

DIN EN 61804-3

ICS 35.060; 35.240.50

Einsprüche bis 2014-04-07
Vorgesehen als Ersatz für
DIN EN 61804-3:2011-09**Entwurf****Funktionsbausteine für die Prozessautomation und elektronische Gerätebeschreibungssprache –****Teil 3: Elektronische Gerätebeschreibungssprache (EDDL)****(IEC 65E/320/CDV:2013);****Englische Fassung FprEN 61804-3:2013**

Function blocks (FB) for process control and EDDL –
Part 3: Electronic Device Description Language (EDDL)
(IEC 65E/320/CDV:2013);
English version FprEN 61804-3:2013

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2014-02-07 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 301 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die Englische Fassung des europäischen Dokuments FprEN 61804-3:2013 „Funktionsbausteine für die Prozessautomation und elektronische Gerätebeschreibungssprache – Teil 3: Elektronische Gerätebeschreibungssprache (EDDL)“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 65E/320/CDV:2013 „Function blocks (FB) for process control and EDDL – Part 3: Electronic Device Description Language (EDDL)“ unverändert in den Entwurf FprEN 61804-3:2013 übernommen worden.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnichtsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für den vorliegenden Normentwurf erfüllt.

Da sich die Benutzer des vorliegenden Normentwurfs der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61804-3 veröffentlicht. Zu deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Normentwurfs hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Dieser Normentwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Änderungen

Gegenüber DIN EN 61804-3:2011-09 wurden folgende Änderungen vorgenommen:

- a) Plug-In für das Menü „Schlüsselwort“
- b) Komponentenkonstrukt für Anforderungen an den Kommunikationsserver (zusätzliche Eigenschaften)
- c) Erweiterung des Klassenattributs
- d) neue Eigenschaft „private“ einigen Kommunikationsobjekten hinzugefügt
- e) Die folgenden Änderungen werden in die EDDL, basierend auf der EDDL Harmonisierung, integriert:
 - einige ungenutzte Eigenschaften wurden entfernt
 - einige Profileigenschaften wurden harmonisiert

Nationaler Anhang NA (informativ)

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 60050-351 (teilweise), IEC 61804-2 und die folgenden Begriffe.

3.1.1

Builtin

(en: **Builtin**)

vordefiniertes Unterprogramm, das von der EDD-Anwendung z. B. für Kommunikations- und Anzeigewecke ausgeführt wird

3.1.2

Gerät

(en: **device**)

unabhängige physische Einheit, die fähig ist, eine oder mehrere spezifische Funktionen in einem speziellen Kontext auszuführen und die durch ihre Schnittstellen abgegrenzt ist

[QUELLE: IEC 61499-1]

3.1.3

EDD-Anwendung

(en: **EDD application**)

Programm, das die EDD oder eine beliebige übersetzte Form der EDD verwendet und Funktionen wie die Kommunikationsdarstellung, die Datendarstellung, die graphische Darstellung usw. bereitstellt

3.1.4

EDDL-Prozessor

(en: **EDDL processor**)

Prozessor oder Programm, der/das die EDD in eine ausführbare Form übersetzt, die von einer EDD-Anwendung verarbeitet werden kann

3.1.5

EDDL-Profil

(en: **EDDL profile**)

Auswahl der unterstützten Elemente der lexikalischen EDDL-Struktur, einschließlich der Syntaxdefinitionen für eine Anzahl von spezifischen Konsortien

3.1.6**elektronische Gerätebeschreibungssprache****EDDL****(en: Electronic Device Description Language)**

formale Sprache zur Beschreibung von Geräten und Kommunikationskomponenten

3.1.7**elektronische Gerätebeschreibung****EDD****(en: Electronic Device Description)**

Datensammlung, welche die Geräteparameter, deren Abhängigkeiten und graphische Darstellung sowie eine Beschreibung der übermittelten Datensätze enthält

Anmerkung 1 zum Begriff: Die elektronische Gerätebeschreibung wird unter Verwendung der EDDL erstellt.

3.1.8**EDD-Compiler****(en: Electronic Device Description Compiler)**

Compiler, der die EDD-Quelle in ein internes Format übersetzt, das vom EDD-Interpreter verwendet wird

3.1.9**EDD-Interpreter****EDDI****(en: Electronic Device Description Interpreter)**

Interpreter, der die EDD-Quelle oder ein vom EDDL-Compiler vorgegebenes internes Format verwendet, um dem EDD-Anwender die EDD-Informationen bereitzustellen

3.1.10**Funktionsbaustein (Funktionsbausteininstanz)****(en: function block (function block instance))**

funktionale Software-Einheit, die eine individuelle, mit einem Namen versehene Kopie einer Datenstruktur und zugeordnete Operationen umfasst, die durch einen entsprechenden FB-Typ festgelegt sind

[QUELLE: IEC 61804-2]

Anmerkung 1 zum Begriff: Typische Operationen eines FB schließen eine Veränderung der Datenwerte in seiner Datenstruktur ein.

3.1.11**Hardware****(en: hardware)**

physische Einrichtung im Gegensatz zu Programmen, Prozeduren, Regeln und der zugehörigen Dokumentation

[QUELLE: ISO/AFNOR Dictionary of Computer Science]

3.1.12**Vorprozessor****(en: preprocessor)**

Teil der Laufzeitumgebung, der die Informationen in einer EDD umwandelt

3.1.13**Anweisungen für den Vorprozessor****(en: preprocessor directives)**

Beschreibung der Bedingungen für die Filterung des EDD-Codes vor der Kompilierung oder Interpretation

Anmerkung 1 zum Begriff: Beispielsweise eine Anweisung für den Vorprozessor zur Festlegung von Namen für Konstanten oder zum Schreiben von Makros für ein vereinfachtes Lesen von Code.

3.1.14

Software

(en: **software**)

geistige Schöpfung, die die Programme, Prozeduren, Regeln und alle zugehörigen Dokumentationen umfasst, die zum Betrieb eines Systems gehören

[QUELLE: IEC 61499-1]

1 CONTENTS

2	FOREWORD.....	13
3	INTRODUCTION.....	15
4	1 Scope.....	16
5	2 Normative references	16
6	3 Terms, definitions, abbreviated terms and acronyms	17
7	3.1 Terms and definitions	17
8	3.2 Abbreviated terms and acronyms	18
9	4 Conformance statement.....	19
10	5 Conventions for lexical structures	19
11	6 EDD and EDDL model	20
12	6.1 Overview of EDD and EDDL	20
13	6.2 EDD architecture	21
14	6.3 Concepts of EDD.....	21
15	6.4 Principles of the EDD development process	21
16	6.5 Interrelations between the lexical structure and formal definitions	22
17	6.6 Builtins	22
18	6.7 Profiles.....	23
19	7 Electronic Device Description Language.....	23
20	7.1 Overview	23
21	7.2 EDD identification information	34
22	7.3 AXIS.....	36
23	7.4 BLOCK.....	38
24	7.5 CHART.....	49
25	7.6 COLLECTION.....	51
26	7.7 COMMAND.....	52
27	7.8 COMPONENT	58
28	7.9 COMPONENT_FOLDER.....	63
29	7.10 COMPONENT_REFERENCE.....	63
30	7.11 COMPONENT_RELATION	64
31	7.12 Void (CONNECTION)	67
32	7.13 Void (DOMAIN)	67
33	7.14 EDIT_DISPLAY	67
34	7.15 FILE	69
35	7.16 GRAPH	70
36	7.17 GRID.....	71
37	7.18 IMAGE	73
38	7.19 IMPORT	74
39	7.20 INTERFACE	89
40	7.21 LIKE.....	90
41	7.22 LIST	91
42	7.23 MENU	92
43	7.24 METHOD.....	99
44	7.25 Void (PROGRAM).....	102
45	7.26 RECORD.....	102
46	7.27 REFERENCE_ARRAY.....	102
47	7.28 Relations.....	103

48	7.29 RESPONSE_CODES.....	105
49	7.30 SOURCE.....	105
50	7.31 TEMPLATE	107
51	7.32 VALUE_ARRAY.....	108
52	7.33 VARIABLE.....	109
53	7.34 VARIABLE_LIST	126
54	7.35 WAVEFORM	126
55	7.36 Common attributes	132
56	7.37 Conditional expression	141
57	7.38 Referencing.....	142
58	7.39 Strings	151
59	7.40 Expression	154
60	7.41 Text dictionary.....	159
61	7.42 PLUGIN.....	159
62	7.43 BLOB	160
63	Annex A (normative) EDDL formal definition	161
64	A.1 EDDL preprocessor	161
65	A.2 Conventions	164
66	A.3 Operators	166
67	A.4 Keywords	169
68	A.5 Terminals	173
69	A.6 Formal EDDL syntax.....	174
70	A.7 Formal dictionary syntax.....	255
71	Annex B (normative) Void (EDDL Builtin library)	256
72	Annex C (informative) EDD example	257
73	C.1 EDD example of a temperature transmitter	257
74	C.2 EDD example	258
75	Annex D (normative) Profiles of EDDL and Builtins	271
76	D.1 Conventions for profiles of EDDL and Builtins.....	271
77	D.2 Profiles for PROFIBUS and PROFINET	271
78	D.3 Profiles for FOUNDATION™ Fieldbus.....	276
79	D.4 Profiles for HART® Communication Foundation (HCF)	281
80	D.5 Profiles for Communication Servers.....	286
81	D.6 Data types	290
82	Annex E (informative) Historical background.....	296
83		
84	Figure 1 – Position of IEC 61804 related to other standards and products	15
85	Figure 2 – EDD generation process	22
86	Figure 3 – BLOCK_A	24
87	Figure 4 – CHART	25
88	Figure 5 – COLLECTION	25
89	Figure 6 – COMMAND	25
90	Figure 7 – COMPONENT	26
91	Figure 8 – COMPONENT_FOLDER	26
92	Figure 9 – COMPONENT_REFERENCE	

93	Figure 10 – COMPONENT_RELATION	26
94	Figure 11 – EDIT_DISPLAY	27
95	Figure 12 – FILE	27
96	Figure 13 – GRAPH	27
97	Figure 14 – GRID	27
98	Figure 15 – IMAGE	28
99	Figure 16 – LIKE	28
100	Figure 17 – LIST	28
101	Figure 18 – MENU	29
102	Figure 19 – RECORD	29
103	Figure 20 – REFERENCE_ARRAY	30
104	Figure 21 – REFRESH	30
105	Figure 22 – UNIT	30
106	Figure 23 – WRITE_AS_ONE	31
107	Figure 24 – SOURCE	31
108	Figure 25 – VALUE_ARRAY	31
109	Figure 26 – VARIABLE	32
110	Figure 27 – VARIABLE_LIST	32
111	Figure 28 – WAVEFORM	32
112	Figure 29 – EDDL import mechanisms	74
113	Figure 30 – MENU activation	99
114	Figure C.1 – Example of an operator screen using EDD	257
115		
116	Table 1 – Field attribute descriptions	20
117	Table 2 – DD_REVISION attribute	34
118	Table 3 – DEVICE_REVISION attribute	34
119	Table 4 – DEVICE_TYPE attribute	35
120	Table 5 – EDD_PROFILE attribute	35
121	Table 6 – EDD_VERSION attribute	35
122	Table 7 – MANUFACTURER attribute	36
123	Table 8 – MANUFACTURER_EXT attribute	36
124	Table 9 – AXIS attributes	36
125	Table 10 – MAX_VALUE, MIN_VALUE attribute	37
126	Table 11 – SCALING attribute	38
127	Table 12 – BLOCK_A attributes	38
128	Table 13 – CHARACTERISTIC attribute	39
129	Table 14 – PARAMETER attributes	40
130	Table 15 – AXIS_ITEMS attribute	40
131	Table 16 – CHART_ITEMS attribute	40
132	Table 17 – COLLECTION_ITEMS attribute	40
133	Table 18 – EDIT_DISPLAY_ITEMS attribute	41
134	Table 19 – FILE_ITEMS attribute	41
135	Table 20 – GRAPH_ITEMS attribute	41

136	Table 21 – GRID_ITEMS attribute	41
137	Table 22 – IMAGE_ITEMS attribute	42
138	Table 23 – LIST_ITEMS attribute	42
139	Table 24 – MENU_ITEMS attribute	42
140	Table 25 – METHOD_ITEMS attribute.....	43
141	Table 26 – PARAMETER_LISTS attributes	43
142	Table 27 – REFERENCE_ARRAY_ITEMS attribute	43
143	Table 28 – REFRESH_ITEMS attribute	44
144	Table 29 – SOURCE_ITEMS attribute.....	44
145	Table 30 – UNIT_ITEMS attribute	44
146	Table 31 – WAVEFORM_ITEMS attribute	44
147	Table 32 – WRITE_AS_ONE_ITEMS attribute.....	45
148	Table 33 – CHARTS attribute.....	45
149	Table 34 – LISTS attribute	45
150	Table 35 – GRAPHS attribute	46
151	Table 36 – GRIDS attribute	46
152	Table 37 – MENUS attribute	46
153	Table 38 – METHODS attribute.....	47
154	Table 39 – FILES attribute	47
155	Table 40 – PLUGIN_ITEMS attribute.....	47
156	Table 41 – PLUGINS attribute.....	48
157	Table 42 – BLOCK_B attributes	48
158	Table 43 – NUMBER attributes	49
159	Table 44 – TYPE attributes	49
160	Table 45 – CHART attributes	50
161	Table 46 – CYCLE_TIME attribute	50
162	Table 47 – LENGTH attribute.....	50
163	Table 48 – TYPE attribute.....	51
164	Table 49 – COLLECTION attributes	51
165	Table 50 – item-type	52
166	Table 51 – COMMAND attributes	53
167	Table 52 – OPERATION attribute.....	53
168	Table 53 – TRANSACTION attributes	54
169	Table 54 – REPLY and REQUEST attributes.....	54
170	Table 55 – INDEX attribute	55
171	Table 56 – BLOCK_B attribute	56
172	Table 57 – NUMBER attribute	56
173	Table 58 – SLOT attribute.....	56
174	Table 59 – SUB_SLOT attribute.....	57
175	Table 60 – HEADER attribute	57
176	Table 61 – API attribute	57
177	Table 62 – POST_RQSTRECEIVE_ACTIONS attributes	58
178	Table 63 – COMPONENT attribute.....	58

