IEC TS 62453-43

Edition 1.0 2024-11

TECHNICAL SPECIFICATION

colour inside

Field device tool (FDT) interface specification – Part 43: Object model integration profile – CLI and HTML

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.05; 35.110

ISBN 978-2-8327-0035-8

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

® Registered trademark of the International Electrotechnical Commission

CONTENTS

FOREWORD 19
INTRODUCTION 21
1 Scope 23
2 Normative references 23
3 Terms, definitions, abbreviated terms and conventions 23
3.1 Terms and definitions 23
3.2 Abbreviated terms 30
3.3 Conventions 31
4 Implementation concept 31
4.1 General 31
4.2 Implementation of abstract FDT Object model 31
4.3 FDT Frame Application (FA) 32
4.3.1 General 32
4.3.2 System communication 33
4.4 DTM Business Logic 34
4.4.1 General 34
4.4.2 DTM, DTM Device Type, and Device Ident Info 35
4.4.3 Device data info 36
4.4.4 Process data info 36
4.4.5 Diagnostic data info 37
4.4.6 Network management info 37
4.4.7 Function info 37
4.4.8 Report info 37
4.4.9 Document reference info 37
4.5 DTM functions 38
4.5.1 DTM User Interface 38
4.5.2 Function access control 38
4.5.3 User interaction in DTM WebUIs 38
4.5.4 Command functions 39
4.5.5 Static Function 39
4.6 Communication Channel 39
4.7 DTM categories 41
4.7.1 General 41
4.7.2 Device DTM 41
4.7.3 Communication DTM (CommDTM) 41
4.7.4 Gateway DTM 42
4.7.5 Composite Device DTM 43
4.7.6 Module DTM 44
4.7.7 Block Type Manager 44
4.8 User management 46
4.8.1 General 46
4.8.2 Multi-user access 46
4.8.3 User levels 46
4.9 FDT and system topology 49
4.9.1 General 49
4.9.2 Topology management 50
4.9.3 Address management 52

4.9.4 Configuration of fieldbus master or communication scheduler 53
4.9.5 Data exchange between Frame Applications 54
4.10 Modularity 54
4.11 FDT communication 55
4.11.1 General 55
4.11.2 Point-to-point communication 56
4.11.3 Nested communication 57
4.11.4 Dynamic changes in network 58
4.12 Identification 59
4.12.1 DTM instance identification 59
4.12.2 System GUI label 59
4.12.3 Hardware identification 59
4.13 Scanning and DTM assignment 60
4.13.1 Scanning introduction 60
4.13.2 Scanning 60
4.13.3 DTM assignment 61
4.13.4 Manufacturer-specific device identification 61
4.14 DTM data persistence and synchronization 62
4.14.1 Persistence overview 62
4.14.2 Relations of DTMDataSet 62
4.14.3 DTMDataSet structure 63
4.14.4 Types of persistent DTM data 65
4.14.5 Data synchronization 65
4.15 Device data and IO information 66
4.15.1 Exposing device data and IO information 66
4.15.2 Data access control 67
4.15.3 Routed IO information 69
4.15.4 Comparison of DTM and device data 70
4.15.5 PLC tool support 70
4.15.6 Support for multirole devices 72
4.16 Clone of DTM instances 73
4.16.1 General 73
4.16.2 Replicating a part of topology with Parent DTM and a subset of its Child DTMs 73
4.16.3 Cloning of a DTM without its children 73
4.16.4 Delayed cloning 73
4.17 Lifecycle concepts 74
4.18 Audit trail 74
4.18.1 General 74
4.18.2 Audit trail events 74
5 Technical concepts 75
5.1 General 75
5.2 Support of HTML versions 77
5.3 Support of JavaScript versions 77
5.4 Support of .NET Common Language Runtime versions 78
5.4.1 General 78
5.4.2 DTM rules 78
5.4.3 Frame Application rules 78
5.5 Support for 32-bit and 64-bit target platforms 79

5.6 Object activation and deactivation 79
5.6.1 General 79
5.6.2 Assembly loading and object creation 79
5.6.3 Assembly dependencies 79
5.6.4 Shared assemblies 80
5.6.5 Object deactivation and unloading 81
5.7 Datatypes 81
5.7.1 General 81
5.7.2 Serialization / deserialization 82
5.7.3 Optional elements 82
5.7.4 Verify 83
5.7.5 Clone 83
5.7.6 Equals 84
5.7.7 Lists 84
5.7.8 Nullable 84
5.7.9 Enumeration 84
5.7.10 Protocol-specific datatypes 84
5.7.11 Custom datatypes 87
5.8 General object interaction 88
5.8.1 General 88
5.8.2 Decoupling of FDT Objects 88
5.8.3 Parameter interchange with .NET datatypes 90
5.8.4 Interaction patterns 90
5.8.5 Properties 90
5.8.6 Synchronous methods 90
5.8.7 Asynchronous methods 90
5.8.8 Events pattern 97
5.8.9 Exception handling 97
5.9 Threading 101
5.9.1 General 101
5.9.2 Races 101
5.9.3 Locks 102
5.9.4 Deadlocks 102
5.10 Threading rules 102
5.10.1 Implementation rules 102
5.10.2 Avoiding deadlocks 103
5.10.3 FDT Object interaction rules 103
5.11 Localization support 104
5.11.1 General 104
5.11.2 Access to localized resources and culture-dependent functions 104
5.11.3 Handling of cultures 104
5.11.4 Switching the User Interface language 105
5.12 DTM User Interface implementation 105
5.12.1 General 105
5.12.2 Private dialogs 106
5.12.3 Modal DTM WebUI 106
5.13 DTM User Interface hosting 106
5.13.1 General 106
5.13.2 Hosting DTM WebUI 107

