1 General 172
7.18.2 Datatypes – DeviceScanInfo 172
7.18.3 Example – HardwareIdentification and scanning for HART 173
7.19 Datatypes – DTM report types 174
7.20 Information related to device modules in a monolithic DTM 174
8 Workflows 176
8.1 General 176
8.2 Instantiation, loading and release 176
8.2.1 Finding a DTM BL object 176
8.2.2 Instantiation of a new DTM BL 178
8.2.3 Configuring access rights 180
8.2.4 Loading a DTM BL 181
8.2.5 Loading a DTM with Expert user level 182

8.2.6 Release of a DTM BL 183
8.3 Persistent storage of a DTM 184
8.3.1 Saving instance data of a DTM 184
8.3.2 Copy and versioning of a DTM instance 185
8.3.3 Dataset commit failed 186
8.3.4 Export a DTM dataset to file 186
8.4 Locking and DataTransactions in multi-user environments 187
8.4.1 General 187
8.4.2 Propagation of changes 188
8.4.3 Synchronizing DTMs in multi-user environments 190
8.5 Execution of DTM Functions 191
8.5.1 General 191
8.5.2 Finding a DTM User Interface object 191
8.5.3 Instantiation of an integrated DTM graphical user interface 192
8.5.4 Instantiation of a DTM UI triggered by the DTM BL 193
8.5.5 Instantiation of a modal DTM UI triggered by DTM BL 194
8.5.6 Release of a DTM User Interface 195
8.5.7 Release of a DTM UI triggered by the DTM BL 196
8.5.8 Release of a DTM User Interface triggered by itself 197
8.5.9 Release of a non-modal DTM User Interface triggered by a standard action 198
8.5.10 Progress indication for prolonged DTM actions 199
8.5.11 Starting an application 200
8.5.12 Terminating applications 201
8.5.13 Execution of command functions 201
8.5.14 Execution of a command function with user interface 201
8.5.15 Opening of documents 202
8.5.16 Interaction between DTM User Interface and DTM Business Logic 203
8.5.17 Interaction between DTM Business Logic and DTM User Interface 205
8.5.18 Interaction between DTM User Interface and DTM Business Logic with Cancel 206
8.5.19 Retrieving information about available Static Functions 207
8.5.20 Executing a Static Function 208
8.5.21 Executing a Static Function with multiple arguments 209
8.6 DTM communication 210
8.6.1 General 210
8.6.2 Establishing a communication connection 211
8.6.3 Cancel establishment of communication connection 212
8.6.4 Communicating with the device 212
8.6.5 Frame Application or Child DTM disconnect a device 213
8.6.6 Terminating a communication connection 214
8.6.7 DTM aborts communication connection 215
8.6.8 Communication Channel aborts communication connection 216
8.7 Nested communication 216
8.7.1 General 216
8.7.2 Communication request for a nested connection 217
8.7.3 Propagation of errors for a nested connection 218
8.8 Topology planning 219
8.8.1 General 219

