

ANSI/ISA—61804-3 (104.00.01)-2007

**Function Blocks (FB) For
Process Control – Part 3:
Electronic Device Description
Language (EDDL)**

Approved 30 March 2007

ANSI/ISA-61804-3 (104.00.01)-2007

Function Blocks (FB) for Process Control – Part 3: Electronic Device Description Language

ISBN: 978-1-934394-17-5

Copyright © 2007 by IEC and ISA. All rights reserved. Not for resale. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), without the prior written permission of the Publisher.

ISA
67 Alexander Drive
P. O. Box 12277
Research Triangle Park, North Carolina 27709
USA

Preface

This preface, as well as all footnotes and annexes, is included for information purposes and is not part of ANSI/ISA-61804-3 (104.00.01)-2007.

The standards referenced within this document may contain provisions which, through reference in this text, constitute requirements of this document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated within this document. Members of IEC and ISO maintain registers of currently valid International Standards. ANSI maintains registers of currently valid U.S. National Standards.

This document has been prepared as part of the service of ISA—The Instrumentation, Systems, and Automation Society, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards, recommended practices, and technical reports to the greatest extent possible. *Standard for Use of the International System of Units (SI): The Modern Metric System*, published by the American Society for Testing & Materials as IEEE/ASTM SI 10-97, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards, recommended practices, and technical reports. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

CAUTION — ISA ADHERES TO THE POLICY OF THE AMERICAN NATIONAL STANDARDS INSTITUTE WITH REGARD TO PATENTS. IF ISA IS INFORMED OF AN EXISTING PATENT THAT IS REQUIRED FOR USE OF THE STANDARD, IT WILL REQUIRE THE OWNER OF THE PATENT TO EITHER GRANT A ROYALTY-FREE LICENSE FOR USE OF THE PATENT BY USERS COMPLYING WITH THE STANDARD OR A LICENSE ON REASONABLE TERMS AND CONDITIONS THAT ARE FREE FROM UNFAIR DISCRIMINATION.

EVEN IF ISA IS UNAWARE OF ANY PATENT COVERING THIS STANDARD, THE USER IS CAUTIONED THAT IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS THAT MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA IS NOT RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE

RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER'S INTENDED APPLICATION.

HOWEVER, ISA ASKS THAT ANYONE REVIEWING THIS STANDARD WHO IS AWARE OF ANY PATENTS THAT MAY IMPACT IMPLEMENTATION OF THE STANDARD NOTIFY THE ISA STANDARDS AND PRACTICES DEPARTMENT OF THE PATENT AND ITS OWNER.

ADDITIONALLY, THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER'S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

The following people served as members of ISA SP104:

NAME	COMPANY
T. Blevins*, Chair	Emerson Process
R. Webb, Managing Director	Consultant
J. Berge*	Emerson Process
R. Caro	CMC Associates
C. Diedrich	University of Magdeburg
G. Hood	Rockwell Automation
E. Ladd	Fieldbus Inc
H. Renjie	Southwest University
C. Verney	Schneider Electric
I. Weber*	Siemens
L. Winkel*	Siemens

* One vote per company.

This standard was approved for publication by the ISA Standards and Practices Board on

22 March 2007.

NAME	AFFILIATION
T. McAviney, Vice President	Jacobs Engineering Group
M. Coppler	Ametek Inc.
B. Dumortier	Schneider Electric
D. Dunn	Aramco Services Co.
W. Holland	Consultant
E. Icayan	ACES Inc.
J. Jamison	Husky Energy Inc.
R. Jones	CDI Business Solutions
K. P. Lindner	Endress + Hauser Process Solutions
V. Maggioli	Feltronics Corp.
A. McCauley	Chagrin Valley Controls Inc.

G. McFarland
R. Reimer
N. Sands
H. Sasajima
T. Schnaare
J. Tatera
I. Verhappen
R. Webb
W. Weidman
J. Weiss
M. Widmeyer
M. Zielinski

Emerson Process Management
Rockwell Automation
E I du Pont
Yamatake Corp.
Rosemount Inc.
Tatera & Associates
MTL Instrument Group
Consultant
Worley Parsons
KEMA Inc.
Consultant
Emerson Process Management

This page intentionally left blank.