5.14 Static Function implementation ... 110
5.15 Persistence ... 112
5.15.1 Overview ... 112
5.15.2 Data format ... 113
5.15.3 Adding / reading / writing / deleting of data ... 113
5.15.4 Searching for data ... 115
5.16 Comparison of DTM and device data ... 116
5.16.1 Comparison of datasets using IDeviceData / IInstanceData ... 116
5.16.2 Comparison of datasets using IComparison ... 117
5.17 Tracing ... 117
5.18 Report generation ... 117
5.18.1 Introduction ... 117
5.18.2 Report types ... 118
5.18.3 DTM report data format ... 118
5.18.4 Report data exchange ... 119
5.19 Security ... 119
5.19.1 General ... 119
5.19.2 Strong naming of assemblies ... 119
5.19.3 Identification of origin ... 120
5.19.4 Code access security ... 120
5.19.5 Validation of FDT compliance certification ... 120
6 FDT Objects and interfaces ... 122
6.1 General ... 122
6.2 Frame Application ... 123
6.2.1 General ... 123
6.2.2 Frame Application Business Logic ... 123
6.2.3 Frame Application WebUI ... 126
6.3 DTM Business Logic ... 126
6.3.1 DTM BL interfaces ... 126
6.3.2 State machines related to DTM BL ... 131
6.3.3 State machine of instance data ... 138
6.4 DTM User Interface ... 142
6.4.1 DTM WebUI ... 142
6.5 Communication Channel ... 143
6.6 Availability of interface methods ... 145
7 FDT datatypes ... 146
7.1 General ... 146
7.2 Datatypes – Base ... 146
7.3 General datatypes ... 147
7.4 Datatypes – DtmInfo / TypeInfo ... 147
7.5 Datatypes – DeviceIdentInfo ... 149
7.6 Datatypes for installation and deployment ... 154
7.6.1 Datatypes – DtmPackageManifest ... 154
7.6.2 Datatypes – DtmManifest ... 155
7.6.3 Datatypes – DtmWebUiManifest ... 156
7.7 Datatypes – Communication ... 157
7.8 Datatypes – BusCategory ... 163
7.9 Datatypes – Device / instance data ... 163
7.9.1 General ... 163

7.9.2 Datatypes used in reading and writing DeviceData 170
7.10 Datatypes for export and import 172
7.10.1 Datatypes – TopologyImportExport 172
7.10.2 Datatypes – ImportExportDataset 174
7.11 Datatypes for process data description 174
7.11.1 Datatypes – ProcessDataInfo 174
7.11.2 Datatypes – Process Image 179
7.12 Datatypes – Address information 181
7.13 Datatypes – NetworkDataInfo 184
7.14 Datatypes – DTM functions 186
7.15 Datatypes – DTM messages 188
7.16 Datatypes – CommunicationChannelInfo 189
7.17 Datatypes – HardwareIdentification and scanning 191
7.17.1 General 191
7.17.2 Datatypes – DeviceScanInfo 191
7.17.3 Example – HardwareIdentification and scanning for HART® 192
7.18 Datatypes – DTM report types 193
7.19 Information related to device modules in a monolithic DTM 194
8 Workflows 196
8.1 General 196
8.2 Instantiation, loading and release 196
8.2.1 Finding a DTM BL object 196
8.2.2 Instantiation of a new DTM BL 198
8.2.3 Configuring access rights 200
8.2.4 Loading a DTM BL 201
8.2.5 Loading a DTM with Expert user level 202
8.2.6 Release of a DTM BL 204
8.3 Persistent storage of a DTM 204
8.3.1 Saving instance data of a DTM 204
8.3.2 Copy and versioning of a DTM instance 205
8.3.3 Dataset commit failed 206
8.3.4 Export a DTM dataset to file 206
8.4 Locking and DataTransactions in multi-user environments 207
8.4.1 General 207
8.4.2 Propagation of changes 208
8.4.3 Synchronizing DTMs in multi-user environments 209
8.5 Execution of DTM Functions 211
8.5.1 General 211
8.5.2 Finding a DTM WebUI 211
8.5.3 Instantiation of a DTM WebUI 212
8.5.4 Release of a DTM WebUI 213
8.5.5 Execution of command functions 214
8.5.6 Opening of documents 215
8.5.7 Interaction between DTM WebUI and DTM Business Logic 216
8.5.8 Interaction between DTM Business Logic and DTM WebUI 218
8.5.9 Interaction between DTM WebUI and DTM Business Logic with Cancel 218
8.5.10 Retrieving information about available Static Functions 219
8.5.11 Executing a Static Function 221
8.6 DTM communication 223