8.8.2 Adding a DTM to the topology 219
8.8.3 Removing a DTM from topology 220
8.8.4 Frame Application creates topology 221
8.8.5 DTM generates sub-topology 222
8.8.6 Physical Layer and DataLinkLayer 224
8.9 Instantiation, configuration, move and release of Child DTMs 224
8.9.1 General 224
8.9.2 Instantiation and configuration of Child DTM BL 224
8.9.3 Interaction between Parent DTM and Child DTM 225
8.9.4 Interaction between Parent DTM and Child DTM using IDtmMessaging 227
8.9.5 Parent DTM moves a Child DTM 227
8.9.6 Parent DTM removes Child DTM 228
8.10 Topology scan 229
8.10.1 General 229
8.10.2 Scan of network topology 229
8.10.3 Cancel topology scan 230
8.10.4 Scan based DTM assignment 231
8.10.5 Manufacturer-specific device identification 232
8.11 Configuration of communication networks 234
8.11.1 Configuration of a fieldbus master 234
8.11.2 Integration of a passive device 235
8.12 Using IO information 235
8.12.1 Assignment of symbolic name to process data 235
8.12.2 Creation of Process Image 237
8.12.3 Validation of changes in process image while PLC is running 238
8.12.4 Changing of variable names using process image interface 239
8.13 Managing addresses 240
8.13.1 Set DTM address with user interface 240
8.13.2 Set DTM addresses without user interface 241
8.13.3 Display or modify addresses of all Child DTMs with user interface 242
8.14 Device-initiated data transfer 243
8.15 Reading and writing data 244
8.15.1 Read/write instance data 244
8.15.2 Read/write device data 246
8.16 Comparing data 248
8.16.1 Comparing device dataset and instance dataset 248
8.16.2 Comparing different instance datasets 248
8.17 Reassigning a different DtmDeviceType at a device node 249
8.17.1 General 249
8.17.2 DTM detects a change in connected device type 250
8.17.3 Search matching DtmDeviceTypes after incompatible device exchange 252
8.17.4 Reassign DtmDeviceType after incompatible device exchange 253
8.18 Copying part of FDT Topology 255
8.18.1 Cloning of a single DTM without Children 255
8.18.2 Cloning of a DTM with all its Children 256
8.19 Sequences for audit trail 256
8.19.1 General 256
8.19.2 Audit trail of parameter modifications in instance dataset 256
8.19.3 Audit trail of parameter modifications in device dataset 257

8.19.4 Audit trail of function calls 258
8.19.5 Audit trail of general notification 259
9 Installation 259
9.1 General 259
9.2 Common rules 259
9.2.1 Predefined installation paths 259
9.2.2 Manifest files 262
9.2.3 Paths in manifest files 263
9.2.4 Common command line arguments 263
9.2.5 Digital signatures of setup components 264
9.3 Installation of FDT core assemblies 264
9.4 Installation of communication protocols 264
9.4.1 General 264
9.4.2 Registration 264
9.4.3 Protocol manifest 264
9.5 Installation of DTMs 265
9.5.1 General 265
9.5.2 Registration 266
9.5.3 DTM manifest 267
9.5.4 DTM User Interface manifest 268
9.6 DTM setup 269
9.6.1 Structure 269
9.6.2 DTM setup manifest 270
9.6.3 DTM device identification manifest 271
9.6.4 Setup creation rules 273
9.7 DTM deployment 274
9.8 Paths and file information 276
9.8.1 Path information provided by a DTM 276
9.8.2 Paths and persistence 276
9.8.3 Multi-user systems 276
10 Life cycle concept 276
10.1 General 276
10.2 Technical concept 277
10.2.1 General 277
10.2.2 DtmManifest / DtmInfo 278
10.2.3 TypeInfo 278
10.2.4 Supported DataSet formats 279
10.2.5 DeviceIdentInfo 279
10.2.6 Dataset 280
10.2.7 DeviceScanInfo 280
10.3 DTM setup 280
10.4 Life Cycle Scenarios 281
10.4.1 Overview 281
10.4.2 Search for device type in DTM setups 282
10.4.3 Search for installed DTMs 283
10.4.4 Dataset migration for reassigned DTM 285
11 Frame Application architectures 286
11.1 General 286
11.2 Standalone application 286