179	Table 64 – CAN_DELETE attribute	59
180	Table 65 – CHECK_CONFIGURATION attribute	59
181	Table 66 – COMPONENT_RELATIONS attribute	60
182	Table 67 – DECLARATION attribute	60
183	Table 68 – DETECT attribute	60
184	Table 69 – EDD attribute	61
185	Table 70 – INITIAL_VALUES attribute	61
186	Table 71 – REDUNDANCY attribute	61
187	Table 72 – SCAN attribute	62
188	Table 73 – SCAN_LIST attribute	62
189	Table 74 – BYTE_ORDER attribute	62
190	Table 75 – CONNECTION_POINT attribute	62
191	Table 76 – PRODUCT_URI attribute	63
192	Table 77 – COMPONENT_FOLDER attribute	63
193	Table 78 – COMPONENT_REFERENCE attribute	64
194	Table 79 – COMPONENT_RELATION attribute	64
195	Table 80 – COMPONENTS attribute	65
196	Table 81 – RELATION_TYPE attribute	65
197	Table 82 – ADDRESSING attribute	66
198	Table 83 – MAXIMUM_NUMBER attribute	66
199	Table 84 – MINIMUM_NUMBER attribute	66
200	Table 85 – REQUIRED_INTERFACE attribute	67
201	Table 86 – EDIT_DISPLAY attributes	67
202	Table 87 – EDIT_ITEMS attribute	67
203	Table 88 – DISPLAY_ITEM attributes	68
204	Table 89 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS attribute	69
205	Table 90 – FILE attributes	69
206	Table 91 – SHARED attribute	69
207	Table 92 – ON_UPDATE_ACTIONS attribute	70
208	Table 93 – GRAPH attributes	70
209	Table 94 – CYCLE_TIME attribute	71
210	Table 95 – X_AXIS attribute	71
211	Table 96 – GRID attributes	71
212	Table 97 – VECTORS attribute	72
213	Table 98 – ORIENTATION attribute	72
214	Table 99 – IMAGE attributes	73
215	Table 100 – PATH attribute	73
216	Table 101 – LINK attribute	73
217	Table 102 – Importing Device Description	74
218	Table 103 – Redefinition attributes	76
219	Table 104 – Redefinition rules for AXIS attributes	76
220	Table 105 – Redefinition rules for BLOB attributes	76
221	Table 106 – Redefinition rules for BLOCK_A attributes	77

222	Table 107 – Redefinition rules for BLOCK_B attributes	78
223	Table 108 – Redefinition rules for CHART attributes	78
224	Table 109 – Redefinition rules for COLLECTION attributes	79
225	Table 110 – Redefinition rules for COMMAND attributes	79
226	Table 111 – Redefinition rules for COMPONENT attributes	80
227	Table 112 – Redefinition rules for COMPONENT_FOLDER attributes	80
228	Table 113 – Redefinition rules for COMPONENT_REFERENCE attributes	81
229	Table 114 – Redefinition rules for COMPONENT_RELATION attributes	81
230	Table 115 – Redefinition rules for EDIT_DISPLAY attributes	82
231	Table 116 – Redefinition rules for FILE attributes	82
232	Table 117 – Redefinition rules for GRAPH attributes	82
233	Table 118 – Redefinition rules for GRID attributes	83
234	Table 119 – Redefinition rules for IMAGE attributes	83
235	Table 120 – Redefinition rules for INTERFACE attributes	83
236	Table 121 – Redefinition rules for LIST attributes	84
237	Table 122 – Redefinition rules for MENU attributes	84
238	Table 123 – Redefinition rules for METHOD attributes	85
239	Table 124 – Redefinition rules for PLUGIN attributes	85
240	Table 125 – Redefinition rules for RECORD attributes	85
241	Table 126 – Redefinition rules for REFERENCE_ARRAY attributes	86
242	Table 127 – Redefinition rules for RESPONSE_CODES attributes	86
243	Table 128 – Redefinition rules for SOURCE attributes	86
244	Table 129 – Redefinition rules for TEMPLATE attributes	87
245	Table 130 – Redefinition rules for VALUE_ARRAY attributes	87
246	Table 131 – Redefinition rules for VARIABLE attributes	88
247	Table 132 – Redefinition rules for VARIABLE_LIST attributes	89
248	Table 133 – Redefinition rules for WAVEFORM attributes	89
249	Table 134 – INTERFACE attribute	90
250	Table 135 – DECLARATION attribute	90
251	Table 136 – LIKE attributes	90
252	Table 137 – LIST attributes	91
253	Table 138 – TYPE attribute	91
254	Table 139 – CAPACITY attribute	92
255	Table 140 – COUNT attribute	92
256	Table 141 – MENU attribute	92
257	Table 142 – ITEMS attribute	93
258	Table 143 – ACCESS attribute	94
259	Table 144 – EXIT_ACTIONS, INIT_ACTIONS, POST_EDIT_ACTIONS,	
260	PRE_EDIT_ACTIONS, POST_READ_ACTIONS, PRE_READ_ACTIONS,	
261	POST_WRITE_ACTIONS, PRE_WRITE_ACTIONS attributes	95
262	Table 145 – STYLE attribute	97
263	Table 146 – METHOD attributes	100
264	Table 147 – ACCESS attributes	100
265	Table 148 – CLASS attributes	100

266	Table 149 – TYPE attributes	102
267	Table 150 – RECORD attributes	102
268	Table 151 – REFERENCE_ARRAY attribute	103
269	Table 152 – ELEMENTS attribute	103
270	Table 153 – REFRESH attributes	104
271	Table 154 – UNIT attributes	104
272	Table 155 – WRITE_AS_ONE attribute	104
273	Table 156 – RESPONSE_CODES attributes	105
274	Table 157 – SOURCE attributes	105
275	Table 158 – Y_AXIS attribute	107
276	Table 159 – TEMPLATE attributes	107
277	Table 160 – DEFAULT_VALUES attribute	108
278	Table 161 – VALUE_ARRAY attributes	108
279	Table 162 – NUMBER_OF_ELEMENTS attribute	108
280	Table 163 – TYPE attribute	109
281	Table 164 – VARIABLE attributes	109
282	Table 165 – CLASS attributes	110
283	Table 166 – TYPE attributes	112
284	Table 167 – DOUBLE, FLOAT, INTEGER, UNSIGNED_INTEGER attributes	113
285	Table 168 – DATE, DATE_AND_TIME, DURATION, TIME, TIME_VALUE attributes	116
286	Table 169 – BIT_ENUMERATED attributes	117
287	Table 170 – status-class attributes	118
288	Table 171 – ALL, AO, DV, TV attributes	119
289	Table 172 – Enumerated types attributes	119
290	Table 173 – Index type attributes	120
291	Table 174 – String types attributes	121
292	Table 175 – CONSTANT_UNIT attribute	122
293	Table 176 – DEFAULT_VALUE attribute	122
294	Table 177 – INITIAL_VALUE attribute	123
295	Table 178 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS, POST_READ_ACTIONS,	
296	PRE_READ_ACTIONS, POST_WRITE_ACTIONS, PRE_WRITE_ACTIONS,	
297	REFRESH_ACTIONS attributes	123
298	Table 179 – POST_USERCHANGE_ACTIONS, POST_RQSTUPDATE_ACTIONS	
299	attributes	125
300	Table 180 – VARIABLE_LIST attributes	126
301	Table 181 – WAVEFORM attributes	126
302	Table 182 – TYPE attribute	127
303	Table 183 – XY attribute	127
304	Table 184 – YT attribute	128
305	Table 185 – HORIZONTAL attribute	129
306	Table 186 – VERTICAL attribute	129
307	Table 187 – EXIT_ACTIONS, INIT_ACTIONS, REFRESH_ACTIONS attribute	130
308	Table 188 – KEY_POINTS attribute	130
309	Table 189 – X_VALUES, Y_VALUES attribute	

310	Table 190 – Y_AXIS attribute.....	131
311	Table 191 – CLASSIFICATION attribute.....	132
312	Table 192 – COMPONENT_PARENT attribute.....	133
313	Table 193 – COMPONENT_PATH attribute.....	134
314	Table 194 – DEFINITION attributes	134
315	Table 195 – EMPHASIS attribute	134
316	Table 196 – HANDLING attribute	135
317	Table 197 – HEIGHT/WIDTH attribute.....	135
318	Table 198 – HELP attribute.....	136
319	Table 199 – LABEL attribute	136
320	Table 200 – LINE_COLOR attribute	136
321	Table 201 – LINE_TYPE attribute	137
322	Table 202 – MEMBERS attributes	137
323	Table 203 – PROTOCOL attribute.....	138
324	Table 204 – RESPONSE_CODES attribute.....	139
325	Table 205 – SUPPLIED_INTERFACE attribute.....	139
326	Table 206 – VALIDITY attributes.....	139
327	Table 207 – PRIVATE attributes	140
328	Table 208 – VISIBILITY attributes.....	140
329	Table 209 – WRITE_MODE attributes	140
330	Table 210 – IDENTITY attribute	141
331	Table 211 – IF, SELECT conditional	142
332	Table 212 – Referencing an EDD instance.....	142
333	Table 213 – Referencing elements of VARIABLE	143
334	Table 214 – Referencing elements of RECORD	143
335	Table 215 – Referencing elements of VALUE_ARRAY	143
336	Table 216 – Referencing members of COLLECTION.....	143
337	Table 217 – Referencing members of REFERENCE_ARRAY	144
338	Table 218 – Referencing members of VARIABLE_LISTS	144
339	Table 219 – Referencing members of a BLOCK_A PARAMETERS	144
340	Table 220 – Referencing members of BLOCK_A PARAMETER_LISTS	145
341	Table 221 – Referencing members of BLOCK_A LOCAL_PARAMETER.....	145
342	Table 222 – Referencing BLOCK_A CHARACTERISTICS	145
343	Table 223 – Referencing members of FILE	145
344	Table 224 – Referencing elements of LIST	146
345	Table 225 – Referencing members of CHART.....	146
346	Table 226 – Referencing members of GRAPH	146
347	Table 227 – Referencing members of SOURCE	147
348	Table 228 – Referencing AXIS of a GRAPH. SOURCE, WAVEFORM.....	147
349	Table 229 – Referencing PARAMETERS of specific BLOCK_A instance	147
350	Table 230 – Referencing LOCAL_PARAMETERS of specific BLOCK_A instance	148
351	Table 231 – Referencing CHARACTERISTICS of specific BLOCK_A instance	148
352	Table 232 – Referencing CHARTS of specific BLOCK_A instance	148

353	Table 233 – Referencing LISTS of specific BLOCK_A instance	149
354	Table 234 – Referencing GRAPHS of specific BLOCK_A instance	149
355	Table 235 – Referencing GRIDS of specific BLOCK_A instance	149
356	Table 236 – Referencing MENUS of specific BLOCK_A instance	150
357	Table 237 – Referencing METHODS of specific BLOCK_A instance	150
358	Table 238 – Referencing a COMPONENT instance	150
359	Table 239 – Referencing a COMPONENT type	151
360	Table 240 – Referencing FILES of specific BLOCK_A instance	151
361	Table 241 – Referencing PLUGINS of specific BLOCK_A instance	151
362	Table 242 – string as a string literal	152
363	Table 243 – string as a string variable	152
364	Table 244 – String as an enumeration value	152
365	Table 245 – String as a dictionary reference	153
366	Table 246 – Referencing HELP and LABEL attributes of EDD instances	153
367	Table 247 – String operation	153
368	Table 248 – Format specifier	154
369	Table 249 – Primary expressions	154
370	Table 250 – Attribute values of VARIABLES	155
371	Table 251 – AXIS Attribute Values	156
372	Table 252 – BLOB Attribute Values	156
373	Table 253 – LIST Attribute Values	156
374	Table 254 – ARRAY Attribute Values	156
375	Table 255 – Unary expressions	156
376	Table 256 – Multiplicative operators	157
377	Table 257 – Additive operators	157
378	Table 258 – Shift operators	157
379	Table 259 – Relational operators	158
380	Table 260 – Equality operators	158
381	Table 261 – Text dictionary attributes	159
382	Table 262 – PLUGIN attributes	159
383	Table 263 – UUID attribute	160
384	Table 264 – BLOB attributes	160
385	Table A.1 – Conventions for integer constants	164
386	Table A.2 – Using escape sequences in string literals	165
387	Table A.3 – Language code examples for string literals	166
388	Table A.4 – Precedence and associativity for EDDL operators	166
389	Table A.5 – Operations for VARIABLES or METHOD local variables	168
390	Table A.6 – EDDL keywords	169
391	Table D.7 – Profile selection tables	271
392	Table D.8 – EDDL Formal Definition profile tables	271
393	Table D.9 – Contents of selection tables	271
394	Table D.10 – EDDL element selection for PROFIBUS&PROFINET	272
395	Table D.11 – EDDL element selection for Fieldbus Foundation	276

396	Table D.12 – EDDL element selection for HCF.....	281
397	Table D.13 – EDDL element selection for Communication Servers.....	286
398	Table D.14 – METHOD DEFINITIONS data types	291
399	Table D.15 – VARIABLE TYPES	291
400	Table D.16 – DATE coding.....	293
401	Table D.17 – DATE_AND_TIME coding	293
402	Table D.18 – DURATION coding	293
403	Table D.19 – TIME coding	293
404	Table D.20– TIME_VALUE coding (four octets).....	294
405	Table D.21 – TIME_VALUE coding (eight octets)	294
406	Table D.22 – PACKED_ASCII coding	295
407	Table D.23 – BOOLEAN coding	295
408		
409		
410		

411 INTERNATIONAL ELECTROTECHNICAL COMMISSION
412
413
414 **FUNCTION BLOCKS (FB) FOR PROCESS CONTROL –**
415
416 **Part 3: Electronic Device Description Language (EDDL)**
417

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**FUNCTION BLOCKS (FB) FOR PROCESS CONTROL AND ELECTRONIC
DEVICE DESCRIPTION LANGUAGE (EDDL) –**

Part 3: EDDL syntax and semantics

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.
- International Standard IEC 61804-3 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition cancels and replaces the second edition published in 2010. This edition constitutes a technical revision.