8.6.1 General 223
8.6.2 Establishing a communication connection 223
8.6.3 Cancel establishment of communication connection 224
8.6.4 Communicating with the device 225
8.6.5 Frame Application or Child DTM disconnect a device 226
8.6.6 Terminating a communication connection 227
8.6.7 DTM aborts communication connection 228
8.6.8 Communication Channel aborts communication connection 229
8.7 Nested communication 230
8.7.1 General 230
8.7.2 Communication request for a nested connection 231
8.7.3 Propagation of errors for a nested connection 232
8.8 Topology planning 233
8.8.1 General 233
8.8.2 Adding a DTM to the topology 233
8.8.3 Removing a DTM from topology 234
8.8.4 Frame Application creates topology 235
8.8.5 DTM generates sub-topology 236
8.8.6 Physical Layer and DataLinkLayer 238
8.9 Instantiation, configuration, move and release of Child DTMs 238
8.9.1 General 238
8.9.2 Instantiation and configuration of Child DTM BL 239
8.9.3 Interaction between Parent DTM and Child DTM 240
8.9.4 Interaction between Parent DTM and Child DTM using IDtmMessaging 242
8.9.5 Parent DTM moves a Child DTM 242
8.9.6 Parent DTM removes Child DTM 243
8.10 Topology scan 244
8.10.1 General 244
8.10.2 Scan of network topology 244
8.10.3 Cancel topology scan 245
8.10.4 Scan based DTM assignment 246
8.10.5 Manufacturer-specific device identification 247
8.11 Configuration of communication networks 249
8.11.1 Configuration of a fieldbus master 249
8.11.2 Integration of a passive device 250
8.12 Using IO information 250
8.12.1 Assignment of symbolic name to process data 250
8.12.2 Creation of Process Image 252
8.12.3 Validation of changes in process image while PLC is running 253
8.12.4 Changing of variable names using process image interface 254
8.13 Managing addresses 255
8.13.1 Set DTM address with user interface 255
8.13.2 Set DTM addresses without user interface 257
8.13.3 Display or modify addresses of all Child DTMs with user interface 258
8.14 Device-initiated data transfer 259
8.15 Reading and writing data 260
8.15.1 Read/write instance data 260
8.15.2 Read/write device data 261
8.16 Comparing data 263

8.16.1 Comparing device dataset and instance dataset 263
8.16.2 Comparing different instance datasets 263
8.17 Reassigning a different DtmDeviceType at a device node 264
8.17.1 General 264
8.17.2 DTM detects a change in connected device type 265
8.17.3 Search matching DtmDeviceTypes after incompatible device exchange 267
8.17.4 Reassign DtmDeviceType after incompatible device exchange 268
8.18 Copying part of FDT Topology 270
8.18.1 Cloning of a single DTM without Children 270
8.18.2 Cloning of a DTM with all its Children 271
8.19 Sequences for audit trail 271
8.19.1 General 271
8.19.2 Audit trail of parameter modifications in instance dataset 271
8.19.3 Audit trail of parameter modifications in device dataset 272
8.19.4 Audit trail of function calls 273
8.19.5 Audit trail of general notification 274
9 Installation 274
9.1 General 274
9.2 Common rules 274
9.2.1 Predefined installation paths 274
9.2.2 Predefined web-server paths 276
9.2.3 Manifest files 277
9.2.4 Paths in manifest files 277
9.2.5 DTM Installation package format 277
9.2.6 Digital signatures of package components 278
9.3 Installation of FDT core assemblies 278
9.4 Installation of communication protocols 278
9.4.1 General 278
9.4.2 Registration 279
9.4.3 Protocol manifest 279
9.4.4 NuGet package 280
9.5 Installation of DTMs 281
9.5.1 General 281
9.5.2 Registration 282
9.5.3 DTM manifest 283
9.5.4 DTM WebUI manifest 284
9.5.5 WebUI container files 284
9.6 DTM package file 285
9.6.1 Naming convention 285
9.6.2 Structure (physical model) 286
9.6.3 Relationships (logical model) 287
9.6.4 Core properties 288
9.6.5 License file 289
9.6.6 Readme file 289
9.6.7 Icon file 289
9.6.8 DTM package manifest 290
9.6.9 DTM device identification manifest 292
9.6.10 DTM package file creation rules 293
9.6.11 Countersignatures 293