11.3 Remoted user Interface 286
11.4 Distributed multi-user application 287
11.5 OPC UA 287
Annex A (normative) FDT2 Use case model 289
A.1 Use case model overview 289
A.2 Actors 289
A.3 Use cases 290
A.3.1 Use case overview 290
A.3.2 Observation use cases 291
A.3.3 Operation use cases 292
A.3.4 Maintenance use cases 294
A.3.5 Planning use cases 299
A.3.6 Main Operation 301
A.3.7 OEM Service 302
A.3.8 Administration 302
Annex B (normative) FDT interface definition and datatypes 303
Annex C (normative) Mapping of services to interface methods 304
C.1 General 304
C.2 DTM services 304
C.3 Presentation object services 308
C.4 General channel services 308
C.5 Process channel services 308
C.6 Communication Channel Services 309
C.7 Frame Application Services 310
Annex D (normative) FDT version interoperability guide 313
D.1 Overview 313
D.2 General 313
D.3 Component interoperability 314
Annex E (normative) FDT1.2.x / IEC 62453-42 Backward-Compatibility 315
E.1 Overview 315
E.2 Parallel FDT topologies 315
E.3 Mixed FDT topologies 316
E.4 FDT1.2.x / IEC 62453-42 Adapters 318
E.5 FDT1.2.x XML / IEC TR 62453-42 Datatype Transformers 319
E.5.1 General 319
E.5.2 Installation and Registration of Protocol-specific Transformers 320
E.5.3 Interaction between FDT2 and FDT1.2 components using Transformers 321
E.6 Sequences related to backward compatibility 322
E.6.1 General 322
E.6.2 Dataset migration from FDT1.x DTM to FDT2.x DTM 322
Annex F (informative) Implementation Hints 324
F.1 IAsyncResult pattern 324
F.2 Threading Best Practices 325
Annex G (informative) Trade names 326
Annex H (informative) UML Notation 327
H.1 General 327
H.2 Class diagram 327
H.3 Statechart diagram 330

H.4 Use case diagram 331
H.5 Sequence diagram 332
H.6 Object diagram 336
Annex I (informative) Physical Layer Examples 337
I.1 General 337
I.2 Interbus S 337
I.3 PROFIBUS 337
I.4 PROFINET 337
Annex J (informative) Predefined SemanticIds 339
J.1 General 339
J.2 Data 339
J.3 Images 339
J.4 Documents 339
Bibliography 341

Figure 1 – Relation of IEC 62453-42 to the IEC 62453 series 21
Figure 2 – IEC 62453-42 Object Model 32
Figure 3 – Frame Application 32
Figure 4 – DTM Business Logic 34
Figure 5 – DTM, Device Type and Device Ident Info 35
Figure 6 – Process Data Info 36
Figure 7 – Logical topology and physical topology 43
Figure 8 – FDT and logical topology 43
Figure 9 – DTMs and physical topology 44
Figure 10 – Point–to-point communication 46
Figure 11 – Nested communication 47
Figure 12 – Identification of connected devices 49
Figure 13 – FDT storage and synchronization mechanism 50
Figure 14 – Relation between DTMDataSet, DTM instance, and device 50
Figure 15 – DTMDataSet structure 51
Figure 16 – Data Synchronization 53
Figure 17 – Routed IO information 56
Figure 18 – Multirole Device 57
Figure 19 – FDT .NET Assemblies 60
Figure 20 – FDT Object implementation 61
Figure 21 – FDT CLR extension concept 63
Figure 22 – Example: Assembly.LoadFrom() 64
Figure 23 – Example: Assembly dependencies 65
Figure 24 – Example: Datatype definition 67
Figure 25 – Example: Data cloning 69
Figure 26 – Example: Methods without data cloning 69
Figure 27 – Protocol-specific datatypes 70
Figure 28 – Protocol manifest and type info attributes 71
Figure 29 – Example: Protocol assembly attributes 72