The main changes with respect to the previous edition are listed below:

- Builtins and their profiles removed and relocated into IEC 61804-5.
- The following extensions are integrated in the EDDL specification to meet FDI requirements:
 - Plugin for Menu keyword
 - Component construct for communication server requirements (additional attributes)
 - Extension of the class attribute

- 473 — New attribute – “private” is added to some communication objects
- 474 • The following changes will be integrated in the EDDL based on EDDL harmonization:
- 475 — Removed some unused features
- 476 — Harmonize some profile features
- 477 The text of this standard is based on the following documents:

FDIS	Report on voting
65E/XXX/FDIS	65E/XXX/RVD

- 478
- 479 Full information on the voting for the approval of this standard can be found in the report on
- 480 voting indicated in the above table.
- 481 This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.
- 482 The committee has decided that the contents of this publication will remain unchanged until
- 483 the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data
- 484 related to the specific publication. At this date, the publication will be
- 485 • reconfirmed,
 - 486 • withdrawn,
 - 487 • replaced by a revised edition, or
 - 488 • amended.
- 489 A bilingual version of this publication may be issued at a later date.

490 The National Committees are requested to note that for this publication the stability date

491 is 2016.

492 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE

493 DELETED AT THE PUBLICATION STAGE.

494

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.

495

496

497

INTRODUCTION

The EDDL fills the gap between the conceptual Function Block specification of IEC 61804-2 and a product implementation. It allows the manufacturers to use the same description method for devices based on different technologies and platforms. Figure 1 shows these aspects.

IEC 61804 has the general title "Function blocks (FB) for process control and EDDL" and consists of the following parts:

Part 2: FB concept

Part 3: EDDL syntax and semantics

Part 4: EDD interpretation

Part 5: EDDL Builtin library

Part 6: Meeting the requirements for integrating fieldbus devices in engineering tools for field devices

This part of IEC 61804 has integrated some parts of IEC/TS 61804-1:2003, which is withdrawn January 2013.

Headings named 'Void' are used to retain the numbering of previous editions.

The EDDL may also be used for the description of product properties in other domains as industrial automation. Industrial automation may include devices such as generic digital and analog input/output modules, motion controllers, human machine interfaces, sensors, closed-loop controllers, encoders, hydraulic valves, and programmable controllers.

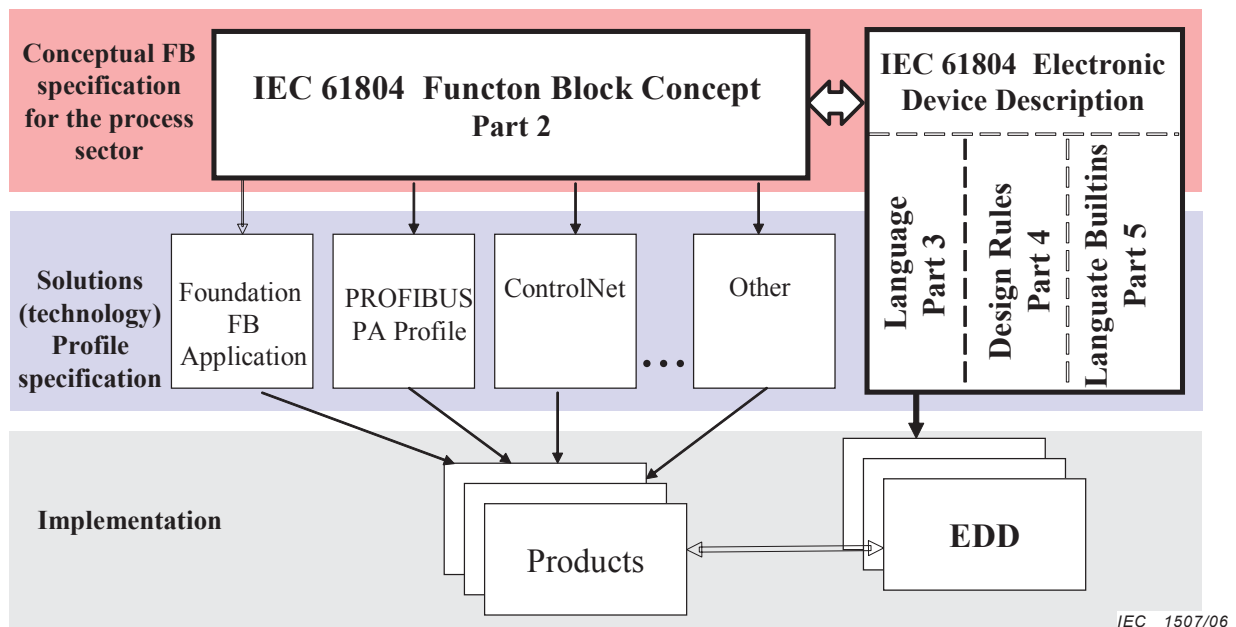


Figure 1 – Position of IEC 61804 related to other standards and products

FUNCTION BLOCKS (FB) FOR PROCESS CONTROL AND ELECTRONIC DEVICE DESCRIPTION LANGUAGE (EDDL)–

Part 3: EDDL syntax and semantics

1 Scope

This International Standard specifies the Electronic Device Description Language (EDDL) technology, which enables the integration of real product details using the tools of the engineering life cycle.

This part of IEC 61804 specifies EDDL as a generic language for describing the properties of automation system components. EDDL is capable of describing

- device parameters and their dependencies;
- device functions, for example, simulation mode, calibration;
- graphical representations, for example, menus;
- interactions with control devices;
- graphical representations:
 - enhanced user interface;
 - graphing system.
- persistent data store.

EDDL is to be used to create Electronic Device Description (EDD). This EDD is used with appropriate tools to generate interpretative code to support parameter handling, operation, and monitoring of automation system components such as remote I/Os, controllers, sensors, and programmable controllers. Tool implementation is outside the scope of this standard.

This International Standard specifies the semantic and lexical structure in a syntax-independent manner. A specific syntax is defined in Annex A, but it is possible to use the semantic model also with different syntaxes.

2 Normative references

The following referenced documents are indispensable for the application 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 61499-1:2005, *Function blocks – Part 1: Architecture*

IEC 61804-5:2013, *Devices and integration in enterprise systems – Function blocks (FB) for process control and EDDL – Part 5: EDDL Builtin library*

ISO/IEC 2022, *Information technology – Character code structure and extension techniques*

ISO/IEC 2375:2003, *Information technology – Procedure for registration of escape sequences and coded character sets*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

ISO/IEC 9899, *Programming languages – C*

565 ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character*
566 *Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

567 ISO 639, *Code for the representation of names of languages*

568 ISO 3166-1:2006, *Codes for the representation of names of countries and their subdivisions –*
569 *Part 1: Country codes*

570 IEEE 754:1985 (R1990), *Binary Floating-Point Arithmetic*

571 RFC 3629:2003, *UTF-8, User Datagram Protocol*, available at
572 <<http://www.ietf.org/rfc/rfc0768.txt> >

573 **3 Terms, definitions, abbreviated terms and acronyms**

574 **3.1 Terms and definitions**

575 For the purposes of this document, the terms and definitions given in IEC 60050-351 apply
576 partially, in IEC 61804-2 and the following apply.

577 **3.1.1**

578 **builtin**

579 predefined subroutine executed by the EDD application, e.g. for communication and display
580 purpose.

581 **3.1.2**

582 **device**

583 independent physical entity capable of performing one or more specified functions in a
584 particular context and delimited by its interfaces

585 [SOURCE: IEC 61499-1]

586 **3.1.3**

587 **EDD application**

588 program using the EDD, or any translated form, which offers functionality like communication
589 representation, data representation, graphical representation, etc.

590 **3.1.4**

591 **EDDL processor**

592 processor or program, which translates the EDD into an executable form that can be
593 processed by an EDD application

594 **3.1.5**

595 **EDDL profile**

596 selection of the supported elements of the EDDL lexical structure including the syntax
597 definitions for a number of specific consortia

598 **3.1.6**

599 **Electronic Device Description Language**

600 **EDDL**

601 a formal language for describing devices and communication components

602 **3.1.7**

603 **Electronic Device Description**

604 **EDD**

605 data collection containing the device parameter(s), their dependencies, their graphical
606 representation and a description of the data sets which are transferred.

607 Note 1 to entry: The Electronic Device Description is created using the EDDL.

608 **3.1.8**609 **Electronic Device Description Compiler**

610 compiler which translates the EDD source in an internal format that is used by the EDD
611 Interpreter

612 **3.1.9**613 **Electronic Device Description Interpreter**614 **EDDI**

615 Interpreter which uses the EDD source or an internal format that is given by the EDDL
616 Compiler to provide the EDD information to the EDD user

617 **3.1.10**618 **function block (function block instance)**

619 software functional unit comprising an individual, named copy of a data structure and
620 associated operations specified by a corresponding FB type

621 [SOURCE: IEC 61804-2]

622 Note 1 to entry: Typical operations of a FB include modification of the values of the data in its associated data
623 structure.

624 **3.1.11**625 **hardware**

626 physical equipment, as opposed to programs, procedures, rules and associated
627 documentation

628 [SOURCE: ISO/AFNOR Dictionary of Computer Science]

629 **3.1.12**630 **preprocessor**

631 part of the run-time environment that transforms the information given in an EDD

632 **3.1.13**633 **preprocessor directives**

634 description of conditions for filtering the EDD code before compilation or interpretation

635 Note 1 to entry: For example, a preprocessor directive providing the facility to define names for constants or to
636 write macros to make code easier to read.

637 **3.1.14**638 **software**

639 intellectual creation comprising the programs, procedures, rules and any associated
640 documentation pertaining to the operation of a system

641 [SOURCE: IEC 61499-1]

642 **3.2 Abbreviated terms and acronyms**

ANSI American National Standard Institute

ANSI C American National Standard Institute for the programming language C
(see ISO/IEC 9899)

ASCII American Standard Code for Information Interchange (see ISO/IEC 10646-1)

CP Communication Profile

CPF Communication Profile Family

EDD Electronic Device Description

EDDL Electronic Device Description Language

EUC Extended Unit Code (see ISO/IEC 2022)

FB	Function Block
FF	Fieldbus Foundation ¹
HCF	HART® ² Communication Foundation
HMI	Human Machine Interface
I/O	Input/Output
ID	Identifier
OSI	Open Systems Interconnection
PI	PROFIBUS & PROFINET ³ International
PNO	PROFIBUS Nutzerorganisation

643

644 **4 Conformance statement**

645 A statement of conformance to a profile of this International Standard shall be stated⁴ as
646 either

647 – compliance to IEC 61804-3:2013 Annex D.n <Type>

648 where the Type within the angle brackets < > is optional and the angle brackets are not to be
649 included.

650 The Type may be, for example, the name of the communication protocol used like PROFIBUS,
651 Foundation Fieldbus, or HART Communication Foundation, or in an encoded way according to
652 the Communication Profiles specified in IEC 61784-1 or in IEC 61784-2.

653 Product standards shall not include any conformity assessment aspects (including QM
654 provisions), neither normative nor informative, other than provisions for product testing
655 (evaluation and examination).

656 **5 Conventions for lexical structures**

657 The EDDL is generally described using lexical structures in which the elements and the
658 presence of fields are specified. A general form of the lexical structures is shown below.

659 **ABC field1, field2**

660 ABC is a lexical element. This element shall be coded in a concrete syntax. It is not required
661 to code this element with the name "ABC". It is also possible to code this element, for
662 example, with a tag number.

663 Field1 and field2 are fields of the lexical element ABC. Each field is mandatory and may have
664 more than one attribute. If a field has attributes, the presence of the attributes is specified in a

¹ FOUNDATION™ Fieldbus is the trade name of the consortium Fieldbus Foundation (non-profit organization). This information is given for the convenience of users of this standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name. Use of the trade names requires permission from the trade name holder.

² HART® is a registered trademark of the consortium HART Communication Foundation (HCF) (non-profit organization). This information is given for the convenience of users of this standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name. Use of the trade names requires permission from the trade name holder.

³ PROFIBUS & PROFINET are trade names of PROFIBUS Nutzerorganisation e.V. (PNO) as part of PROFIBUS & PROFINET International. PNO is a non-profit trade organization to support the fieldbus technologies PROFIBUS & PROFINET. This information is given for the convenience of users of this standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the tradename. Use of the trade names requires permission from the trade name holder.

⁴ In accordance with the ISO/IEC Directives.

665 table. A comma separates Field1 and 2. The comma is a lexical element and is not coded
666 explicitly.

667 If a field has additional attributes, the attributes are defined in a table. The table layout and
668 the possible usage qualifiers are shown in Table 1.

669 **Table 1 – Field attribute descriptions**

Usage	Attribute	Description
m	yyy	The presence of this attribute is mandatory.
o	xxx	The presence of this attribute is optional.
s	z1	The presence of this attribute is selectable with other attributes, which are also marked with “s” in the usage column). Only one of the selectable attributes (z1 or z2) is present.
s	z2	The presence of this attribute is selectable with other attributes, which are also marked with “s” in the usage column). Only one of the selectable attributes (z1 or z2) is present.
c	uuu	The presence of this attribute is conditional and it is only present if the condition is true.