9.6.12 Referenced NuGet packages 293
9.7 DTM deployment 294
9.8 Paths and file information 295
9.8.1 Path information provided by a DTM via IFunction 295
9.8.2 Paths and persistence 296
9.8.3 Multi-user systems 296
10 Life cycle concept 296
10.1 General 296
10.2 Technical concept 296
10.2.1 General 296
10.2.2 DtmManifest / DtmInfo 297
10.2.3 TypeInfo 298
10.2.4 Supported DataSet formats 299
10.2.5 DeviceIdentInfo 299
10.2.6 Dataset 300
10.2.7 DeviceScanInfo 300
10.3 DTM installation 301
10.3.1 General 301
10.3.2 Handling of DTM installations 302
10.4 Life cycle scenarios 302
10.4.1 Overview 302
10.4.2 Search for device type in DTM packages 304
10.4.3 Search for installed DTMs 304
10.4.4 Dataset migration for reassigned DTM 306
11 Frame Application architectures 307
11.1 General 307
11.2 Standalone application 307
11.3 Remote User Interface 307
11.4 Distributed multi-user application 308
11.5 OPC UA 309
11.6 Web services 310
Annex A (normative) Use case model 312
A.1 Use case model overview 312
A.2 Actors 312
A.3 Use cases 313
A.3.1 Use case overview 313
A.3.2 Observation use cases 314
A.3.3 Operation use cases 315
A.3.4 Maintenance use cases 317
A.3.5 Planning use cases 321
A.3.6 Main operation 324
A.3.7 OEM service 324
A.3.8 Administration 325
Annex B (normative) FDT interface definition and datatypes 326
Annex C (normative) Mapping of services to interface methods 327
C.1 General 327
C.2 DTM services 327
C.3 Presentation object services 331

C.4 General channel services....331
C.5 Process Channel services....332
C.6 Communication Channel Services....332
C.7 Frame Application Services....333
Annex D (normative) FDT version interoperability guide....337
D.1 Overview....337
D.2 General....337
D.3 Component interoperability....338
Annex E (normative) Definition of JavaScript APIs....339
E.1 General....339
E.2 Request / Response datatypes....339
E.3 Message datatypes....340
E.3.1 SendMessages....340
E.3.2 Request datatype....340
E.3.3 Response datatype....341
E.4 API for WebUIs....341
E.5 API for tracing....344
Annex F (informative) Implementation hints....347
F.1 IAsyncResult pattern....347
F.1.1 Overview....347
F.1.2 BeginOperationName....347
F.1.3 EndOperationName....347
F.1.4 Asynchronous Completed callback....348
F.2 Threading best practices....348
F.3 Testing DTM messages in regard to JSON serialization/deserialization....349
Annex G (informative) Comparison of IEC TR 62453-42 and IEC TS 62453-43....351
Annex H (informative) Physical layer examples....352
H.1 General....352
H.2 Interbus S....352
H.3 PROFIBUS....352
H.4 PROFINET....353
Annex I (informative) Predefined SemanticIds....354
I.1 General....354
I.2 Data....354
I.3 Images....354
I.4 Documents....355
Annex J (informative) Standard StaticFunctions....356
J.1 General....356
J.2 StaticFunction GetDeviceStatus....356
J.3 StaticFunction GetProcessValue....357
Bibliography....360

Figure 1 – Relation of IEC TS 62453-43 to the IEC 62453 series....21
Figure 2 – IEC TS 62453-43 Object model....32
Figure 3 – Frame Application....32
Figure 4 – Frame Application with integrated Communication Channel....34
Figure 5 – DTM Business Logic....34

Figure 6 – DTM, Device Type and Device Ident Info 35
Figure 7 – Process data info 36
Figure 8 – Communication Channel 40
Figure 9 – Device DTM 41
Figure 10 – Communication DTM 42
Figure 11 – Gateway DTM 42
Figure 12 – Composite Device DTM 43
Figure 13 – Module DTM 44
Figure 14 – Block Type Manager 45
Figure 15 – Logical topology and physical topology 50
Figure 16 – FDT and logical topology 50
Figure 17 – DTMs and physical topology 51
Figure 18 – Address setting via DTM WebUI 53
Figure 19 – Fieldbus master configuration tool as part of a DTM 54
Figure 20 – Point-to-point communication 57
Figure 21 – Nested communication 58
Figure 22 – Identification of connected devices 60
Figure 23 – Fieldbus scanning 61
Figure 24 – FDT storage and synchronization mechanism 62
Figure 25 – Relation between DTMDataSet, DTM instance, and device 63
Figure 26 – DTMDataSet structure 64
Figure 27 – Data Synchronization 65
Figure 28 – Example for same organization of FunctionInfo and DataInfo 67
Figure 29 – Routed IO information 69
Figure 30 – Process Image 70
Figure 31 – Transfer of layout information using IProcessImage 71
Figure 32 – Multirole device 72
Figure 33 – FDT .NET Assemblies 76
Figure 34 – FDT Object implementation 77
Figure 35 – Example: Assembly dependencies 80
Figure 36 – Example: Datatype definition 82
Figure 37 – Example: Data cloning 83
Figure 38 – Example: Methods without data cloning 84
Figure 39 – Protocol-specific datatypes 85
Figure 40 – Protocol manifest and type info attributes 86
Figure 41 – Example: Protocol assembly attributes 87
Figure 42 – Example: Handling of protocol-specific assemblies in Frame Application 87
Figure 43 – Example: Handling of KnownType for custom data types 88
Figure 44 – Decoupled FDT Objects in IEC TS 62453-43 89
Figure 45 – IAsyncResult pattern: blocking call 91
Figure 46 – Example: Blocking use of asynchronous interface 92
Figure 47 – IAsyncResult pattern (simplified): blocking call 92
Figure 48 – IAsyncResult pattern: non-blocking call 93