Figure 30 – Example: Handling of protocol-specific assemblies in Frame Application....72
Figure 31 – Decoupled FDT Objects in IEC 62453-42....73
Figure 32 – IAsyncResult pattern: blocking call....76
Figure 33 – Example: Blocking use of asynchronous interface....76
Figure 34 – IAsyncResult pattern (simplified): blocking call....77
Figure 35 – IAsyncResult pattern: non-blocking call....77
Figure 36 – Example: Non-blocking use of asynchronous interface....78
Figure 37 – IAsyncResult pattern (simplified depiction): non-blocking call....78
Figure 38 – IAsyncResult pattern: canceling an operation....80
Figure 39 – IAsyncResult pattern: providing progress events....81
Figure 40 – Frame Application's host window providing scroll bars....91
Figure 41 – Control using internal scrollbars....91
Figure 42 – Example: Hosting a DTM WPF control in a WPF Frame Application....93
Figure 43 – Example: Hosting a DTM WPF control in a WinForms Frame Application....93
Figure 44 – Example: Hosting DTM WinForms controls in a WinForms Frame Application....94
Figure 45 – Example: Hosting a DTM WinForms control in a WPF Frame Application....94
Figure 46 – Relation of StaticFunctionDescription to Static Function....95
Figure 47 – DTMDataset structure....96
Figure 48 – Example: Initialization of DTMDataSubset with DTM data....98
Figure 49 – Example: Writing of DTM data in DTMDataSubset....98
Figure 50 – Example: Reading of DTM data from a DTMDataSubset....99
Figure 51 – Example: Creation of a BulkData.DTMDataSubset with descriptor....100
Figure 52 – Example: Searching for DTMDataSubsets with specific descriptor....100
Figure 53 – Skeleton of a DTM-specific report fragment....103
Figure 54 – Example: Authenticode check....104
Figure 55 – Example: Conformity record file....105
Figure 56 – Example: checking conformity record file....106
Figure 57 – Frame Application interfaces....107
Figure 58 – DTM Business Logic interfaces (Part 1)....110
Figure 59 – DTM Business Logic interfaces (Part 2)....111
Figure 60 – State machine of DTM BL....115
Figure 61 – Online state machine of DTM....117
Figure 62 – Modifications of data through a DTM....120
Figure 63 – ModifiedInDtm: State machine of instance data....121
Figure 64 – ModifiedInDevice: State machine related to device data....122
Figure 65 – DTM UI interfaces....123
Figure 66 – Communication Channel interfaces....124
Figure 67 – FdtDatatype and FdtList....127
Figure 68 – DtmInfo / TypeInfo – datatypes....129
Figure 69 – DeviceIdentInfo – datatypes....131
Figure 70 – DeviceIdentInfo – Example for HART....132
Figure 71 – Example: DeviceIdentInfo creation....134

Figure 72 – Example: Using DeviceIdentInfo 135
Figure 73 – Example: DeviceIdentInfoTypeAttribute 135
Figure 74 – SetupManifest – datatypes 135
Figure 75 – DtmManifest – datatypes 136
Figure 76 – DtmUiManifest – datatypes 137
Figure 77 – Communication datatypes – Connect 138
Figure 78 – Communication datatypes – Transaction 138
Figure 79 – Communication datatypes – Disconnect 139
Figure 80 – Communication datatypes – Subscribe 139
Figure 81 – Communication datatypes – Scanning 140
Figure 82 – Communication datatypes – Address setting 140
Figure 83 – Example: Communication – Connect for HART 142
Figure 84 – Example: Communication – CommunicationType for HART 143
Figure 85 – BusCategory – datatypes 143
Figure 86 – Device / Instance data – datatypes 144
Figure 87 – Example: Providing information on data of a HART device 146
Figure 88 – Example: Providing information on module data of a PROFIBUS device 147
Figure 89 – Example: Providing information on data 148
Figure 90 – Example: Providing information on structured data 149
Figure 91 – EnumInfo – datatype 150
Figure 92 – Read and Write Request – datatypes 150
Figure 93 – ResponseInfo – datatype 151
Figure 94 – TopologyImportExport – datatypes 152
Figure 95 – ImportExportDataset – datatypes 153
Figure 96 – ProcessDataInfo – datatypes 154
Figure 97 – IOSignalInfo – datatypes 155
Figure 98 – Example: ProcessDataInfo for HART (UML) 157
Figure 99 – Example: ProcessDataInfo creation for HART 158
Figure 100 – Example: Using ProcessData for HART 159
Figure 101 – Example: IOSignalInfoType attribute 159
Figure 102 – ProcessImage – datatypes 160
Figure 103 – AddressInfo – datatypes 161
Figure 104 – Example: AddressInfo creation 162
Figure 105 – Example: Using AddressInfo 163
Figure 106 – Example: DeviceAddressTypeAttribute 163
Figure 107 – NetworkDataInfo – datatypes 164
Figure 108 – Example: NetworkDataInfo creation example 165
Figure 109 – Example: NetworkDataInfo using example 166
Figure 110 – Example: NetworkDataTypeAttribute example 166
Figure 111 – DTM Function – datatypes 167
Figure 112 – DTM Messages – datatypes 169
Figure 113 – ActionItem – datatypes 170
Figure 114 – CommunicationChannelInfo – datatypes 170