670

671 The characters in the Usage column have the following meanings.

672 m: this attribute is mandatory and shall be present.

673 o: this attribute is optional and need not be present.

674 s: this attribute is a selection. One, and only one, of the fields marked with s (Z1 or Z2)
675 shall be present.

676 c: this attribute is conditional. The condition is described in the Description column.

677 In the Attribute column, if more than one attribute exists, which have the same usage, the
678 attributes are sorted in alphabetical order.

679 **ABC field1+**

680 The plus (+) behind field1 is used to indicate that field1 is used at least once. It may be used
681 more than once.

682 **ABC field2***

683 The asterisk (*) behind field2 is used to indicate that field2 is optional and need not be used.
684 If it is used it may be used more than once.

685 **ABC [field1, field2]+**

686 The elements field1 and field2 in the square brackets [] are an unsorted list. The plus behind
687 the closing bracket indicates that field1 and field2 are used at least once and may be used
688 more than once as group.

689 **ABC field1, (field2, field3)<exp>**

690 <exp> indicates that field2 and field3 are used in conjunction with the conditional expression,
691 (conditional constructs are specified in 0). The expression <exp> refers only to the fields
692 within the preceding curved brackets (). Usage of conditional expressions is optional.

693 **6 EDD and EDDL model**

694 **6.1 Overview of EDD and EDDL**

695 An Electronic Device Description (EDD) contains all device parameter(s) of an automation
696 system component. The technology used to describe an EDD is called Electronic Device
697 Description Language (EDDL). The EDDL provides a set of scalable basic language elements
698 to handle simple, complex or modular devices. The EDDL is a descriptive language based on
699 an ASCII format with clear separation between data and program.

700 Data in a text field, which is marked with a country code like Japanese, may use multi-byte
701 code.

702 **6.2 EDD architecture**

703 From the viewpoint of the ISO/OSI model (ISO/IEC 7498-1), an EDD is above Layer 7.
704 However, the EDD application uses the communication system to transfer its information. An
705 EDD contains constructs that support mapping to a supporting communication system.

706 The device manufacturer defines the objects, which are reflected by the logical representation
707 of the objects within an EDD application. For that reason, EDDL has language elements,
708 which map the EDD data to the data representation of the communication system, so that the
709 user of an EDD does not need to know the physical location (address) of a device object.

710 EDD describes the management of information to be displayed to the user. The specific
711 representation of such visualization is not part of an EDD or EDDL definitions.

712 **6.3 Concepts of EDD**

713 The manufacturer of a device or of an automation system component describes the properties
714 of the device by using the EDDL. The resulting EDD contains information such as:

- 715 • description of the device parameters;
- 716 • description of parameter dependencies;
- 717 • logical grouping of the device parameters;
- 718 • selection and execution of supported device functions;
- 719 • description of the transferred data sets.

720 Depending on the required usage, the EDD may be physically located

- 721 • in a device;
- 722 • in an external data storage medium such as a compact disk, floppy or a server;
- 723 • partially distributed in the device and an external storage medium.

724 EDD supports text strings (common terms, phrases etc.) in more than one language (English,
725 German, French, etc.). Text strings may be stored in separate dictionaries. There may be
726 more than one dictionary for one EDD.

727 An EDD implementation includes sufficient information about the target device, for example,
728 manufacturer, device type, revision, etc. This is used to match a specific EDD to a specific
729 device.

730 **6.4 Principles of the EDD development process**

731 **6.4.1 General**

732 The creation of an EDD is a three-stage process: EDD source generation, EDD preprocessing
733 and EDD compilation.

734 **6.4.2 EDD source generation**

735 The EDD generation process is shown in Figure 2.

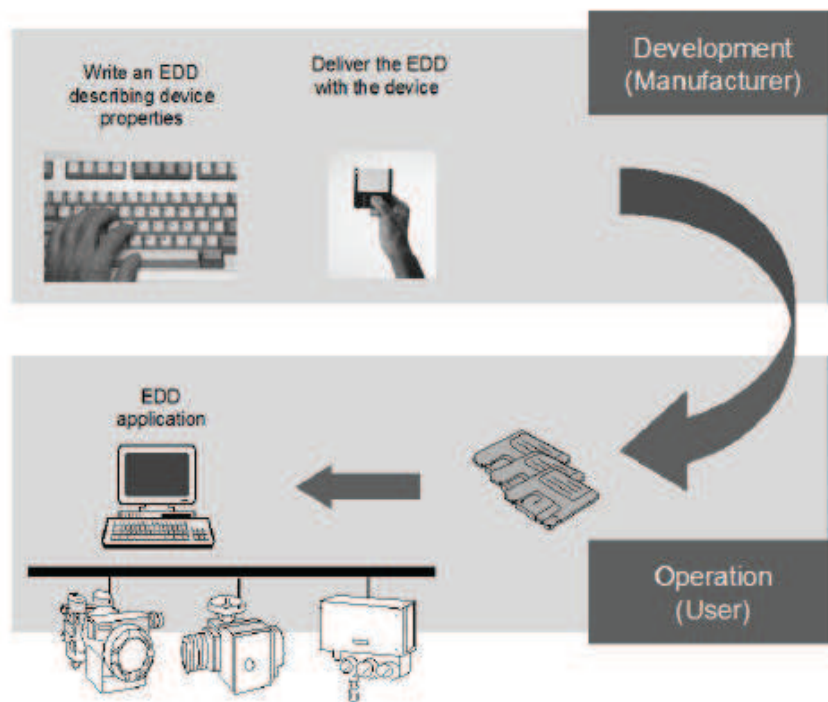


Figure 2 – EDD generation process

IEC 1511/06

The device manufacturer writes an appropriate EDD for his device and delivers both (EDD and device). If the automation system supports the EDD method, a system integration can be made by the user.

The EDDs for devices may be embedded in device memory or delivered using separate storage media or downloaded from an appropriate network server. The EDD is 'interpreted' or 'browsed' by an EDD-application. EDDs are normally stored as source files or preprocessed files.

6.4.3 EDD preprocessing

In the preprocessing stage an EDD preprocessor generates a consistent EDD representation suitable for final compilation.

Preprocessing supports, for example, substitution of definitions and inclusion of external text. The output of the preprocessor is a complete EDD without any preprocessor directive.

6.4.4 EDD compilation

The EDD compilation stage produces an EDD application internal representation from a preprocessed EDD to be used in the EDD application.

6.5 Interrelations between the lexical structure and formal definitions

The lexical structure of EDDL and its elements are described in Clause 7. Formal definitions and syntax for each EDDL element are given in Annex A. The lexical structure and its formal description use the same name.

NOTE Instead of the specified formal definitions and syntax in Annex A, another specification may be developed as an additional option.

6.6 Builtins

Builtins are predefined subroutines which are executed in the EDD application.

EXAMPLE A hand-held terminal is a simple device having a small display and limited cursor functions. For this type of device, a Builtin could be specified to provide display entry using only up/down right/left cursor actions.

The library of Builtins is defined in IEC 61804-5.

6.7 Profiles

EDDL is a harmonized specification of existing legacy EDD concepts. Concrete EDD applications use a subset of the EDDL specification. Selection of EDDL elements and Builtins is made from the profiles defined in Annex D.

In addition to EDD profiles, implementing consortia also publish “Device Profiles”, which are used to support interchangeability of compliant devices. These Device Profiles may be described using the EDDL specified in this document.

7 Electronic Device Description Language

7.1 Overview

7.1.1 EDDL features

The EDDL is a structured and interpretative language for describing device properties. Also the interactions between the devices and the EDD run-time environment are incorporated in the EDDL. The EDDL provides a set of language elements for these purposes.

For a specific EDD implementation it is not necessary to use all of the elements provided by the language.

Compatible subsets of EDDL are permitted and may be specified using profiles (for example, choice of constructs, number of recursions, and selection of options). Profiles supported by some industrial consortia are specified in Annex D. EDD developers are required to identify within each device details as to which profile has been used.

7.1.2 Syntax representation

The specification of the EDDL language in this clause of the standard uses an abstract syntax. The abstract lexical structure in this section is converted into specific syntaxes specified in Annex A by the element name (for example, VARIABLE in the lexical structure equates to the keyword “VARIABLE” in the annex).

Beside the defined syntax in this standard, it is possible to use other syntax definitions, which may be added in future to allow other features and representations for future progress.

7.1.3 EDD language elements

The language is structured in the following language elements:

- identification element;
- basic construction elements;
- special elements.

Frequently used text strings, for example, help text and multi-lingual lists of labels should be separated in a text dictionary (see 7.41).

Identification information shall be the first entry in every EDD file and shall appear only once. The identification information uniquely identifies the version of EDDL used, together with the specific device type, model codes and revision details covered by the EDD file.

7.1.4 Basic construction elements

7.1.4.1 General

These basic constructs have been specified to support descriptions of devices used within industrial control applications, together with their properties and functionality.

Some constructs have similar names and functions while differing in their detailed specification. This additional variety has been included to ensure compatibility with several existing description languages. Appropriate mapping cross-references are given by profiles.

Each of the basic constructs has a set of attributes associated with it. Attributes can also have subattributes, which refine the definition of the attribute and hence the definition of the construct itself.

810 The definition of an attribute may be static or dynamic. A static attribute definition never
 811 changes, while a dynamic attribute definition may change in order to accommodate parameter
 812 value changes.

813 7.1.4.2 AXIS

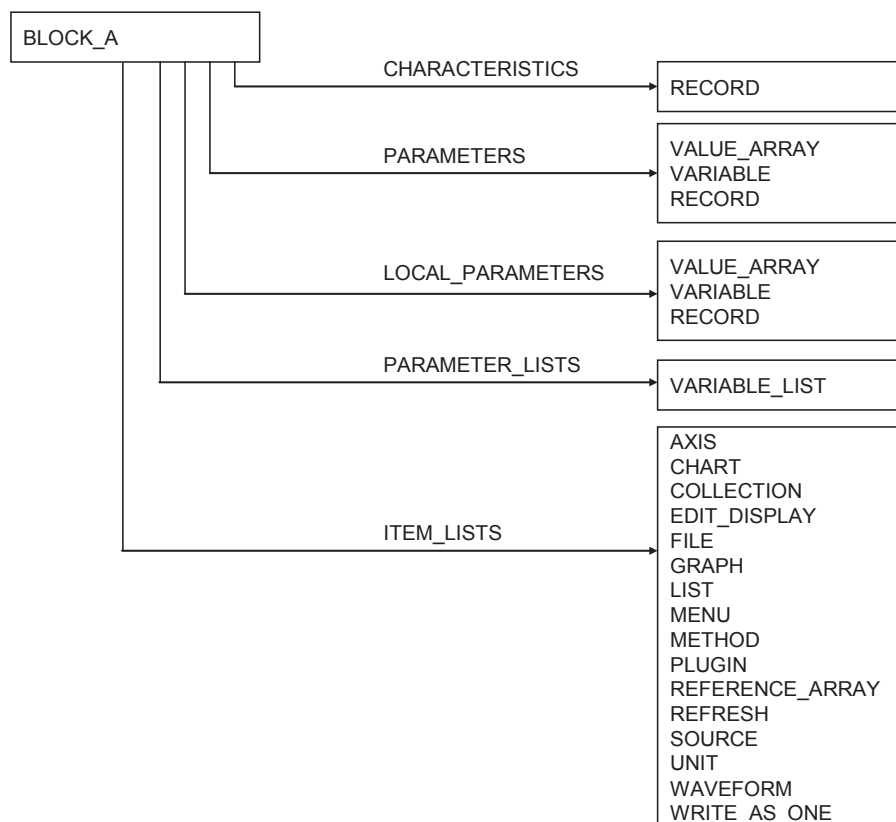
814 AXIS describes the axis of a CHART or GRAPH (see 7.3).

815 7.1.4.3 BLOB

816 BLOB describes a Binary Large Object used for transferring binary data to or from a field
 817 device (see 7.43).

818 7.1.4.4 BLOCK_A

819 BLOCK_A is a logical grouping of CHARACTERISTICS, PARAMETERS,
 820 PARAMETER_LISTS, and ITEM_LISTS, see Figure 3. To access one item of BLOCK_A, the
 821 instance of the block should be used (see 7.4.1 and Figure 3).



822
 823 **Figure 3 – BLOCK_A**

IEC 1512/06

824 7.1.4.5 BLOCK_B

825 BLOCK_B is used to structure variables in instances of logical block types. For accessing a
 826 VARIABLE within a BLOCK_B construct, the basic construct COMMAND is used to provide
 827 the relative addressing (see 7.4.2).

828 7.1.4.6 CHART

829 CHART describes a chart used to display data from a device in the EDD application (see 7.5
 830 and Figure 4).

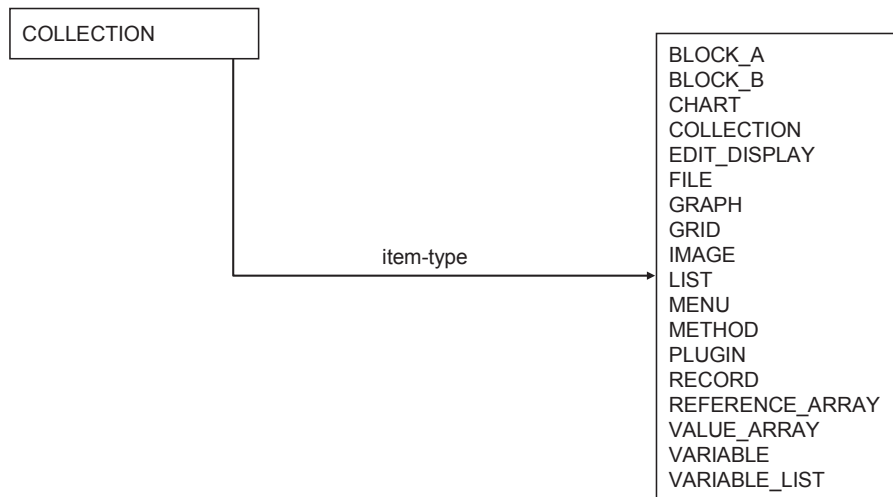


IEC 1513/06

Figure 4 – CHART

7.1.4.7 COLLECTION

COLLECTION describes logical groupings of data (see 7.6 and Figure 5).