CONTENTS

1	Scope	25
2	Normative references	25
3	Terms, definitions, abbreviated terms and acronyms	26
3.1	Terms and definitions	26
3.53	Abbreviated terms and acronyms	33
4	Conformance statement.....	34
5	Conventions for lexical structures	34
6	EDD and EDDL model	36
6.1	Overview of EDD and EDDL.....	36
6.2	EDD architecture	36
6.3	Concepts of EDD	36
6.4	Principles of the EDD development process	37
6.5	Interrelations between the lexical structure and formal definitions	38
6.6	Builtins	38
6.7	Profiles.....	38
7	Electronic Device Description Language	39
7.1	Overview	39
7.2	EDD identification information	50
7.3	AXIS	54
7.4	BLOCK.....	56
7.5	CHART.....	66
7.6	COLLECTION.....	68
7.7	COMMAND.....	70
7.8	CONNECTION.....	76
7.9	DOMAIN.....	77
7.10	EDIT_DISPLAY	78
7.11	FILE	80
7.12	GRAPH	81
7.13	GRID.....	84
7.14	IMAGE	88
7.15	IMPORT	89
7.16	LIKE.....	105
7.17	LIST	106
7.18	MENU	107
7.19	METHOD.....	119
7.20	PROGRAM	121
7.21	RECORD.....	123
7.22	REFERENCE_ARRAY	123
7.23	Relations.....	124
7.24	RESPONSE_CODES.....	126

7.25 SOURCE	126
7.26 VALUE_ARRAY	130
7.27 VARIABLE	131
7.28 VARIABLE_LIST	150
7.29 WAVEFORM	150
7.30 Common attributes	159
7.31 Output redirection (OPEN and CLOSE)	162
7.32 Conditional expression	162
7.33 Referencing	164
7.34 Strings	171
7.35 Expression	174
7.36 Text dictionary	180
Annex A (normative) EDDL formal definition	181
Annex B (normative) EDDL Builtin library	269
Annex C (informative) EDD example	410
Annex D (normative) Profiles of EDDL and Builtins	425
Annex E	454

Figure 1 – Position of the IEC 61804 series related to other standards and products	24
Figure 2 – EDD generation process	37
Figure 3 – BLOCK_A	40
Figure 4 – CHART	41
Figure 5 – COLLECTION	41
Figure 6 – COMMAND	41
Figure 7 – DOMAIN	42
Figure 8 – EDIT_DISPLAY	43
Figure 9 – FILE	43
Figure 10 – GRAPH	43
Figure 11 – GRID	43
Figure 12 – IMAGE	44
Figure 13 – LIKE	44
Figure 14 – LIST	44
Figure 15 – MENU	45
Figure 16 – PROGRAM	45
Figure 17 – RECORD	45
Figure 18 – REFERENCE_ARRAY	46
Figure 19 – REFRESH	46
Figure 20 – UNIT	46
Figure 21 – WRITE_AS_ONE	47

Figure 22 – SOURCE.....	47
Figure 23 – VALUE_ARRAY.....	47
Figure 24 – VARIABLE.....	47
Figure 25 – VARIABLE_LIST.....	48
Figure 26 – WAVEFORM	48
Figure 27 – EDDL import mechanisms.....	90
Figure 28 – MENU activation (ACCESS OFFLINE)	116
Figure 29 – Action performed after a new value is entered	116
Figure 30 – Action performed after all VARIABLE inputs of the MENU are accepted (ACCESS OFFLINE)	117
Figure 31 – Method execution	117
Figure 32 – MENU activation (ACCESS ONLINE)	118
Figure 33 – Cyclic reading of dynamic VARIABLES (ACCESS ONLINE)	118
Figure 34 – Action performed after all VARIABLE inputs of the MENU are accepted (ACCESS ONLINE)	119
Figure 35 – Time for read-and-write operation	149
Figure C.1 – Example of an operator screen using EDD.....	410
Table 1 – Field attribute descriptions.....	34
Table 2 – DD_REVISION attribute.....	50
Table 3 – DEVICE_REVISION attribute	51
Table 4 – DEVICE_TYPE attribute	51
Table 5 – EDD_PROFILE attribute	51
Table 6 – EDD_VERSION attribute.....	53
Table 7 – MANUFACTURER attribute.....	53
Table 8 – MANUFACTURER_EXT attribute	53
Table 9 – AXIS attributes	54
Table 10 – MAX_VALUE, MIN_VALUE attribute.....	54
Table 11 – SCALING attribute	55
Table 12 – BLOCK_A attributes.....	56
Table 13 – CHARACTERISTIC attribute	57
Table 14 – PARAMETER attributes	57
Table 15 – AXIS_ITEMS attribute.....	58
Table 16 – CHART_ITEMS attribute	58
Table 17 – COLLECTION_ITEMS attribute	58
Table 18 – EDIT_DISPLAY_ITEMS attribute.....	59
Table 19 – FILE_ITEMS attribute	59
Table 20 – GRAPH_ITEMS attribute.....	59
Table 21 – GRID_ITEMS attribute	60
Table 22 – IMAGE_ITEMS attribute.....	60