Figure 115 – Example: Channel information 171
Figure 116 – DeviceScanInfo – datatypes 172
Figure 117 – Example: HARTDeviceScanInfo – datatype 173
Figure 118 – DTM Report – datatypes 174
Figure 119 – Information related to device modules 175
Figure 120 – Finding a DTM BL object 177
Figure 121 – Instantiation of a new DTM BL 179
Figure 122 – Configuration of user permissions 181
Figure 123 – Loading a DTM BL 182
Figure 124 – Loading a DTM with Expert user level 183
Figure 125 – Release of a DTM BL 184
Figure 126 – Saving data of a DTM 185
Figure 127 – Dataset commit failed 186
Figure 128 – Export a DTM dataset to file 187
Figure 129 – Propagation of changes 189
Figure 130 – Synchronizing DTMs in multi-user environments 190
Figure 131 – Finding a DTM User Interface 192
Figure 132 – Instantiation of a DTM User Interface 193
Figure 133 – Instantiation of a DTM UI triggered by DTM BL 194
Figure 134 – Instantiation of a modal DTM UI triggered by DTM BL 195
Figure 135 – Release of a DTM User Interface 196
Figure 136 – Release of a DTM UI triggered by the DTM BL 197
Figure 137 – Release of a DTM User Interface triggered by itself 198
Figure 138 – Release of a non-modal DTM UI triggered by a standard action 198
Figure 139 – Progress indication for prolonged DTM actions 199
Figure 140 – Starting an application 200
Figure 141 – Execute a command function 201
Figure 142 – Execute a command function with user interface 202
Figure 143 – Opening a document 203
Figure 144 – Interaction triggered by the DTM User Interface 204
Figure 145 – Interaction triggered by the DTM Business Logic 205
Figure 146 – Interaction triggered and canceled by the DTM User Interface 206
Figure 147 – Retrieving information about available Static Functions 207
Figure 148 – Example: Information about available Static Functions 208
Figure 149 – Executing a Static Function 209
Figure 150 – Executing a Static Function with multiple Arguments 210
Figure 151 – Establishing a communication connection 211
Figure 152 – DTM cancels ongoing Connect operation 212
Figure 153 – Communicating with the device 213
Figure 154 – Child DTM disconnects 214
Figure 155 – Child DTM terminates a connection 215
Figure 156 – Child DTM aborts a connection 215
Figure 157 – Communication Channel aborts a connection 216