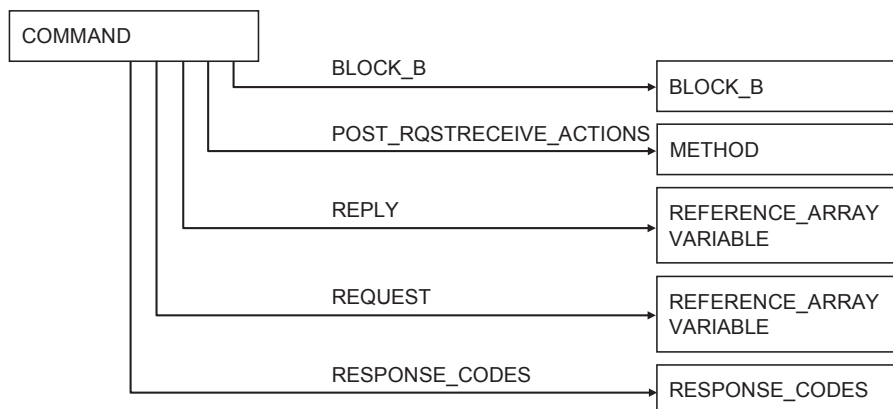


IEC 1514/06

Figure 5 – COLLECTION

7.1.4.8 COMMAND

COMMAND describes the structure and the addressing of the variables in the device (see 7.7 and Figure 6).



IEC 1515/06

Figure 6 – COMMAND

7.1.4.9 COMPONENT

COMPONENT describes the configuration behavior of the device that is described in the EDD (see 7.8 and Figure 7).

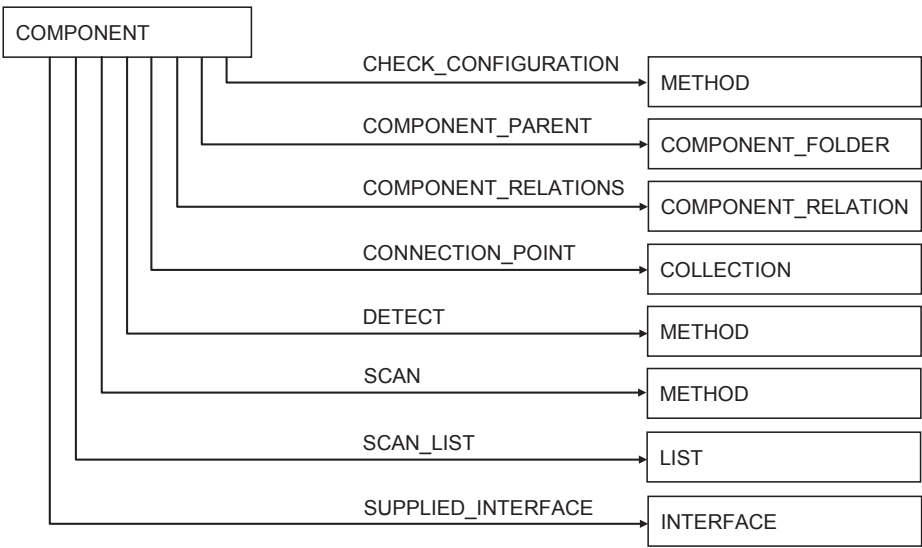


Figure 7 – COMPONENT

7.1.4.10 COMPONENT_FOLDER

COMPONENT_FOLDER describes a folder in a catalog. Folders allow groupings of components in a catalog (see 7.9 and Figure 8).



Figure 8 – COMPONENT_FOLDER

7.1.4.11 COMPONENT_REFERENCE

COMPONENT_REFERENCE refers to a COMPONENT or a COMPONENT_FOLDER (see 7.10 and Figure 9).



Figure 9 – COMPONENT_REFERENCE

7.1.4.12 COMPONENT_RELATION

COMPONENT_RELATION specifies a relation between COMPONENTs, e.g. a module and one or more other modules (see 7.11 and Figure 10).

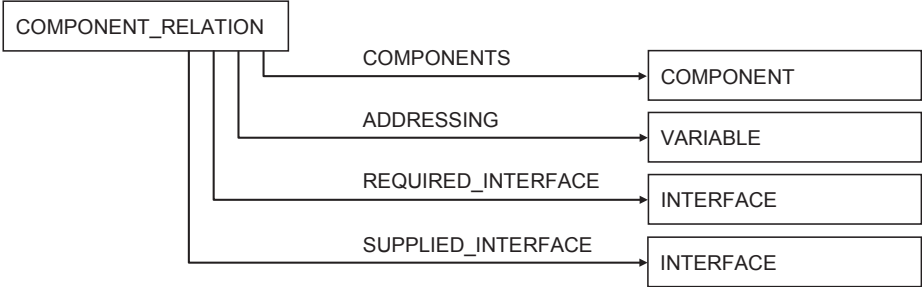


Figure 10 – COMPONENT_RELATION

7.1.4.13 EDIT_DISPLAY

EDIT_DISPLAY describes how the data will be represented to a user by a display device (see 7.14 and Figure 11).

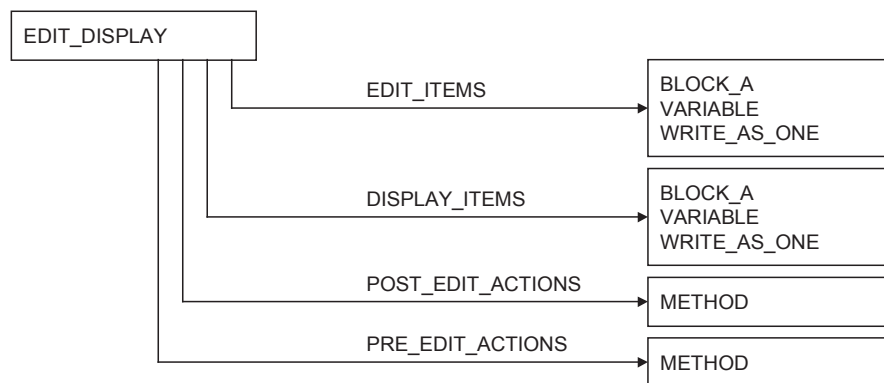


Figure 11 – EDIT_DISPLAY

IEC 1517/06

7.1.4.14 FILE

FILE describes an area of persistent storage (see 7.15 and Figure 12).

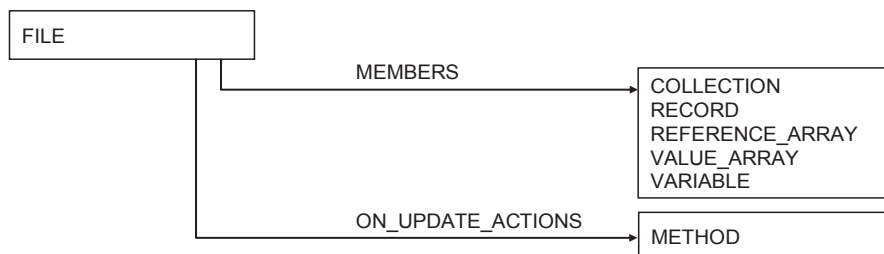


Figure 12 – FILE

IEC 1518/06

7.1.4.15 GRAPH

GRAPH describes a graph used to display data from a device in the EDD application (see 7.16 and Figure 13).

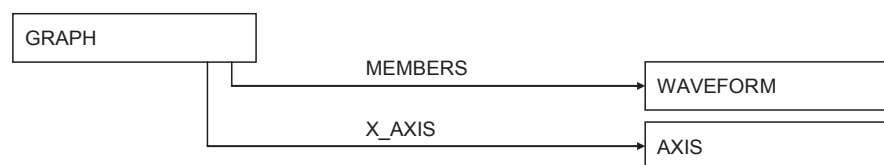


Figure 13 – GRAPH

IEC 1519/06

7.1.4.16 GRID

GRID describes a matrix used to display data from a device in the EDD application (see 7.17 and Figure 14).



Figure 14 – GRID

IEC 1520/06

7.1.4.17 IMAGE

IMAGE describes a visual image (see 7.18 and Figure 15).



IEC 1521/06

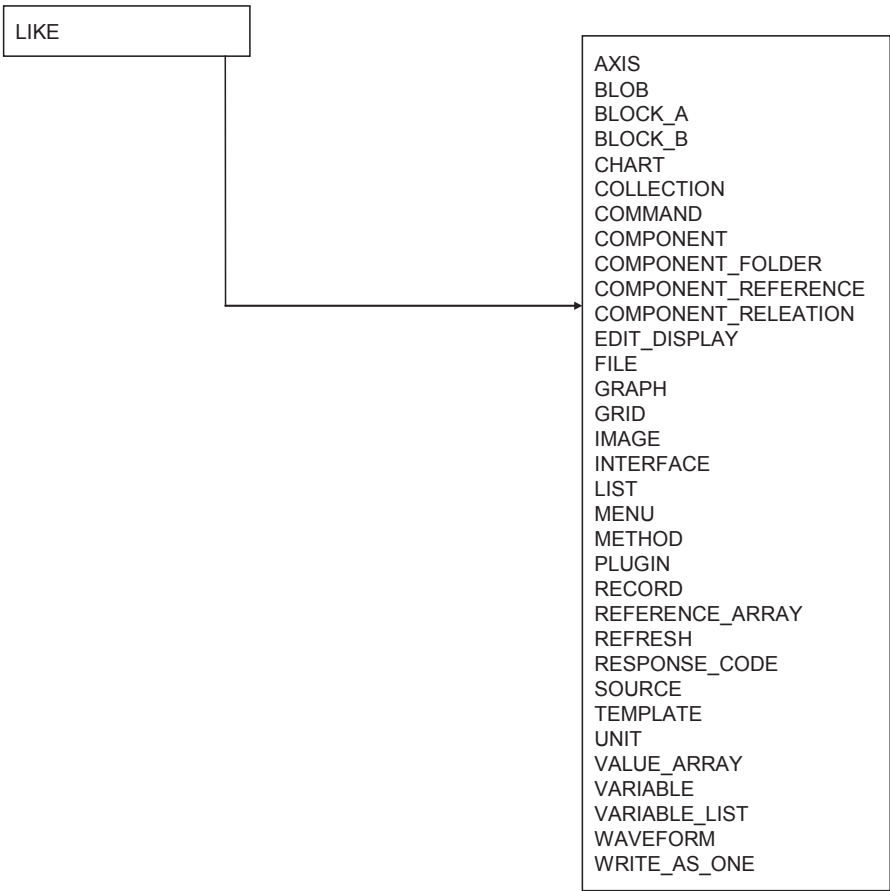
Figure 15 – IMAGE

7.1.4.18 IMPORT

IMPORT is used to import an EDD and to modify its existing definitions (see 7.19).

7.1.4.19 LIKE

LIKE creates a new instance of an existing instance of a basic construct. The attributes of the new instance may be redefined, added or deleted (see 7.21 and Figure 16).

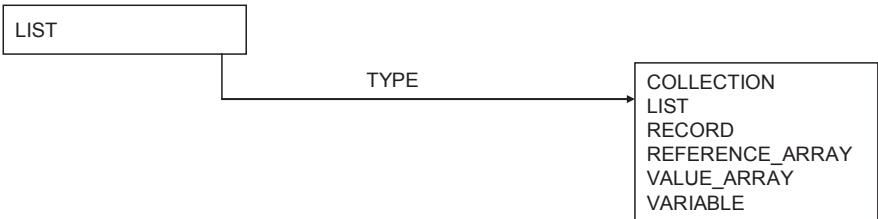


IEC 1522/06

Figure 16 – LIKE

7.1.4.20 LIST

LIST describes a sequence of elements (see 7.22 and Figure 17).



IEC 1523/06

Figure 17 – LIST

7.1.4.21 MENU

MENU describes how the data will be presented to a user by an EDD application (see 7.23 and Figure 18).