Figure 49 – Example: Non-blocking use of asynchronous interface 94
Figure 50 – IAsyncResult pattern (simplified depiction): non-blocking call 94
Figure 51 – IAsyncResult pattern: canceling an operation 95
Figure 52 – IAsyncResult pattern: providing progress events 96
Figure 53 – General concept for hosting DTM WebUI 107
Figure 54 – Example: Integration of a DTM WebUI into a Frame WebUI 108
Figure 55 – Example: Hosting a DTM WebUI in a WPF Frame Application 109
Figure 56 – Example: Integrating the WebData connector in a DTM WebUI 110
Figure 57 – Relation of StaticFunctionDescription to Static Function 111
Figure 58 – DTMDataset structure 112
Figure 60 – Example: Writing of DTM data in DTMDataSubset 114
Figure 61 – Example: Reading of DTM data from a DTMDataSubset 115
Figure 62 – Example: Creation of a BulkData.DTMDataSubset with descriptor 116
Figure 63 – Example: Searching for DTMDataSubsets with specific descriptor 116
Figure 64 – Skeleton of a DTM-specific report fragment 119
Figure 65 – Example: Conformity record file 121
Figure 66 – Example: checking conformity record file 122
Figure 67 – Frame Application BL interfaces 124
Figure 68 – Frame Application WebUI 126
Figure 69 – DTM Business Logic interfaces (Part 1) 127
Figure 70 – DTM Business Logic interfaces (Part 2) 128
Figure 71 – State machine of DTM BL 133
Figure 72 – Online state machine of DTM 135
Figure 73 – Modifications of data through a DTM 139
Figure 74 – ModifiedInDtm: State machine of instance data 140
Figure 75 – ModifiedInDevice: State machine related to device data 141
Figure 76 – DTM WebUI interfaces 142
Figure 77 – Example: Definition of DtmWebUi object 142
Figure 78 – Example: Access to DtmWebUi by Frame WebUI 142
Figure 79 – Communication Channel interfaces 143
Figure 80 – Connection state machine 144
Figure 81 – FdtDatatype and FdtList 146
Figure 82 – DtmInfo / TypeInfo – datatypes 148
Figure 83 – DeviceIdentInfo – datatypes 150
Figure 84 – DeviceIdentInfo – Example for HART® 151
Figure 85 – Example: DeviceIdentInfo creation 153
Figure 86 – Example: Using DeviceIdentInfo 154
Figure 87 – Example: DeviceIdentInfoTypeAttribute 154
Figure 88 – DtmPackageManifest – datatypes 155
Figure 89 – DtmManifest – datatypes 156
Figure 90 – DtmWebUiManifest – datatypes 157
Figure 91 – Communication datatypes – Connect 158
Figure 92 – Communication datatypes – Transaction 158

Figure 93 – Communication datatypes – Disconnect ... 159
Figure 94 – Communication datatypes – Subscribe ... 159
Figure 95 – Communication datatypes – Scanning ... 160
Figure 96 – Communication datatypes – Address setting ... 160
Figure 97 – Example: Communication – Connect for HART® ... 162
Figure 98 – Example: Communication – CommunicationType for HART ... 162
Figure 99 – BusCategory – datatypes ... 163
Figure 100 – Device / instance data – datatypes ... 164
Figure 101 – Example: Providing information on data of a HART® device ... 166
Figure 102 – Example: Providing information on module data of a PROFIBUS device ... 167
Figure 103 – Example: Providing information on data ... 168
Figure 104 – Example: Providing information on structured data ... 169
Figure 105 – EnumInfo – datatype ... 170
Figure 106 – Read and Write Request – datatypes ... 170
Figure 107 – ResponseInfo – datatype ... 171
Figure 108 – TopologyImportExport – datatypes ... 173
Figure 109 – ImportExportDataset – datatypes ... 174
Figure 110 – ProcessDataInfo – datatypes ... 175
Figure 111 – IOSignalInfo – datatypes ... 176
Figure 112 – Example: ProcessDataInfo for HART (UML) ... 177
Figure 113 – Example: ProcessDataInfo creation for HART® ... 178
Figure 114 – Example: Using ProcessData for HART® ... 179
Figure 115 – Example: IOSignalInfoType attribute ... 179
Figure 116 – ProcessImage – datatypes ... 180
Figure 117 – AddressInfo – datatypes ... 181
Figure 118 – Example: AddressInfo creation ... 182
Figure 119 – Example: Using AddressInfo ... 183
Figure 120 – Example: DeviceAddressTypeAttribute ... 183
Figure 121 – NetworkDataInfo – datatypes ... 184
Figure 122 – Example: NetworkDataInfo creation example ... 185
Figure 123 – Example: NetworkDataInfo using example ... 186
Figure 124 – Example: NetworkDataTypeAttribute example ... 186
Figure 125 – DTM Function – datatypes ... 187
Figure 126 – DTM Messages – datatypes ... 188
Figure 127 – CommunicationChannelInfo – datatypes ... 190
Figure 128 – Example: Channel information ... 191
Figure 129 – DeviceScanInfo – datatypes ... 192
Figure 130 – Example: HARTDeviceScanInfo – datatype ... 193
Figure 131 – DTM Report – datatypes ... 194
Figure 132 – Information related to device modules ... 195
Figure 133 – Finding a DTM BL object ... 197
Figure 134 – Instantiation of a new DTM BL ... 199