Figure 158 – Example: Nested communication behavior 217
Figure 159 – Example: Nested communication data exchange 218
Figure 160 – Add DTM to topology 220
Figure 161 – Removing a DTM from topology 221
Figure 162 – Frame Application creates topology 222
Figure 163 – DTM generates sub-topology 223
Figure 164 – Instantiation and configuration of Child DTM BL 225
Figure 165 – Interaction between Parent DTM and Child DTM 226
Figure 166 – Interaction using IDtmMessaging 227
Figure 167 – Parent DTM moves a Child DTM 228
Figure 168 – Parent DTM removes Child DTM 229
Figure 169 – Scan of network topology 230
Figure 170 – Cancel topology scan 231
Figure 171 – Scan based DTM assignment 232
Figure 172 – Manufacturer-specific device identification 233
Figure 173 – Configuration of a fieldbus master 234
Figure 174 – Integration of a passive device 235
Figure 175 – Assignment of process data 236
Figure 176 – Creation of process image 238
Figure 177 – Validation of changes while PLC is running 239
Figure 178 – Changing of variable names using process image interface 240
Figure 179 – Set DTM address with UI 241
Figure 180 – Set DTM addresses without UI 242
Figure 181 – Display or modify child addresses with UI 243
Figure 182 – Device-initiated data transfer 244
Figure 183 – Read/write instance data 245
Figure 184 – Read/write device data 247
Figure 185 – Comparing device dataset and instance dataset 248
Figure 186 – Compare instance data with persisted dataset 249
Figure 187 – DTM triggers ActiveTypeChanged event 251
Figure 188 – Find matching DtmDeviceTypes after incompatible device exchange 253
Figure 189 – Reassign a DtmDeviceType after incompatible device exchange 254
Figure 190 – Clone DTM without children 255
Figure 191 – Clone DTM with all children 256
Figure 192 – Audit trail of parameter modifications in instance dataset 257
Figure 193 – Audit trail of parameter modifications in device 258
Figure 194 – Audit trail of function calls 258
Figure 195 – GAC and FDT_Registry 261
Figure 196 – Installation paths (with example DTM) 262
Figure 197 – Example: Protocol manifest 265
Figure 198 – Search for installed DTMs 266
Figure 199 – Example: DtmManifest 268
Figure 200 – Example: DtmUiManifest 269

Figure 201 – DTM setup structure 270
Figure 202 – Example: DtmSetupManifest 271
Figure 203 – Example: DeviceIdentManifest 273
Figure 204 – DTM deployment 275
Figure 205 – Overview DTM identification 277
Figure 206 – Identification attributes in DTM setup 281
Figure 207 – Check DTM Setup for list of supported types 283
Figure 208 – Scan installed DTMs 284
Figure 209 – Dataset migration to a reassigned DtmDeviceType 285
Figure 210 – Client / Server Application 286
Figure 211 – Example for distributed multi-user application 287
Figure 212 – OPC UA server based on IEC TR 62453-42 288
Figure A.1 – Main use case diagram 289
Figure A.2 – Observation use cases 291
Figure A.3 – Operation use cases 293
Figure A.4 – Maintenance use cases 295
Figure A.5 – Planning use cases 299
Figure E.1 – Example: IEC TR 62453-42 Frame Application with FDT1.2.x backward-compatibility support 315
Figure E.2 – IEC TR 62453-42 Frame Application with FDT1.2.x Device DTM 316
Figure E.3 – IEC TR 62453-42 Frame Application with FDT1.2.x Comm. and Gateway DTM 317
Figure E.4 – IEC TR 62453-42 Frame Application with FDT1.2.x Gateway DTM 317
Figure E.5 – IEC TR 62453-42 – FDT1.2 interaction using transformer 322
Figure E.6 – Dataset migration from FDT1.x DTM to FDT2.x DTM 323
Figure H.1 – Note 327
Figure H.2 – Class 327
Figure H.3 – Association 327
Figure H.4 – Navigable Association 328
Figure H.5 – Composition 328
Figure H.6 – Aggregation 328
Figure H.7 – Dependency 328
Figure H.8 – Association class 328
Figure H.9 – Abstract class, Generalization and Interface 329
Figure H.10 – Interface related notations 329
Figure H.11 – Multiplicity 330
Figure H.12 – Enumeration datatype 330
Figure H.13 – Elements of UML statechart diagrams 330
Figure H.14 – Example of UML state chart diagram 331
Figure H.15 – UML use case syntax 331
Figure H.16 – UML sequence diagram 332
Figure H.17 – Empty UML sequence diagram frame 332
Figure H.18 – Object with life line and activation 333
Figure H.19 – Method calls 333

Figure H.20 – Modeling guarded call and multiple calls 333
Figure H.21 – Call to itself 334
Figure H.22 – Continuation / StateInvariant 334
Figure H.23 – Alternative fragment 335
Figure H.24 – Option fragment 335
Figure H.25 – Loop combination fragment 335
Figure H.26 – Break notation 335
Figure H.27 – Sequence reference 336
Figure H.28 – Objects 336
Figure H.29 – Object association 336