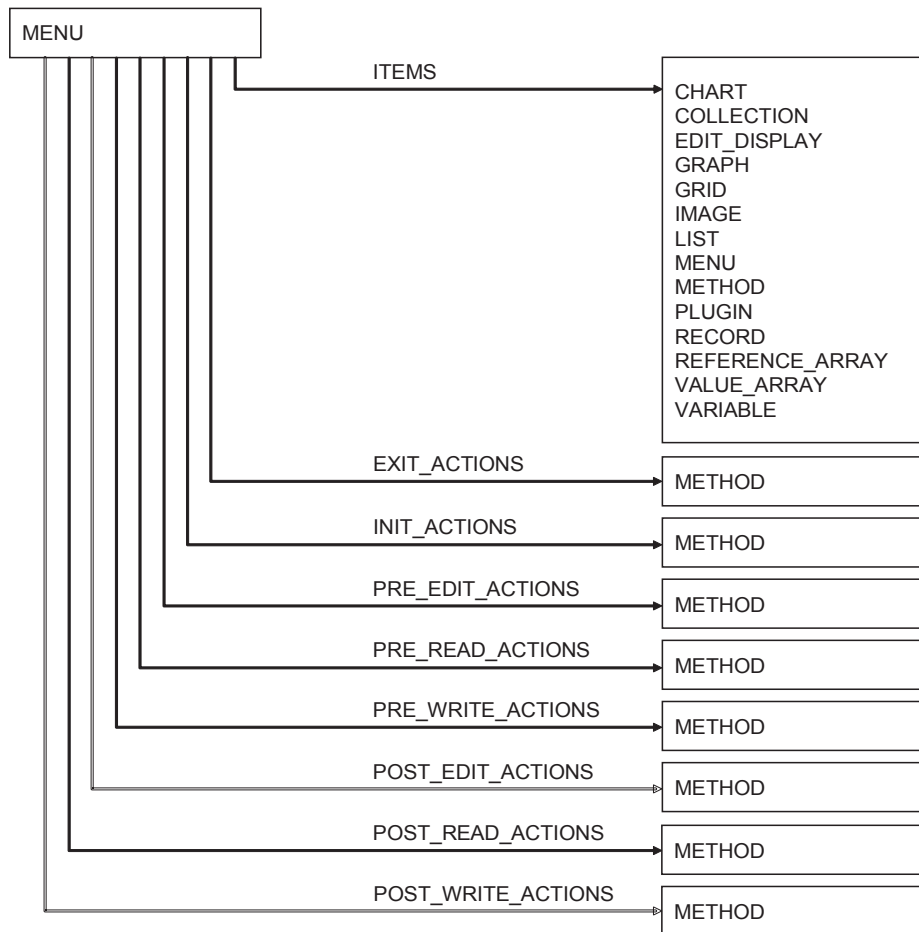


Figure 18 – MENU

IEC 1524/06

7.1.4.22 METHOD

METHOD describes the execution of complex sequence of event interactions that shall take place between system devices such as display units, configurators and field devices (see 7.24).

7.1.4.23 PLUGIN

PLUGIN describes a User Interface Plug-in (UIP) as specified in 7.42.

7.1.4.24 RECORD

RECORD describes the data contained in the device (see 7.26 and Figure 19).

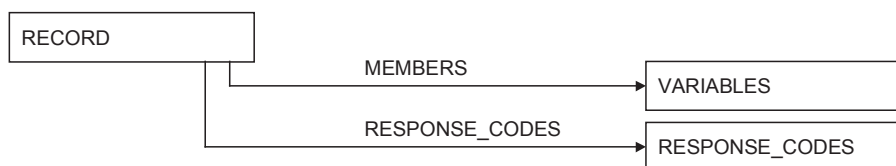
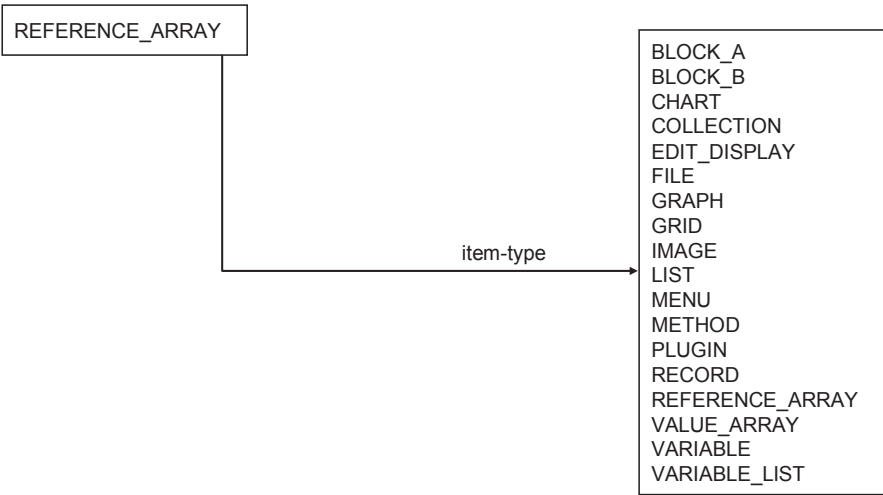


Figure 19 – RECORD

IEC 1526/06

911 7.1.4.25 REFERENCE_ARRAY

912 REFERENCE_ARRAY is a set of EDD items of the same EDDL item type (for example,
913 VARIABLE or MENU). An item can be referenced in the EDD via the REFERENCE_ARRAY
914 name and the index associated with the item (see 7.27 and Figure 20).



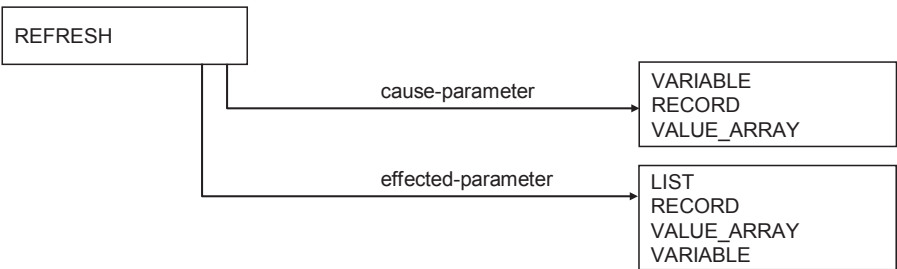
IEC 1527/06

Figure 20 – REFERENCE_ARRAY

917 7.1.4.26 Relations

918 7.1.4.26.1 REFRESH

919 REFRESH describes relationships between LISTs, RECORDs, VALUE_ARRAYs and
920 VARIABLES (see 7.28.1 and Figure 21).

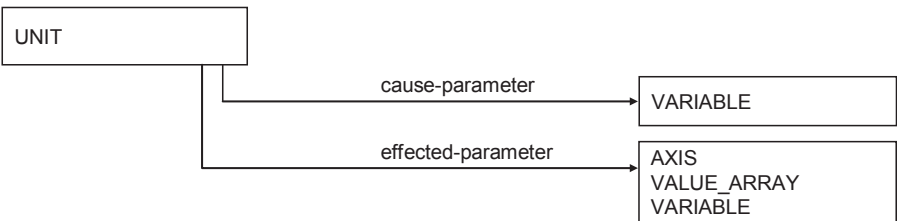


IEC 1528/06

Figure 21 – REFRESH

923 7.1.4.26.2 UNIT

924 UNIT describes relationships between AXISs, VALUE_ARRAYs and VARIABLES (see 7.28.2
925 and Figure 22).



IEC 1529/06

Figure 22 – UNIT

928 7.1.4.26.3 WRITE_AS_ONE

929 WRITE_AS_ONE describes relationships between VARIABLES, RECORDs, and
930 VALUE_ARRAYs (see 7.28.3 and Figure 23).



Figure 23 – WRITE_AS_ONE

IEC 1530/06

7.1.4.27 RESPONSE_CODES

RESPONSE_CODES specifies the list of response code elements that shall be used by the EDD application to inform the user about the communication result, e.g. error, warning (see 7.29).

7.1.4.28 SOURCE

SOURCE describes a source of data values for a CHART (see 7.30 and Figure 24).

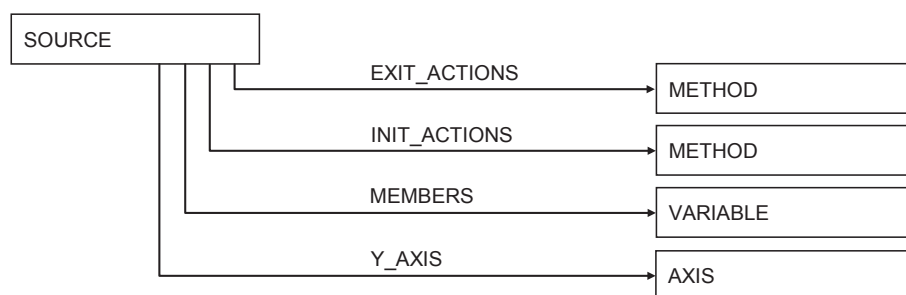


Figure 24 – SOURCE

IEC 1531/06

7.1.4.29 VALUE_ARRAY

VALUE_ARRAY describes logical groups of EDDL variables (see 7.32 and Figure 25).

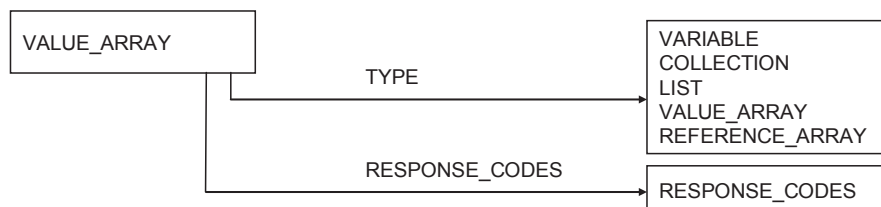
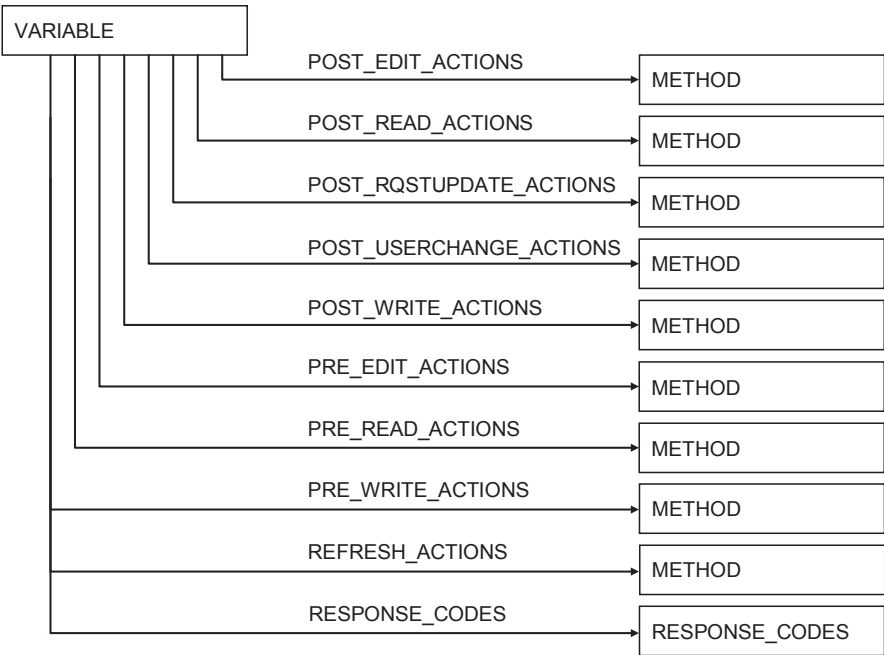


Figure 25 – VALUE_ARRAY

IEC 1532/06

7.1.4.30 VARIABLE

VARIABLE describes the data contained in the device (see 7.33 and Figure 26).

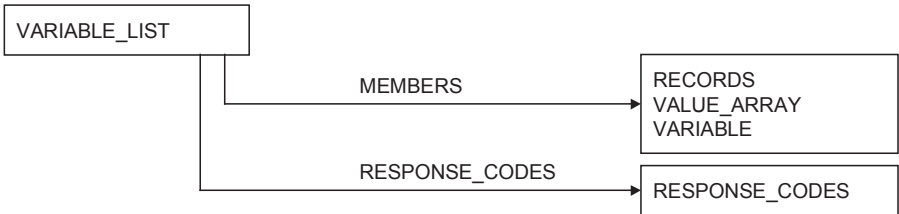


IEC 1533/06

Figure 26 – VARIABLE

7.1.4.31 VARIABLE_LIST

VARIABLE_LIST describes logical groupings of data contained in the device that may be communicated as a list (see 7.34 and Figure 27).

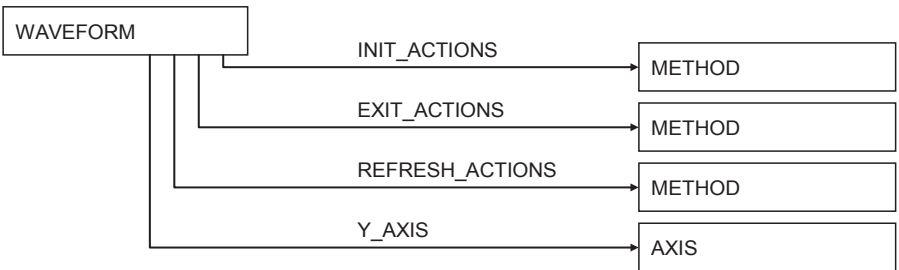


IEC 1534/06

Figure 27 – VARIABLE_LIST

7.1.4.32 WAVEFORM

WAVEFORM describes a data set that may be displayed by a GRAPH (see 7.35 and Figure 28).



IEC 1535/06

Figure 28 – WAVEFORM

7.1.5 Common attributes

The following common attributes are used in many basic constructs:

- CLASSIFICATION
- COMPONENT_PARENT

- 963 • COMPONENT_PATH
- 964 • DEFINITION
- 965 • EMPHASIS
- 966 • HANDLING
- 967 • HEIGHT
- 968 • HELP
- 969 • IDENTITY
- 970 • LABEL
- 971 • LINE_COLOR
- 972 • LINE_TYPE
- 973 • MEMBERS
- 974 • PRIVATE
- 975 • PROTOCOL
- 976 • RESPONSE_CODES
- 977 • SUPPLIED_INTERFACE
- 978 • VALIDITY
- 979 • VISIBILITY
- 980 • WIDTH
- 981 • WRITE_MODE

982 7.1.6 Special elements

983 The special elements are additional EDDL mechanisms to support file handling, multiple
984 instances and modification of EDDL basic elements, such as:

- 985 • **Conditional Expressions** specify attributes values, which are dependent on the values of
986 variables at run-time
- 987 • **Reference** is used throughout a device description to refer to other items within the same
988 EDDL
- 989 • **Expression** is a logical or mathematical operation on variable values in order to modify
990 attribute values of a basic construct at run-time

991 7.1.7 Rules for instances

992 Each instance of a basic construct shall be identified by an identifier of type string. This
993 identifier shall be unique within the complete EDD. All EDDL constructs represent actual
994 instances of device information. If a device contains two or more pieces of information which
995 have the same structure the appropriate construct shall be repeated in the EDD with different
996 identifiers. Alternatively the LIKE construct may be used to create a copy of the construct with
997 a different identifier. The identifier in the lexical definition part is not described within each
998 construct again.