Figure 135 – Configuration of user permissions ... 201
Figure 136 – Loading a DTM BL ... 202
Figure 137 – Loading a DTM with Expert user level ... 203
Figure 138 – Release of a DTM BL ... 204
Figure 139 – Saving data of a DTM ... 205
Figure 140 – Dataset commit failed ... 206
Figure 141 – Export a DTM dataset to file ... 207
Figure 142 – Propagation of changes ... 209
Figure 143 – Synchronizing DTMs in multi-user environments ... 210
Figure 144 – Finding a DTM WebUI ... 212
Figure 145 – Instantiation of a DTM WebUI ... 213
Figure 146 – Release of a DTM WebUI ... 214
Figure 147 – Execute a command function ... 215
Figure 148 – Opening a document ... 216
Figure 149 – Interaction triggered by the DTM WebUI ... 217
Figure 150 – Interaction triggered between DTM BL and DTM WebUI ... 218
Figure 151 – Interaction triggered and canceled by the DTM WebUI ... 219
Figure 152 – Retrieving information about available Static Functions ... 220
Figure 153 – Example: Information about available Static Functions ... 221
Figure 154 – Executing a Static Function ... 222
Figure 155 – Establishing a communication connection ... 224
Figure 156 – DTM cancels ongoing Connect operation ... 225
Figure 157 – Communicating with the device ... 226
Figure 158 – Child DTM disconnects ... 227
Figure 159 – Child DTM terminates a connection ... 228
Figure 160 – Child DTM aborts a connection ... 229
Figure 161 – Communication Channel aborts a connection ... 230
Figure 162 – Example: Nested communication behavior ... 231
Figure 163 – Example: Nested communication data exchange ... 232
Figure 164 – Add DTM to topology ... 234
Figure 165 – Removing a DTM from topology ... 235
Figure 166 – Frame Application creates topology ... 236
Figure 167 – DTM generates sub-topology ... 237
Figure 168 – Instantiation and configuration of Child DTM BL ... 239
Figure 169 – Interaction between Parent DTM and Child DTM ... 241
Figure 170 – Interaction using IDtmMessaging ... 242
Figure 171 – Parent DTM moves a Child DTM ... 243
Figure 172 – Parent DTM removes Child DTM ... 244
Figure 173 – Scan of network topology ... 245
Figure 174 – Cancel topology scan ... 246
Figure 175 – Scan based DTM assignment ... 247
Figure 176 – Manufacturer-specific device identification ... 248
Figure 177 – Configuration of a fieldbus master ... 249

Figure 178 – Integration of a passive device 250
Figure 179 – Assignment of process data 251
Figure 180 – Creation of process image 253
Figure 181 – Validation of changes while PLC is running 254
Figure 182 – Changing of variable names using process image interface 255
Figure 183 – Set DTM address with WebUI 256
Figure 184 – Set DTM addresses without UI 257
Figure 185 – Display or modify child addresses with UI 258
Figure 186 – Device-initiated data transfer 259
Figure 187 – Read/write instance data 260
Figure 188 – Read/write device data 262
Figure 189 – Comparing device dataset and instance dataset 263
Figure 190 – Compare instance data with persisted dataset 264
Figure 191 – DTM triggers ActiveTypeChanged event 266
Figure 192 – Find matching DtmDeviceTypes after incompatible device exchange 268
Figure 193 – Reassign a DtmDeviceType after incompatible device exchange 269
Figure 194 – Clone DTM without children 270
Figure 195 – Clone DTM with all children 271
Figure 196 – Audit trail of parameter modifications in instance dataset 272
Figure 197 – Audit trail of parameter modifications in device 273
Figure 198 – Audit trail of function calls 273
Figure 199 – Installation paths (with example DTM) 276
Figure 200 – Installed FDT and protocol assemblies 279
Figure 201 – Example: Protocol manifest 280
Figure 202 – NuGet file structure 281
Figure 203 – Search for installed DTMs 282
Figure 204 – Example: DtmManifest 283
Figure 205 – Example: DtmWebUiManifest 284
Figure 206 – Structure of a WebUI container file 285
Figure 207 – DTM package file structure 287
Figure 208 – DTM package file relationships 288
Figure 209 – Example: DtmPackageManifest 291
Figure 210 – Example: DeviceIdentManifest 292
Figure 211 – DTM deployment 294
Figure 212 – Unpacking the DTM package 295
Figure 213 – Overview DTM identification 297
Figure 214 – Identification attributes in DTM package 301
Figure 215 – Search DTM package for list of supported types 304
Figure 216 – Scan installed DTMs 305
Figure 217 – Dataset migration to a reassigned DtmDeviceType 306
Figure 218 – Client / Server Application 308
Figure 219 – Example for distributed multi-user application 309
Figure 220 – OPC UA server based on IEC TS 62453-43 310