Table 23 – LIST_ITEMS attribute	60
Table 24 – MENU_ITEMS attribute	61
Table 25 – METHOD_ITEMS attribute	61
Table 26 – PARAMETER_LISTS attributes	62
Table 27 – REFERENCE_ARRAY_ITEMS attribute	62
Table 28 – REFRESH_ITEMS attribute	63
Table 29 – SOURCE_ITEMS attribute	63
Table 30 – UNIT_ITEMS attribute	63
Table 31 – WAVEFORM_ITEMS attribute	63
Table 32 – WRITE_AS_ONE_ITEMS attribute	64
Table 33 – BLOCK_B attributes	64
Table 34 – NUMBER attributes	65
Table 35 – TYPE attributes	65
Table 36 – CHART attributes	66
Table 37 – CYCLE_TIME attribute	66
Table 38 – HEIGHT/WIDTH attribute	67
Table 39 – LENGTH attribute	68
Table 40 – TYPE attribute	68
Table 41 – COLLECTION attributes	69
Table 42 – item-type	69
Table 43 – COMMAND attributes	71
Table 44 – OPERATION attribute	71
Table 45 – TRANSACTION attributes	72
Table 46 – REPLY and REQUEST attributes	73
Table 47 – INDEX attribute	74
Table 48 – BLOCK_B attribute	74
Table 49 – NUMBER attribute	75
Table 50 – SLOT attribute	75
Table 51 – CONNECTION attribute	75
Table 52 – HEADER attribute	76
Table 53 – MODULE attribute	76
Table 54 – CONNECTION attribute	77
Table 55 – APPINSTANCE attribute	77
Table 56 – DOMAIN attributes	77
Table 57 – HANDLING attribute	78
Table 58 – EDIT_DISPLAY attributes	78
Table 59 – EDIT_ITEMS attribute	79
Table 60 – DISPLAY_ITEM attributes	79
Table 61 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS attribute	80

Table 62 – FILE attributes	81
Table 63 – GRAPH attributes	82
Table 64 – HEIGHT/WIDTH attribute	82
Table 65 – CYCLE_TIME attribute	83
Table 66 – X_AXIS attribute	83
Table 67 – GRID attributes	85
Table 68 – VECTORS attribute	85
Table 69 – HANDLING attribute	86
Table 70 – HEIGHT/WIDTH attribute	86
Table 71 – ORIENTATION attribute	87
Table 72 – VALIDITY attributes	87
Table 73 – IMAGE attributes	88
Table 74 – PATH attribute	88
Table 75 – LINK attribute	89
Table 76 – VALIDITY attributes	89
Table 77 – Importing Device Description	90
Table 78 – Redefinition attributes	92
Table 79 – Redefinition rules for AXIS attributes	92
Table 80 – Redefinition rules for BLOCK_A attributes	93
Table 81 – Redefinition rules for BLOCK_B attributes	93
Table 82 – Redefinition rules for CHART attributes	94
Table 83 – Redefinition rules for COLLECTION attributes	94
Table 84 – Redefinition rules for COMMAND attributes	94
Table 85 – Redefinition rules for CONNECTION attributes	95
Table 86 – Redefinition rules for DOMAIN attributes	95
Table 87 – Redefinition rules for EDIT_DISPLAY attributes	95
Table 88 – Redefinition rules for FILE attributes	96
Table 89 – Redefinition rules for GRAPH attributes	97
Table 90 – Redefinition rules for GRID attributes	97
Table 91 – Redefinition rules for IMAGE attributes	97
Table 92 – Redefinition rules for LIST attributes	98
Table 93 – Redefinition rules for MENU attributes	98
Table 94 – Redefinition rules for METHOD attributes	100
Table 95 – Redefinition rules for PROGRAM attributes	100
Table 96 – Redefinition rules for RECORD attributes	100
Table 97 – Redefinition rules for REFERENCE_ARRAY attributes	100
Table 98 – Redefinition rules for RESPONSE_CODES attributes	101
Table 99 – Redefinition rules for SOURCE attributes	101
Table 100 – Redefinition rules for VALUE_ARRAY attributes	102