999 7.1.8 Rules for list of VARIABLES

1000 A list of VARIABLES may contain VARIABLES, elements of composed types or composed
1001 types. Composed types are:

- 1002 • BLOCK_A PARAMETERS
- 1003 • COLLECTIONS
- 1004 • RECORDS
- 1005 • REFERENCE_ARRAY
- 1006 • VALUE_ARRAY

- 1007 • VARIABLE_LISTS

1008 Instead of referencing every single member of a composed type, a reference can be made to
1009 the type instance.

1010 7.2 EDD identification information

1011 7.2.1 General structure

1012 Purpose

1013 EDD identification information uniquely identifies the device description of a specific device
1014 type from a device manufacturer. It also specifies the EDDL version and EDDL profile used.

1015 This identification information shall be the first entry in every EDD and shall appear only once.

1016 Lexical structure

1017 DD_REVISION, DEVICE_REVISION, DEVICE_TYPE, EDD_PROFILE, EDD_VERSION,
1018 MANUFACTURER, MANUFACTURER_EXT

1019 NOTE 1 EDD identification information is not structured in the same way as other EDDL constructs.

1020 NOTE 2 Strict order of the keywords of the EDD identification information is not necessary.

1021 7.2.2 Specific attributes

1022 7.2.2.1 DD_REVISION

1023 Purpose

1024 The DD_REVISION attribute specifies a unique code for the EDD revision of this device, as
1025 defined by the manufacturer.

1026 The manufacturer should change the revision number each time a new device description is
1027 issued for a device.

1028 There can be multiple EDDs for a particular revision of a particular device, in which case the
1029 DD_REVISION distinguishes them.

1030 Lexical structure

1031 DD_REVISION integer

1032 The attribute of DD_REVISION is specified in Table 2.

1033 Table 2 – DD_REVISION attribute

Usage	Attribute	Description
m	integer	A two-octet unsigned integer containing the code for a specific EDD file revision.

1034

1035 7.2.2.2 DEVICE_REVISION

1036 Purpose

1037 The DEVICE_REVISION attribute specifies a unique code for the device revision of the field
1038 device, as defined by the manufacturer.

1039 Lexical structure

1040 DEVICE_REVISION integer

1041 The attribute of DEVICE_REVISION is specified in Table 3.

1042 Table 3 – DEVICE_REVISION attribute

Usage	Attribute	Description
m	integer	A two-octet unsigned integer containing the code for a specific device revision.

1043

7.2.2.3 DEVICE_TYPE**Purpose**

The DEVICE_TYPE attribute specifies an identifier for the device type, as defined by the manufacturer of the device. The identifier should be unique for each type.

The device type may specify either a category of devices or a specific product.

Lexical structure

DEVICE_TYPE (integer, identifier)

The attribute of DEVICE_TYPE is specified in Table 4.

Table 4 – DEVICE_TYPE attribute

Usage	Attribute	Description
s	integer	A two-octet unsigned integer containing the code for a specific device type or group of device types.
s	identifier	A reference to a two-octet integer which contains the code for a specific device type or group of device types.
The mapping of the identifier to an integer constant may be stored in an extra file named "devices.cfg". This file contains a list of identifiers for devices or groups of device types.		

7.2.2.4 EDD_PROFILE**Purpose**

The EDD_PROFILE attribute specifies the selected profile of the EDDL as defined in Annex D.

Lexical structure

EDD_PROFILE integer

The attribute of EDD_PROFILE is specified in Table 5.

Table 5 – EDD_PROFILE attribute

Usage	Attribute	Description
o	integer	A two-octet unsigned integer containing the code for the specific profiles selected from Annex D. The following codes are currently defined: 0x01 Profile for HART Communication Foundation (see D.4) 0x02 Profile for Fieldbus Foundation (see D.3) 0x03 Profile for PROFIBUS (see D.2) 0x04 Profile for PROFINET (see D.2)

7.2.2.5 EDD_VERSION**Purpose**

The EDD_VERSION attribute specifies a unique code for the EDD version, which has been used by the manufacturer in constructing the EDD. This attribute shall be included with all device descriptions conforming to this standard.

Lexical structure

EDD_VERSION integer

The attribute of EDD_VERSION is specified in Table 6.

Table 6 – EDD_VERSION attribute

Usage	Attribute	Description
o	integer	A two-octet unsigned integer containing the number 1 for this first edition of the standard.

1072 **7.2.2.6 MANUFACTURER**1073 **Purpose**

1074 The MANUFACTURER attribute identifies the manufacturer. The code allocation should be
 1075 managed by the specific organizations which are responsible for the different EDDL profiles.
 1076 The combination of EDD_PROFILE and MANUFACTURER shall be unique.

1077 **Lexical structure**

1078 MANUFACTURER (long integer, identifier)

1079 The attribute of MANUFACTURER is specified in Table 7.

1080 **Table 7 – MANUFACTURER attribute**

Usage	Attribute	Description
s	long integer	An unsigned long integer containing the code for a specific manufacturer. If all bits in the value of the integer are set to one, the lexical structure "MANUFACTURER_EXT" shall be present and identifies the manufacturer.
sm	identifier	A reference to an unsigned long integer which contains the code for a specific manufacturer.
The mapping of the identifier to an integer constant should be stored in an extra file named "devices.cfg". This file contains a list of identifiers for manufacturers.		

1081

1082 **7.2.2.7 MANUFACTURER_EXT**1083 **Purpose**

1084 The MANUFACTURER_EXT attribute is reserved for future use. It is a placeholder for a string
 1085 providing global identification for the MANUFACTURER.

1086 NOTE The specification of this attribute will be included when it becomes internationally available.

1087 **Lexical structure**

1088 MANUFACTURER_EXT string

1089 The attribute of MANUFACTURER_EXT is specified in Table 8.

1090 **Table 8 – MANUFACTURER_EXT attribute**

Usage	Attribute	Description
c	string	A string containing the information for a unique worldwide manufacturer identification.

1091

1092 **7.3 AXIS**1093 **7.3.1 General structure**1094 **Purpose**

1095 AXIS describes an axis of a CHART or GRAPH.

1096 **Lexical structure**

1097 AXIS identifier [CONSTANT_UNIT , HELP, LABEL, MAX_VALUE, MIN_VALUE,
 1098 SCALING]

1099 The attributes of AXIS are specified in Table 9.

1100 **Table 9 – AXIS attributes**

Usage	Attribute	Description
o	CONSTANT_UNIT	Lexical element (see 7.3.2.1).
o	HELP	Lexical element (see 7.36.8).

Usage	Attribute	Description
o	LABEL	Lexical element (see 7.36.9).
o	MAX_VALUE	Lexical element (see 7.3.2.1).
o	MIN_VALUE	Lexical element (see 7.3.2.3).
o	SCALING	Lexical element (see 7.3.2.4).

1101

1102 **7.3.2 Specific attributes**1103 **7.3.2.1 CONSTANT_UNIT**1104 **Purpose**

1105 The CONSTANT_UNIT attribute is used, if an AXIS has a unit code which never changes. The
 1106 CONSTANT_UNIT is specified as a string that will be displayed along with the value. An AXIS
 1107 without a constant unit either has no units associated with it or the units are not constant.

1108 **Lexical structure**

1109 `CONSTANT_UNIT (string)<exp>`

1110 The attribute of CONSTANT_UNIT is specified in Table 175.

1111 **7.3.2.2 MAX_VALUE**1112 **Purpose**

1113 The MAX_VALUE attribute specifies the value associated with the last tick mark of the AXIS.
 1114 The last tick mark of an X_AXIS and Y_AXIS is the right-most and top-most tick mark,
 1115 respectively. By default, the value associated with the last tick mark of an AXIS without a
 1116 MAX_VALUE attribute is inferred from the WAVEFORMS on the GRAPH.

1117 **Lexical structure**

1118 `MAX_VALUE (reference, double, float, integer, unsigned_integer)<exp>`

1119 The attributes of MAX_VALUE are specified in Table 10.

1120

Table 10 – MAX_VALUE, MIN_VALUE attribute

Usage	Attribute	Description
o	reference	A reference to either a VARIABLE instance, a member of a RECORD instance, an element of a VALUE_ARRAY instance or an element of a REFERENCE_ARRAY instance with a numeric or a date/time data type. MAX_VALUE and MIN_VALUE shall reference VARIABLES of the same data type. This shall apply for X_AXIS and Y_AXIS.
o	double	A double-precision floating-point constant.
o	float	A single-precision floating-point constant.
o	integer	An integer constant.
o	unsigned_integer	An unsigned integer constant.

1121

1122 **7.3.2.3 MIN_VALUE**1123 **Purpose**

1124 The MIN_VALUE attribute specifies the value associated with the first tick mark of the AXIS.
 1125 The first tick mark of an X_AXIS and Y_AXIS is the left-most and bottom-most tick mark,
 1126 respectively. By default, the value associated with the first tick mark of an AXIS without a
 1127 MIN_VALUE attribute is inferred from the WAVEFORMS on the GRAPH.

1128 **Lexical structure**

1129 `MIN_VALUE (reference, double, float, integer, unsigned_integer)<exp>`

1130 The attributes of MIN_VALUE are specified in Table 10.

1131 **7.3.2.4 SCALING**1132 **Purpose**

1133 The SCALING attribute specifies the way the tick marks of the AXIS should be scaled. By
 1134 default, an AXIS without a SCALING attribute is scaled linearly.

1135 **Lexical structure**

1136 SCALING (LINEAR, LOGARITHMIC) <exp>

1137 The attributes of SCALING are specified in Table 11.

1138 **Table 11 – SCALING attribute**

Usage	Attribute	Description
s	LINEAR	The tick marks on the axis should be scaled linearly.
s	LOGARITHMIC	The tick marks on the axis should scaled logarithmically (base 10).

1139

1140 **7.4 BLOCK**1141 **7.4.1 BLOCK_A**1142 **7.4.1.1 General structure**1143 **Purpose**

1144 A device may be organized in logical blocks. Each BLOCK_A is a logical grouping of
 1145 PARAMETERS and additional information for BLOCK_A applications. Types of blocks are
 1146 described by the CHARACTERISTICS attribute. More than one BLOCK_A from the same type
 1147 can be instantiated.

1148 NOTE This BLOCK_A approach is optimized for access efficiency.

1149 **Lexical structure**

1150 BLOCK_A identifier [CHARACTERISTICS, LABEL, PARAMETERS, AXIS_ITEMS,
 1151 CHART_ITEMS, COLLECTION_ITEMS, EDIT_DISPLAY_ITEMS, FILE_ITEMS, GRAPH_ITEMS,
 1152 GRID_ITEMS, HELP, IMAGE_ITEMS, LIST_ITEMS, MENU_ITEMS, METHOD_ITEMS,
 1153 PARAMETER_LISTS, PLUGIN_ITEMS, REFERENCE_ARRAY_ITEMS, REFRESH_ITEMS,
 1154 SOURCE_ITEMS, UNIT_ITEMS, WAVEFORM_ITEMS, WRITE_AS_ONE_ITEMS, CHARTS,
 1155 FILES, LISTS, GRAPHS, GRIDS, MENUS, METHODS, PLUGINS]

1156 The attributes of BLOCK_A are specified in Table 12.

1157 **Table 12 – BLOCK_A attributes**

Usage	Attribute	Description
m	CHARACTERISTICS	Lexical element (see 7.4.1.2.1).
m	LABEL	Lexical element (see 7.36.9).
m	PARAMETERS	Lexical element (see 7.4.1.2.2).
o	AXIS_ITEMS	Lexical element (see 7.4.1.2.3).
o	CHART_ITEMS	Lexical element (see 7.4.1.2.4).
o	COLLECTION_ITEMS	Lexical element (see 7.4.1.2.5).
o	EDIT_DISPLAY_ITEMS	Lexical element (see 7.4.1.2.6).
o	FILE_ITEMS	Lexical element (see 7.4.1.2.7).
o	GRAPH_ITEMS	Lexical element (see 7.4.1.2.8).
o	GRID_ITEMS	Lexical element (see 7.4.1.2.9).
o	HELP	Lexical element (see 7.36.8).
o	IMAGE_ITEMS	Lexical element (see 7.4.1.2.10).
o	LIST_ITEMS	Lexical element (see 7.4.1.2.11).
o	LOCAL_PARAMETERS	Lexical element (see 7.4.1.2.12).

Usage	Attribute	Description
o	MENU_ITEMS	Lexical element (see 7.4.1.2.13).
o	METHOD_ITEMS	Lexical element (see 7.4.1.2.14).
o	PARAMETER_LISTS	Lexical element (see 7.4.1.2.15).
o	PLUGIN_ITEMS	Lexical element (see 7.4.1.2.29).
o	REFERENCE_ARRAY_ITEMS	Lexical element (see 7.4.1.2.16).
o	REFRESH_ITEMS	Lexical element (see 7.4.1.2.17).
o	SOURCE_ITEMS	Lexical element (see 7.4.1.2.18).
o	UNIT_ITEMS	Lexical element (see 7.4.1.2.19).
o	WAVEFORM_ITEMS	Lexical element (see 7.4.1.2.20).
o	WRITE_AS_ONE_ITEMS	Lexical element (see 7.4.1.2.21).
o	CHARTS	Lexical element (see 7.4.1.2.22).
o	FILES	Lexical element (see 7.4.1.2.28).
o	LISTS	Lexical element (see 7.4.1.2.23).
o	GRAPHS	Lexical element (see 7.4.1.2.24).
o	GRIDS	Lexical element (see 7.4.1.2.25).
o	MENUS	Lexical element (see 7.4.1.2.26).
o	METHODS	Lexical element (see 7.4.1.2.27).
o	PLUGINS	Lexical element (see 7.4.1.2.30).