Table 1 – FDT User levels 40
Table 2 – Role dependent Access Rights and User Interfaces for DTMs 41
Table 3 – Description of properties related to data access control 55
Table 4 – Supported CLR versions 62
Table 5 – Frame Application interfaces 108
Table 6 – DTM Business Logic interfaces 112
Table 7 – Availability of interfaces depending of type of DTM 113
Table 8 – Definition of DTM BL state machine 116
Table 9 – Definition of online state machine 118
Table 10 – Description of instance dataset states 121
Table 11 – Description of dataset states regarding online modifications 122
Table 12 – DTM UI interfaces 124
Table 13 – Communication Channel interfaces 125
Table 14 – Availability of DTM BL methods in different states 125
Table 15 – FDT base datatypes 127
Table 16 – FDT General datatypes 128
Table 17 – DtmInfo datatype description 129
Table 18 – DeviceIdentInfo datatype description 131
Table 19 – DeviceIdentInfo – Example for HART 133
Table 20 – SetupManifest datatype description 136
Table 21 – DtmManifest datatype description 136
Table 22 – DtmUiManifest datatype description 137
Table 23 – Communication datatype description 141
Table 24 – BusCategory datatype description 143
Table 25 – DeviceData datatype description 145
Table 26 – Reading and Writing datatype description 150
Table 27 – Reading and Writing datatype description 151
Table 28 – TopologyImportExport datatype description 153
Table 29 – ImportExportDataset datatype description 153
Table 30 – ProcessDataInfo datatype description 155
Table 31 – IOSignalInfo datatype description 156
Table 32 – ProcessImage datatype description 160

Table 33 – AddressInfo datatype description 161
Table 34 – NetworkDataInfo datatype description 165
Table 35 – DTM Function datatype description 168
Table 36 – DTM Messages datatype description 169
Table 37 – ActionItem datatype description 170
Table 38 – CommunicationChannelInfo datatype description 171
Table 39 – DeviceScanInfo datatype description 172
Table 40 – Example: HARTDeviceScanInfo datatype description 173
Table 41 – Reporting datatype description 174
Table 42 – Predefined FDT installation paths 259
Table 43 – Predefined setup properties 263
Table 44 – Setup command line parameters 263
Table 45 – DTM identification 278
Table 46 – DtmType – user readable description of supported types 278
Table 47 – TypeInfo identification 279
Table 48 – DtmType – Dataset support identification 279
Table 49 – Dataset identification 280
Table 50 – DeviceScanInfo – scanned device identification 280
Table 51 – Setup information 281
Table 52 – Changing DTM–- overview 282
Table A.1 – Actors 290
Table A.2 – Observation use cases 291
Table A.3 – Operation use cases 293
Table A.4 – Maintenance use cases 296
Table A.5 – Planning use cases 299
Table C.1 – General services 304
Table C.2 – DTM services related to installation 304
Table C.3 – DTM service related to DTM Information 304
Table C.4 – DTM services related to DTM state machine 305
Table C.5 – DTM services related to function 305
Table C.6 – DTM services related to documentation 306
Table C.7 – DTM services to access the instance data 306
Table C.8 – DTM services to access diagnosis 306
Table C.9 – DTM services to access to device data 306
Table C.10 – DTM services related to network management information 307
Table C.11 – DTM services related to online operation 307
Table C.12 – DTM services related to FDT-Channel objects 307
Table C.13 – DTM services related to import and export 308
Table C.14 – DTM services related to data synchronization 308
Table C.15 – DTM UI state control 308
Table C.16 – General channel service 308
Table C.17 – Channel services for IO related information 309
Table C.18 – Channel services related to communication 309

Table C.19 – Channel services related sub-topology management 309
Table C.20 – Channel services related to functions 310
Table C.21 – Channel services related to scan 310
Table C.22 – FA services related to general events 310
Table C.23 – FA services related to topology management 311
Table C.24 – FA services related to redundancy 311
Table C.25 – FA services related to storage of DTM data 311
Table C.26 – FA services related to DTM data synchronization 311
Table C.27 – FA related to presentation 312
Table C.28 – FA services related to audit trail 312
Table D.1 – Interoperability between components of different versions 314
Table E.1 – Adapter interface mappings 319

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 42: Object model integration profile – Common Language Infrastructure

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 62453-42, 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:

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

Enquiry draft	Report on voting
65E/439/DTR	65E/486/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 publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

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

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

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

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 62543, which is a technical report, is an interface specification for developers of FDT (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

With the integration of 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, 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, in general, is 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 TR 62453-42 is related to the IEC 62453 series.