Table 101 – Redefinition rules for VARIABLE attributes	103
Table 102 – Redefinition rules for VARIABLE_LIST attributes	105
Table 103 – Redefinition rules for WAVEFORM attributes	105
Table 104 – LIKE attributes	106
Table 105 – LIST attributes	106
Table 106 – TYPE attribute	107
Table 107 – CAPACITY, COUNT attribute	107
Table 108 – MENU attribute	109
Table 109 – ITEMS attribute	110
Table 110 – ACCESS attribute	110
Table 111 – ENTRY attribute	111
Table 112 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS, POST_READ_ACTIONS, PRE_READ_ACTIONS, POST_WRITE_ACTIONS, PRE_WRITE_ACTIONS attributes	111
Table 113 – PURPOSE attribute	113
Table 114 – ROLE attribute	114
Table 115 – STYLE attribute	115
Table 116 – VALIDITY attributes	115
Table 117 – METHOD attributes	119
Table 118 – ACCESS attributes	120
Table 119 – TYPE attributes	121
Table 120 – VALIDITY attributes	121
Table 121 – PROGRAM attributes	122
Table 122 – ARGUMENT attribute	122
Table 123 – RECORD attributes	123
Table 124 – REFERENCE_ARRAY attribute	123
Table 125 – ELEMENTS attribute	124
Table 126 – REFRESH attributes	125
Table 127 – UNIT attributes	125
Table 128 – WRITE_AS_ONE attribute	126
Table 129 – RESPONSE_CODES attributes	126
Table 130 – SOURCE attributes	127
Table 131 – EMPHASIS attribute	127
Table 132 – LINE_COLOR attribute	128
Table 133 – LINE_TYPE attribute	128
Table 134 – Y_AXIS attribute	129
Table 135 – VALUE_ARRAY attributes	130
Table 136 – NUMBER_OF_ELEMENTS attribute	131
Table 137 – TYPE attribute	131
Table 138 – VARIABLE attributes	131

Table 139 – CLASS attributes	132
Table 140 – TYPE attributes	134
Table 141 – DOUBLE, FLOAT, INTEGER, UNSIGNED_INTEGER attributes	136
Table 142 – BIT_ENUMERATED attributes.....	140
Table 143 – status–class attributes	140
Table 144 – ALL, AO, DV, TV attributes	141
Table 145 – Enumerated types attributes	142
Table 146 – Index type attributes	142
Table 147 – Object reference type attribute	143
Table 148 – DEFAULT_REFERENCE attributes	143
Table 149 – String types attributes	144
Table 150 – CONSTANT_UNIT attribute.....	145
Table 151 – DEFAULT_VALUE attribute.....	145
Table 152 – HANDLING attribute	146
Table 153 – INITIAL_VALUE attribute	146
Table 154 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS, POST_READ_ACTIONS, PRE_READ_ACTIONS, POST_WRITE_ACTIONS, PRE_WRITE_ACTIONS, REFRESH_ACTIONS attributes	147
Table 155 – READ/WRITE_TIMEOUT attributes	149
Table 156 – STYLE attribute	149
Table 157 – VALIDITY attributes	150
Table 158 – VARIABLE_LIST attributes.....	150
Table 159 – WAVEFORM attributes	151
Table 160 – TYPE attribute	151
Table 161 – XY attribute	152
Table 162 – YT attribute	152
Table 163 – HORIZONTAL attribute	154
Table 164 – VERTICAL attribute	154
Table 165 – EMPHASIS attribute	154
Table 166 – HANDLING attribute	155
Table 167 – EXIT_ACTIONS, INIT_ACTIONS, REFRESH_ACTIONS attribute.....	155
Table 168 – KEY_POINTS attribute.....	156
Table 169 – X_VALUES, Y_VALUES attribute	156
Table 170 – LINE_COLOR attribute.....	157
Table 171 – LINE_TYPE attribute.....	157
Table 172 – Y_AXIS attribute	159
Table 173 – DEFINITION attributes	160
Table 174 – HELP attribute	160
Table 175 – LABEL attribute	160
Table 176 – MEMBERS attributes	161