1158

1159 **7.4.1.2 Specific attributes**1160 **7.4.1.2.1 CHARACTERISTICS**1161 **Purpose**

1162 The CHARACTERISTICS attribute specifies the external characteristics for a BLOCK_A.
 1163 CHARACTERISTICS may include class, function type, block type and execution time. The
 1164 external characteristics of a BLOCK_A are contained in a RECORD.

1165 NOTE In general the BLOCK_A CHARACTERISTICS fix the type of the BLOCK_A.

1166 **Lexical structure**

1167 CHARACTERISTICS reference

1168 The attribute of CHARACTERISTICS is specified in Table 13.

1169 **Table 13 – CHARACTERISTIC attribute**

Usage	Attribute	Description
m	reference	A reference to a RECORD instance.

1170

1171 **7.4.1.2.2 PARAMETERS**1172 **Purpose**

1173 The PARAMETERS attribute contains a list of references to VARIABLE, VALUE_ARRAY or
 1174 RECORD instances. Each reference shall have an identifier and may have a description
 1175 and/or a help text.

1176 NOTE The PARAMETERS in a BLOCK_A correspond to individual communication objects, listed in a device
 1177 object dictionary. The PARAMETERS elements reference the communication definitions of these objects. The
 1178 object dictionary maps into the specific location for data in a device.

1179 **Lexical structure**

1180 PARAMETERS [identifier, reference, description, help] +

1181 The attributes of PARAMETERS are specified in Table 14.

1182

Table 14 – PARAMETER attributes

Usage	Attribute	Description
m	identifier	The identifier of the VARIABLE, VALUE_ARRAY or RECORD. Every element of the PARAMETERS list shall have an identifier to be used in the EDD to refer to it.
m	reference	A reference to a VARIABLE, VALUE_ARRAY or RECORD instance.
o	description	A short description for the reference.
o	help	Help text for the reference.

1183

1184 7.4.1.2.3 AXIS_ITEMS**1185 Purpose**

1186 The AXIS_ITEMS attribute is a list of the AXIS instances associated with the BLOCK_A.

1187 Lexical structure

1188 `AXIS_ITEMS reference+`

1189 The attribute of AXIS_ITEMS is specified in Table 15.

1190

Table 15 – AXIS_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a AXIS instance.

1191

1192 7.4.1.2.4 CHART_ITEMS**1193 Purpose**

1194 The CHART_ITEMS attribute is a list of the CHART instances associated with the BLOCK_A.

1195 Lexical structure

1196 `CHART_ITEMS reference+`

1197 The attribute of CHART_ITEMS is specified in Table 16.

1198

Table 16 – CHART_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a CHART instance.

1199

1200 7.4.1.2.5 COLLECTION_ITEMS**1201 Purpose**

1202 The COLLECTION_ITEMS attribute is a list of the COLLECTION instances associated with
1203 the BLOCK_A.

1204 Lexical structure

1205 `COLLECTION_ITEMS reference+`

1206 The attribute of COLLECTION_ITEMS is specified in Table 17.

1207

Table 17 – COLLECTION_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a COLLECTION instance.

1208

1209 7.4.1.2.6 EDIT_DISPLAY_ITEMS**1210 Purpose**

1211 The EDIT_DISPLAY_ITEMS attribute is a list of the EDIT_DISPLAY instances associated with
1212 BLOCK_A.

1213 Lexical structure

1214 EDIT_DISPLAY_ITEMS reference+

1215 The attribute of EDIT_DISPLAY_ITEMS is specified in Table 18.

1216 Table 18 – EDIT_DISPLAY_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to an EDIT_DISPLAY instance.

1217

1218 7.4.1.2.7 FILE_ITEMS**1219 Purpose**

1220 The FILE_ITEMS attribute is a list of the FILE instances associated with the BLOCK_A.

1221 Lexical structure

1222 FILE_ITEMS reference+

1223 The attribute of FILE_ITEMS is specified in Table 19.

1224 Table 19 – FILE_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a FILE instance.

1225

1226 7.4.1.2.8 GRAPH_ITEMS**1227 Purpose**

1228 The GRAPH_ITEMS attribute is a list of the GRAPH instances associated with the BLOCK_A.

1229 Lexical structure

1230 GRAPH_ITEMS reference+

1231 The attribute of GRAPH_ITEMS is specified in Table 20.

1232 Table 20 – GRAPH_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a GRAPH instance.

1233

1234 7.4.1.2.9 GRID_ITEMS**1235 Purpose**

1236 The GRID_ITEMS attribute is a list of the GRID instances associated with the BLOCK_A.

1237 Lexical structure

1238 GRID_ITEMS reference+

1239 The attribute of GRID_ITEMS is specified in Table 21.

1240 Table 21 – GRID_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a GRID instance.

1241

1242 **7.4.1.2.10 IMAGE_ITEMS**1243 **Purpose**

1244 The IMAGE_ITEMS attribute is a list of the IMAGE instances associated with BLOCK_A.

1245 **Lexical structure**

1246 IMAGE_ITEMS reference+

1247 The attribute of IMAGE_ITEMS is specified in Table 22.

1248

Table 22 – IMAGE_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a IMAGE instance.

1249

1250 **7.4.1.2.11 LIST_ITEMS**1251 **Purpose**

1252 The LIST_ITEMS attribute is a list of the LIST instances associated with BLOCK_A.

1253 **Lexical structure**

1254 LIST_ITEMS reference+

1255 The attribute of LIST_ITEMS is specified in Table 23.

1256

Table 23 – LIST_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a LIST instance.

1257

1258 **7.4.1.2.12 LOCAL_PARAMETERS**1259 **Purpose**

1260 The LOCAL_PARAMETERS attribute contains a list of references to VARIABLE,
1261 VALUE_ARRAY or RECORD instances. Each reference shall have an identifier and may have
1262 a description and/or a help text.

1263 NOTE The LOCAL_PARAMETERS in a BLOCK_A do not correspond to communication objects; they are
1264 parameters that are only used by the EDD application.

1265 **Lexical structure**

1266 LOCAL_PARAMETERS [identifier, reference, description, help]+

1267 The attributes of LOCAL_PARAMETERS are specified in Table 14.

1268 **7.4.1.2.13 MENU_ITEMS**1269 **Purpose**

1270 The MENU_ITEMS attribute specifies the MENUs associated with BLOCK_A.

1271 **Lexical structure**

1272 MENU_ITEMS reference+

1273 The attribute of MENU_ITEMS is specified in Table 24.

1274

Table 24 – MENU_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a MENU.

1275

7.4.1.2.14 METHOD_ITEMS**Purpose**

The METHOD_ITEMS attribute specifies the METHODS associated with BLOCK_A.

Lexical structure

METHOD_ITEMS reference+

The attribute of METHOD_ITEMS is specified in Table 25.

Table 25 – METHOD_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a METHOD instance.

7.4.1.2.15 PARAMETER_LISTS**Purpose**

The PARAMETER_LISTS attribute specify the VARIABLE_LISTs associated with a BLOCK_A, plus an optional description and help for each VARIABLE_LIST. Every BLOCK_A may contain multiple PARAMETER_LISTS.

Lexical structure

PARAMETER_LISTS [identifier, reference, description, help]+

The attributes of PARAMETER_LISTS are specified in Table 26.

Table 26 – PARAMETER_LISTS attributes

Usage	Attribute	Description
m	identifier	The identifier of the VARIABLE_LIST. Every element of the PARAMETER_LIST list shall have an identifier to be used in the EDD to refer to it.
m	reference	A reference to a VARIABLE_LIST instance.
o	description	A short description of the item.
o	help	Help text for the item.

7.4.1.2.16 REFERENCE_ARRAY_ITEMS**Purpose**

The REFERENCE_ARRAY_ITEMS attribute specifies the REFERENCE_ARRAYs associated with the BLOCK_A.

Lexical structure

REFERENCE_ARRAY_ITEMS reference+

The attribute of REFERENCE_ARRAY_ITEMS is specified in Table 27.

Table 27 – REFERENCE_ARRAY_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a REFERENCE_ARRAY instance.

7.4.1.2.17 REFRESH_ITEMS**Purpose**

The REFRESH_ITEMS attribute specify the REFRESH relations associated with the BLOCK_A.

1308 Lexical structure**1309** REFRESH_ITEMS reference+**1310** The attribute of REFRESH_ITEMS is specified in Table 28.**1311 Table 28 – REFRESH_ITEMS attribute**

Usage	Attribute	Description
m	reference	A reference to a REFRESH instance.

1312**1313 7.4.1.2.18 SOURCE_ITEMS****1314 Purpose****1315** The SOURCE_ITEMS attribute specifies the SOURCES associated with the BLOCK_A.**1316 Lexical structure****1317** SOURCE_ITEMS reference+**1318** The attribute of SOURCE_ITEMS is specified in Table 29.**1319 Table 29 – SOURCE_ITEMS attribute**

Usage	Attribute	Description
m	reference	A reference to a SOURCE instance.

1320**1321 7.4.1.2.19 UNIT_ITEMS****1322 Purpose****1323** The UNIT_ITEMS attribute specify the UNIT relations associated with the BLOCK_A.**1324 Lexical structure****1325** UNIT_ITEMS reference+**1326** The attribute of UNIT_ITEMS is specified in Table 30.**1327 Table 30 – UNIT_ITEMS attribute**

Usage	Attribute	Description
m	reference	A reference to an UNIT instance.

1328**1329 7.4.1.2.20 WAVEFORM_ITEMS****1330 Purpose****1331** The WAVEFORM_ITEMS attribute specifies the WAVEFORMs associated with the BLOCK_A.**1332 Lexical structure****1333** WAVEFORM_ITEMS reference+**1334** The attribute of WAVEFORM_ITEMS is specified in Table 31.**1335 Table 31 – WAVEFORM_ITEMS attribute**

Usage	Attribute	Description
m	reference	A reference to a WAVEFORM instance.

1336

7.4.1.2.21 WRITE_AS_ONE_ITEMS**Purpose**

The WRITE_AS_ONE_ITEMS attribute specifies the WRITE_AS_ONE relations associated with the BLOCK_A.

Lexical structure

WRITE_AS_ONE_ITEMS reference+

The attribute of WRITE_AS_ONE_ITEMS is specified in Table 32.

Table 32 – WRITE_AS_ONE_ITEMS attribute

Usage	Attribute	Description
m	reference	A reference to a WRITE_AS_ONE instance.

7.4.1.2.22 CHARTS**Purpose**

The CHARTS attribute specifies the CHARTs associated with the BLOCK_A. Only those CHARTs that will be referenced via a specific BLOCK_A instance (see 7.38.21) shall be specified.

Lexical structure

CHARTS (identifier, reference, description, help)+

The attribute of CHARTS is specified in Table 33.

Table 33 – CHARTS attribute

Usage	Attribute	Description
m	identifier	The name by which the CHART may be referenced.
m	reference	A reference to a CHART instance.
o	description	A short description of the CHART.
o	help	Help text for the CHART.

7.4.1.2.23 LISTS**Purpose**

The LISTS attribute specifies the LISTS associated with the BLOCK_A. Only those LISTS that will be referenced via a specific BLOCK_A instance (see 7.4.1) shall be specified.

Lexical structure

LISTS (identifier, reference, description, help)+

The attribute of LISTS is specified in Table 34.

Table 34 – LISTS attribute

Usage	Attribute	Description
m	identifier	The name by which the LIST may be referenced.
m	reference	A reference to a LIST instance.
o	description	A short description of the LIST.
o	help	help text for the LIST.

1365 **7.4.1.2.24 GRAPHS**1366 **Purpose**

1367 The GRAPHS attribute specifies the GRAPHS associated with the BLOCK_A. Only those
 1368 GRAPHS that will be referenced via a specific BLOCK_A instance (see 7.38.23) shall be
 1369 specified.

1370 **Lexical structure**

1371 GRAPHS (identifier, reference, description, help)+

1372 The attribute of GRAPHS is specified in Table 35.

1373 **Table 35 – GRAPHS attribute**

Usage	Attribute	Description
m	identifier	The name by which the GRAPH may be referenced.
m	reference	A reference to a GRAPH instance.
o	description	A short description of the GRAPH.
o	help	Help text for the GRAPH.

1374

1375 **7.4.1.2.25 GRIDS**1376 **Purpose**

1377 The GRIDS attribute specifies the GRIDS associated with the BLOCK_A. Only those GRIDS
 1378 that will be referenced via a specific BLOCK_A instance (see 7.38.24) shall be specified.

1379 **Lexical structure**

1380 GRIDS (identifier, reference, description, help)+

1381 The attribute of GRIDS is specified in Table 36.

1382 **Table 36 – GRIDS attribute**

Usage	Attribute	Description
m	identifier	The name by which the GRID may be referenced.
m	reference	A reference to a GRID instance.
o	description	A short description of the GRID.
o	help	Help text for the GRID.

1383

1384 **7.4.1.2.26 MENUS**1385 **Purpose**

1386 The MENUS attribute specifies the MENUS associated with the BLOCK_A. Only those MENUS
 1387 that will be referenced via a specific BLOCK_A instance (see 7.38.25) shall be specified.

1388 **Lexical structure**

1389 MENUS (identifier, reference, description, help)+

1390 The attribute of MENUS is specified in Table 37.

1391 **Table 37 – MENUS attribute**

Usage	Attribute	Description
m	identifier	The name by which the MENU may be referenced.
m	reference	A reference to a MENU instance.
o	description	A short description of the MENU.
o	help	Help text for the MENU.