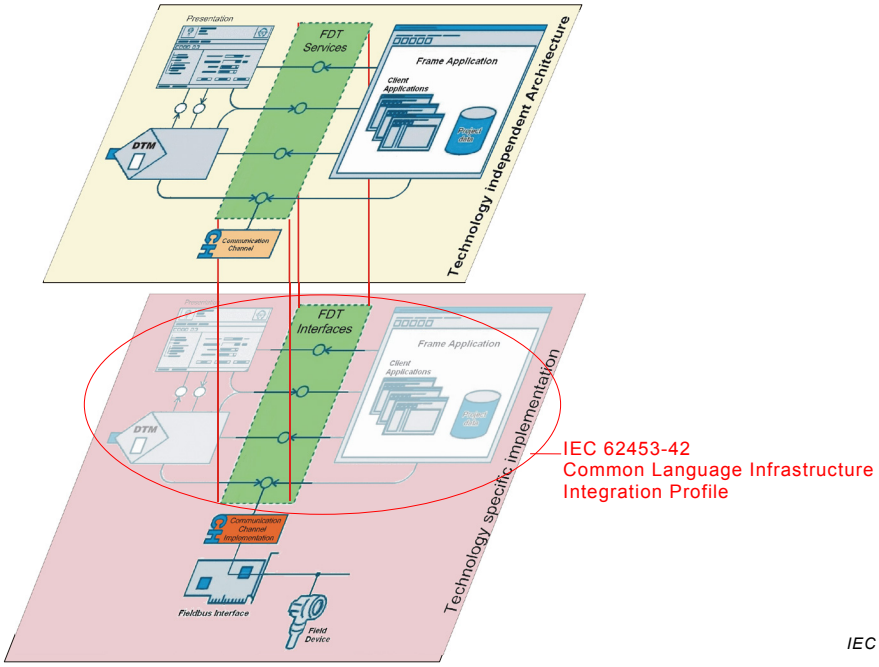

Figure 1 – Relation of IEC 62453-42 to the IEC 62453 series

The document structure is:

- Clause 3 explains the used terms, definitions and conventions
- Clause 4 introduces the general concepts of IEC 62453-42
- Clause 5 describes the technical concepts used to implement IEC 62453-42 and how FDT concepts are mapped to .NET Framework
- Clause 6 provides an overview of the FDT Objects, their interfaces and behavior
- Clause 7 presents an overview of the IEC 62453-42 datatypes
- Clause 8 shows the interaction of FDT Objects at runtime
- Clause 9 explains rules related to installation and deployment of DTMs

- Clause 10 explains how FDT life cycle concepts are implemented
- Clause 11 shows examples for Frame Application architectures

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 42: Object model integration profile – Common Language Infrastructure

1 Scope

This part of IEC 62453, which is a technical report, defines how the common FDT principles are implemented based on the .NET technology, including the object behavior and object interaction via .NET interfaces.

This document specifies FDT version 2.0.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

IEC 62453-2:—[1], *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*

3 Terms, definitions, abbreviations 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 as well as the following apply.

3.1.1
action
execution of a function which may involve several calls to interface methods of different FDT Objects

3.1.2
asynchronous methods
methods that trigger execution of asynchronous operations

Note 1 to entry: See also 5.6.7.

3.1.3
asynchronous operation
operation that is performed while the FDT object (client) that has requested the operation does not wait for the result, but the client is notified when the operation is finished

[1] To be published concurrently with this technical report.