Figure 221 – Web server based on IEC TS 62453-43 311
Figure A.1 – Main use case diagram 312
Figure A.2 – Observation use cases 314
Figure A.3 – Operation use cases 316
Figure A.4 – Maintenance use cases 318
Figure A.5 – Planning use cases 322
Figure F.1 – Example: Test for JSON data 349
Figure F.2 – Example: Erroneous DTM-specific message example 349
Figure F.3 – Example: Failing test for erroneous JSON data 350

Table 1 – FDT user levels 47
Table 2 – Role-dependent access rights and user interfaces for DTMs 47
Table 3 – Description of properties related to data access control 68
Table 4 – Supported HTML versions 77
Table 5 – Supported ECMAScript versions 77
Table 6 – Supported .NET Standard library versions 78
Table 7 – Frame Application BL interfaces 125
Table 8 – Frame Application WebUI interfaces 126
Table 9 – DTM Business Logic interfaces 129
Table 10 – Availability of interfaces depending of type of DTM 130
Table 11 – Definition of DTM BL state machine 134
Table 12 – Definition of online state machine 136
Table 13 – Description of instance dataset states 140
Table 14 – Description of dataset states regarding online modifications 141
Table 15 – DTM WebUI interfaces 142
Table 16 – Communication Channel interfaces 143
Table 17 – Definition of connection state machine 144
Table 18 – Availability of DTM BL methods in different states 145
Table 19 – FDT base datatypes 147
Table 20 – FDT General datatypes 147
Table 21 – DtmInfo datatype description 149
Table 22 – DeviceIdentInfo datatype description 150
Table 23 – DeviceIdentInfo – Example for HART® 152
Table 24 – DtmPackageManifest datatype description 155
Table 25 – DtmManifest datatype description 156
Table 26 – DtmWebUiManifest datatype description 157
Table 27 – Communication datatype description 161
Table 28 – BusCategory datatype description 163
Table 29 – DeviceData datatype description 165
Table 30 – Reading and Writing datatype description 170
Table 31 – Reading and Writing datatype description 172
Table 32 – TopologyImportExport datatype description 173
Table 33 – ImportExportDataset datatype description 174

Table 34 – ProcessDataInfo datatype description 175
Table 35 – IOSignalInfo datatype description 176
Table 36 – ProcessImage datatype description 180
Table 37 – AddressInfo datatype description 181
Table 38 – NetworkDataInfo datatype description 185
Table 39 – DTM Function datatype description 188
Table 40 – DTM Messages datatype description 189
Table 41 – CommunicationChannelInfo datatype description 190
Table 42 – DeviceScanInfo datatype description 192
Table 43 – Example: HARTDeviceScanInfo datatype description 193
Table 44 – Reporting datatype description 194
Table 45 – Predefined FDT installation paths 274
Table 46 – Predefined web-server paths 277
Table 47 – Overview of relationships 288
Table 48 – Overview of core properties 289
Table 49 – DTM identification 298
Table 50 – TypeInfo – User readable description of supported types 298
Table 51 – TypeInfo identification 299
Table 52 – DtmType – Dataset support identification 299
Table 53 – Dataset identification 300
Table 54 – DeviceScanInfo – scanned device identification 301
Table 55 – Package information 302
Table 56 – Changing DTM – overview 303
Table A.1 – Actors 313
Table A.2 – Observation use cases 314
Table A.3 – Operation use cases 316
Table A.4 – Maintenance use cases 318
Table A.5 – Planning use cases 322
Table C.1 – General services 327
Table C.2 – DTM service related to installation 327
Table C.3 – DTM service related to DTM information 328
Table C.4 – DTM services related to DTM state machine 328
Table C.5 – DTM services related to function 328
Table C.6 – DTM services related to documentation 329
Table C.7 – DTM services to access the instance data 329
Table C.8 – DTM services to access diagnosis 329
Table C.9 – DTM services to access to device data 330
Table C.10 – DTM services related to network management information 330
Table C.11 – DTM services related to online operation 330
Table C.12 – DTM services related to FDT-Channel objects 330
Table C.13 – DTM services related to import and export 331
Table C.14 – DTM services related to data synchronization 331
Table C.15 – DTM UI state control 331