Table 177 – RESPONSE_CODES attribute.....	162
Table 178 – OPEN and CLOSE attributes.....	162
Table 179 – IF, SELECT conditional.....	163
Table 180 – Referencing an EDD instance	164
Table 181 – Referencing elements of VARIABLE	164
Table 182 – Referencing elements of RECORD	165
Table 183 – Referencing elements of VALUE_ARRAY	166
Table 184 – Referencing members of COLLECTION.....	166
Table 185 – Referencing members of REFERENCE_ARRAY	166
Table 186 – Referencing members of VARIABLE_LISTS	167
Table 187 – Referencing members of a BLOCK_A PARAMETERS.....	167
Table 188 – Referencing members of BLOCK_A PARAMETER_LISTS.....	167
Table 189 – Referencing members of BLOCK_A LOCAL_PARAMETER	168
Table 190 – Referencing BLOCK_A CHARACTERISTICS	168
Table 191 – Referencing members of FILE	168
Table 192 – Referencing elements of LIST	169
Table 193 – Referencing members of CHART.....	169
Table 194 – Referencing members of GRAPH	169
Table 195 – Referencing members of SOURCE.....	170
Table 196 – Referencing AXIS of a GRAPH. SOURCE, WAVEFORM.....	171
Table 197 – String as a string literal	171
Table 198 – String as a string variable	172
Table 199 – String as an enumeration value	172
Table 200 – String as a dictionary reference.....	172
Table 201 – Referencing HELP and LABEL attributes of EDD instances.....	173
Table 202 – String operation	173
Table 203 – Format specifier.....	173
Table 204 – Primary expressions	174
Table 205 – Attribute values of VARIABLES.....	175
Table 206 – AXIS Attribute Values	175
Table 207 – LIST Attribute Values.....	176
Table 208 – Unary expressions	176
Table 209 – Multiplicative operators	177
Table 210 – Additive operators.....	177
Table 211 – Shift operators.....	177
Table 212 – Relational operators.....	178
Table 213 – Equality operators.....	178
Table 214 – Text dictionary attributes.....	180
Table A.1 – Conventions for integer constants.....	185

Table A.2 – Using escape sequences in string literals.....	186
Table A.3 – Using language codes in string literals.....	187
Table A.4 – EDDL operators	188
Table A.5 – EDDL keywords.....	188
Table B.1 – Format for the Builtins lexical element tables	269
Table B.2 – Contents of the lexical element table	269
Table B.3 – Builtin abort	270
Table B.4 – Builtin abort_on_all_comm_errors.....	270
Table B.5 – Builtin ABORT_ON_ALL_COMM_STATUS.....	271
Table B.6 – Builtin ABORT_ON_ALL_DEVICE_STATUS.....	271
Table B.7 – Builtin ABORT_ON_ALL_RESPONSE_CODES	272
Table B.8 – Builtin abort_on_all_response_codes.....	272
Table B.9 – Builtin abort_on_comm_error.....	273
Table B.10 – Builtin ABORT_ON_COMM_ERROR.....	274
Table B.11 – Builtin ABORT_ON_COMM_STATUS.....	274
Table B.12 – Builtin ABORT_ON_DEVICE_STATUS.....	275
Table B.13 – Builtin ABORT_ON_NO_DEVICE	275
Table B.14 – Builtin ABORT_ON_RESPONSE_CODE	276
Table B.15 – Builtin abort_on_response_code	277
Table B.16 – Builtin abs	277
Table B.17 – Builtin ACKNOWLEDGE	278
Table B.18 – Builtin acknowledge.....	278
Table B.19 – Builtin acos	278
Table B.20 – Builtin add_abort_method	279
Table B.21 – Builtin add_abort_method	280
Table B.22 – Builtin asin	280
Table B.23 – Builtin assign.....	280
Table B.24 – Builtin assign_double.....	281
Table B.25 – Builtin assign_float	281
Table B.26 – Builtin assign_int	282
Table B.27 – Builtin assign_var	283
Table B.28 – Builtin atan.....	283
Table B.29 – Builtin ATOF.....	283
Table B.30 – Builtin atof.....	284
Table B.31 – Builtin ATOI.....	284
Table B.32 – Builtin atoi.....	285
Table B.33 – Builtin BUILD_MESSAGE	286
Table B.34 – Builtin cbrt.....	286
Table B.35 – Builtin ceil	286

Table B.36 – Builtin cos	287
Table B.37 – Builtin cosh	287
Table B.38 – Builtin dassign.....	287
Table B.39 – Builtin Date_to_DayOfMonth.....	288
Table B.40 – Builtin Date_to_Month	288
Table B.41 – Builtin Date_to_Year.....	289
Table B.42 – Builtin DELAY	289
Table B.43 – Builtin delay	290
Table B.44 – Builtin DELAY_TIME.....	291
Table B.45 – Builtin delayfor	291
Table B.46 – Builtin DICT_ID	292
Table B.47 – Builtin discard_on_exit.....	293
Table B.48 – Builtin display.....	293
Table B.49 – Builtin display_builtin_error.....	293
Table B.50 – Builtin display_comm_error.....	295
Table B.51 – Builtin display_comm_status.....	295
Table B.52 – Builtin display_device_status	295
Table B.53 – Builtin display_dynamics.....	296
Table B.54 – Builtin display_message	297
Table B.55 – Builtin display_response_code.....	298
Table B.56 – Builtin display_response_status	299
Table B.57 – Builtin display_xmtr_status	299
Table B.58 – Builtin edit_device_value	300
Table B.59 – Builtin edit_local_value	302
Table B.60 – Builtin exp	302
Table B.61 – Builtin ext_send_command	304
Table B.62 – Builtin ext_send_command_trans.....	304
Table B.63 – Builtin fail_on_all_comm_errors	305
Table B.64 – Builtin fail_on_all_response_codes	306
Table B.65 – Builtin fail_on_comm_error	306
Table B.66 – Builtin fail_on_response_code	307
Table B.67 – Builtin fassign.....	307
Table B.68 – Builtin fgetval	308
Table B.69 – Builtin float_value	308
Table B.70 – Builtin floor.....	308
Table B.71 – Builtin fmod	309
Table B.72 – Builtin fsetval	309
Table B.73 – Builtin ftoa.....	309
Table B.74 – Builtin fvar_value.....	310

Table B.75 – Builtin get_acknowledgement.....	311
Table B.76 – Builtin get_comm_error.....	312
Table B.77 – Builtin get_comm_error_string	312
Table B.78 – Builtin get_date	314
Table B.79 – Builtin get_date_value	314
Table B.80 – Builtin get_dds_error	315
Table B.81 – Builtin GET_DEV_VAR_VALUE	315
Table B.82 – Builtin get_dev_var_value.....	317
Table B.83 – Builtin get_dictionary_string.....	318
Table B.84 – Builtin get_double.....	318
Table B.85 – Builtin get_double_value.....	320
Table B.86 – Builtin get_enum_string	320
Table B.87 – Builtin get_float	321
Table B.88 – Builtin get_float_value	321
Table B.89 – Builtin GET_LOCAL_VAR_VALUE	322
Table B.90 – Builtin get_local_var_value	322
Table B.91 – Builtin get_more_status	324
Table B.92 – Builtin get_resolve_status.....	324
Table B.93 – Builtin get_response_code.....	325
Table B.94 – Builtin get_response_code_string	326
Table B.95 – Builtin get_signed.....	326
Table B.96 – Builtin get_signed_value.....	327
Table B.97 – Builtin get_status_code_string	327
Table B.98 – Builtin get_status_string	328
Table B.99 – Builtin get_stddict_string	329
Table B.100 – Builtin get_string	330
Table B.101 – Builtin get_string_value	330
Table B.102 – Builtin GET_TICK_COUNT	331
Table B.103 – Builtin get_unsigned	331
Table B.104 – Builtin get_unsigned_value	332
Table B.105 – Builtin iassign.....	332
Table B.106 – Builtin igetval.....	333
Table B.107 – Builtin IGNORE_ALL_COMM_STATUS	333
Table B.108 – Builtin IGNORE_ALL_DEVICE_STATUS	334
Table B.109 – Builtin IGNORE_ALL_RESPONSE_CODES.....	335
Table B.110 – Builtin IGNORE_COMM_ERROR	335
Table B.111 – Builtin IGNORE_COMM_STATUS	336
Table B.112 – Builtin IGNORE_DEVICE_STATUS	336
Table B.113 – Builtin IGNORE_NO_DEVICE	337