Table C.16 – General channel service 331
Table C.17 – Channel services for IO related information 332
Table C.18 – Channel services related to communication 332
Table C.19 – Channel services related sub-topology management 333
Table C.20 – Channel services related to functions 333
Table C.21 – Channel services related to scan 333
Table C.22 – FA services related to general event 333
Table C.23 – FA services related to topology management 334
Table C.24 – FA services related to redundancy 334
Table C.25 – FA services related to storage of DTM data 334
Table C.26 – FA services related to DTM data synchronization 334
Table C.27 – FA services related to presentation 335
Table C.28 – FA services related to audit trail 335
Table C.29 – FA services related to sandboxing a DTM 335
Table C.30 – FA services related to sandboxing a Communication Channel 336
Table D.1 – Interoperability between components of different versions 338
Table J.1 – StaticFunctionDescription for GetDeviceStatus 356
Table J.2 – IStaticFunction2:BeginExecute arguments for GetDeviceStatus 357
Table J.3 – Exceptions for GetDeviceStatus 357
Table J.4 – StaticFunctionDescription for GetProcessValue 358
Table J.5 – IStaticFunction2:BeginExecute arguments for GetDeviceStatus 358
Table J.6 – Exceptions for GetProcessValue 359

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 43: Object model integration profile – CLI and HTML

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) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had not received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

IEC TS 62453-43 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation. It is a Technical Specification.

The text of this Technical Specification is based on the following documents:

Draft	Report on voting
65E/1115/DTS	65E/1146/RVDTS

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

The language used for the development of this Technical Specification is English.

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

A list of all parts in the IEC 62543 series, published under the general title *Field Device Tool (FDT) interface specification*, can be found on the IEC website.

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

- reconfirmed,
- withdrawn, or
- revised.

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

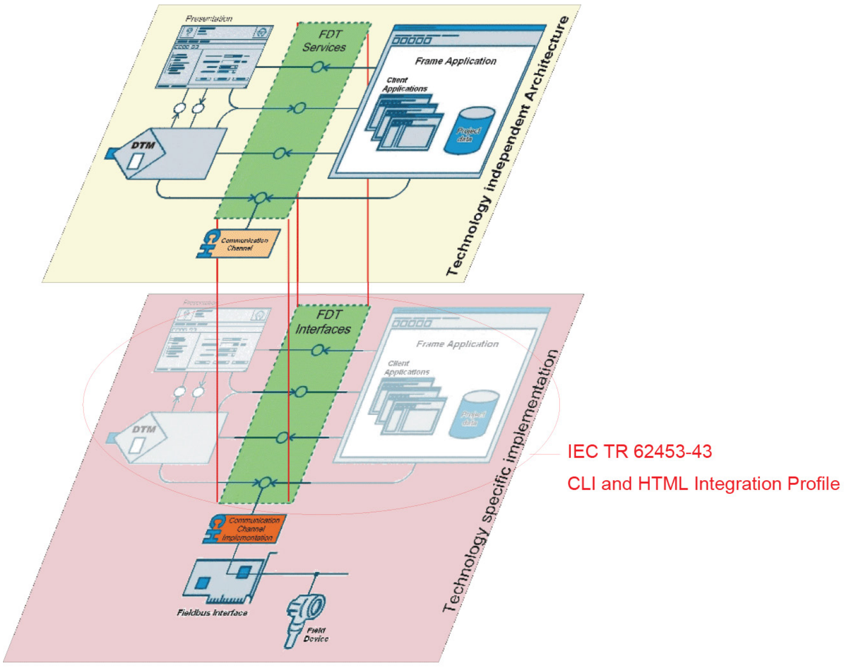
INTRODUCTION

This document 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-specific and device-specific tools, there is a need to integrate these tools into higher-level system-wide planning tools 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 fieldbuses and thus meets the requirements for integrating different kinds of devices into heterogeneous control systems.

Figure 1 shows how IEC TS 62453-43 is related to the IEC 62453 series.



IEC

Figure 1 – Relation of IEC TS 62453-43 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 TS 62453-43
- Clause 5 describes the technical concepts used to implement IEC TS 62453-43 and how FDT concepts are mapped to .NET Standard
- Clause 6 provides an overview of the FDT Objects, their interfaces and behaviour
- Clause 7 presents an overview of the IEC TS 62453-43 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 43: Object model integration profile – CLI and HTML

1 Scope

This part of IEC 62453, which is a technical specification, specifies how the common FDT principles are implemented based on the CLI technology and web technologies for graphical user interfaces. The specification includes the object behaviour and object interaction via .NET Standard interfaces and JavaScript APIs. Emphasis has been placed on support of distributed Frame Application architectures.

This document specifies FDT version 3.0.

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 62453-1:2024, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-2:2022, *Field Device Tool (FDT) interface specification – Part 2: Concepts and detailed description*