Table B.114 – Builtin IGNORE_RESPONSE_CODE.....	338
Table B.115 – Builtin int_value.....	338
Table B.116 – Builtin is_NaN	338
Table B.117 – Builtin isetval.....	339
Table B.118 – Builtin ITEM_ID	339
Table B.119 – Builtin ITOA.....	340
Table B.120 – Builtin itoa	340
Table B.121 – Builtin ivar_value	340
Table B.122 – Builtin lassign	341
Table B.123 – Builtin lgetval.....	341
Table B.124 – Builtin ListDeleteElementAt.....	342
Table B.125 – Builtin ListInsert.....	342
Table B.126 – Builtin log	343
Table B.127 – Builtin log10	343
Table B.128 – Builtin log2	345
Table B.129 – Lexical elements of Builtin LOG_MESSAGE	345
Table B.130 – Builtin long_value	345
Table B.131 – Builtin lsetval.....	346
Table B.132 – Builtin lvar_value	346
Table B.133 – Builtin MEMBER_ID	347
Table B.134 – Builtin MenuDisplay	347
Table B.135 – Builtin method_abort.....	348
Table B.136 – Builtin process_abort.....	348
Table B.137 – Builtin pow	349
Table B.138 – Builtin process_abort.....	349
Table B.139 – Builtin put_date	350
Table B.140 – Builtin put_date_value	350
Table B.141 – Builtin put_double.....	351
Table B.142 – Builtin put_double_value.....	352
Table B.143 – Builtin put_float	352
Table B.144 – Builtin put_float_value	353
Table B.145 – Builtin PUT_MESSAGE.....	353
Table B.146 – Builtin put_message	354
Table B.147 – Builtin put_signed	355
Table B.148 – Builtin put_signed_value	355
Table B.149 – Builtin put_string	356
Table B.150 – Builtin put_string_value	357
Table B.151 – Builtin put_unsigned	357
Table B.152 – Builtin put_unsigned_value	358

Table B.153 – Lexical elements of Builtin READ_COMMAND.....	359
Table B.154 – Builtin read_value.....	359
Table B.155 – Builtin remove_abort_method.....	360
Table B.156 – Builtin remove_abort_method.....	360
Table B.157 – Builtin remove_all_abort_methods	361
Table B.158 – Builtin resolve_array_ref.....	362
Table B.159 – Builtin resolve_block_ref.....	362
Table B.160 – Builtin resolve_param_list_ref	363
Table B.161 – Builtin resolve_parm_ref	363
Table B.162 – Builtin resolve_record_ref.....	364
Table B.163 – Builtin retry_on_all_comm_errors.....	365
Table B.164 – Builtin RETRY_ON_ALL_COMM_STATUS	366
Table B.165 – Builtin RETRY_ON_ALL_DEVICE_STATUS	367
Table B.166 – Builtin RETRY_ON_ALL_RESPONSE_CODES.....	367
Table B.167 – Builtin retry_on_all_response_codes.....	368
Table B.168 – Builtin RETRY_ON_COMM_ERROR	368
Table B.169 – Builtin retry_on_comm_error	369
Table B.170 – Builtin RETRY_ON_COMM_STATUS	370
Table B.171 – Builtin RETRY_ON_DEVICE_STATUS	370
Table B.172 – Builtin RETRY_ON_NO_DEVICE	371
Table B.173 – Builtin RETRY_ON_RESPONSE_CODE.....	372
Table B.174 – Builtin retry_on_response_code	372
Table B.175 – Builtin round.....	373
Table B.176 – Builtin rspcode_string	373
Table B.177 – Builtin save_on_exit.....	374
Table B.178 – Builtin save_values.....	374
Table B.179 – Builtin SELECT_FROM_LIST	375
Table B.180 – Builtin select_from_list.....	375
Table B.181 – Builtin select_from_menu.....	376
Table B.182 – Builtin send	377
Table B.183 – Builtin send_all_values	377
Table B.184 – Builtin send_command.....	378
Table B.185 – Builtin send_command_trans	379
Table B.186 – Builtin send_on_exit	379
Table B.187 – Builtin send_trans.....	380
Table B.188 – Builtin send_value	380
Table B.189 – Builtin SET_NUMBER_OF_RETRIES.....	381
Table B.190 – Builtin sin	381
Table B.191 – Builtin sinh	382

Table B.192 – Builtin sqrt.....	382
Table B.193 – Builtin strcmp	382
Table B.194 – Builtin strlen	383
Table B.195 – Builtin strlwr	383
Table B.196 – Builtin strmid	384
Table B.197 – Builtin strstr.....	384
Table B.198 – Builtin strtrim.....	384
Table B.199 – Builtin strupr.....	385
Table B.200 – Builtin tan	385
Table B.201 – Builtin tanh	385
Table B.202 – Builtin To_Date_and_Time.....	386
Table B.203 – Builtin trunc.....	386
Table B.204 – Builtin VARID	388
Table B.205 – Builtin vassign	388
Table B.206 – Builtin WRITE_COMMAND	388
Table B.207 – Builtin XMTR_ABORT_ON_ALL_COMM_STATUS.....	389
Table B.208 – Builtin XMTR_ABORT_ON_ALL_DEVICE_STATUS.....	390
Table B.209 – Builtin XMTR_ABORT_ON_ALL_RESPONSE_CODES	390
Table B.210 – Builtin XMTR_ABORT_ON_COMM_ERROR.....	391
Table B.211 – Builtin XMTR_ABORT_ON_COMM_STATUS.....	391
Table B.212 – Builtin XMTR_ABORT_ON_DATA	392
Table B.213 – Builtin XMTR_ABORT_ON_DEVICE_STATUS.....	393
Table B.214 – Builtin XMTR_ABORT_ON_NO_DEVICE.....	394
Table B.215 – Builtin XMTR_ABORT_ON_RESPONSE_CODE	394
Table B.216 – Builtin XMTR_IGNORE_ALL_COMM_STATUS.....	395
Table B.217 – Builtin XMTR_IGNORE_ALL_DEVICE_STATUS	395
Table B.218 – Builtin XMTR_IGNORE_ALL_RESPONSE_CODES	396
Table B.219 – Builtin XMTR_IGNORE_COMM_ERROR.....	397
Table B.220 – Builtin XMTR_IGNORE_COMM_STATUS.....	397
Table B.221 – Builtin XMTR_IGNORE_DEVICE_STATUS.....	398
Table B.222 – Builtin XMTR_IGNORE_NO_DEVICE	399
Table B.223 – Builtin XMTR_IGNORE_RESPONSE_CODE	399
Table B.224 – Builtin XMTR_RETRY_ON_ALL_DEVICE_STATUS	400
Table B.225 – Builtin XMTR_RETRY_ON_ALL_RESPONSE_CODE	402
Table B.226 – Builtin XMTR_RETRY_ON_ALL_RESPONSE_CODES	402
Table B.227 – Builtin XMTR_RETRY_ON_COMM_ERROR.....	403
Table B.228 – Builtin XMTR_RETRY_ON_COMM_STATUS.....	403
Table B.229 – Builtin XMTR_RETRY_ON_DATA	404
Table B.230 – Builtin XMTR_RETRY_ON_DEVICE_STATUS.....	404

Table B.231 – Builtin XMTR_RETRY_ON_NO_DEVICE	406
Table B.232 – Builtin XMTR_RETRY_ON_RESPONSE_CODE	406
Table B.233 – Builtin YearMonthDay_to_Date	408
Table B.234 – Contents of the return codes description table	408
Table B.235 – Return Code Description	408
Table D.1 – Profile selection tables	425
Table D.2 – EDDL Formal Definition profile tables	425
Table D.3 – Contents of selection tables	425
Table D.4 – EDDL element selection for PROFIBUS	426
Table D.5 – Builtin profile for PROFIBUS	429
Table D.6 – EDDL element selection for Fieldbus Foundation	433
Table D.7 – Builtin profile for Fieldbus Foundation	437
Table D.8 – EDDL element selection for HCF	441
Table D.9 – Builtin profile for HCF	445
Table D.10 – METHOD DEFINITIONS data types	449
Table D.11 – VARIABLE TYPES	449
Table D.12 – DATE coding	450
Table D.13 – DATE_AND_TIME coding	451
Table D.14 – DURATION coding	451
Table D.15 – TIME coding	452
Table D.16 – TIME_VALUE coding	452
Table D.17 – PACKED_ASCII coding	453

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FUNCTION BLOCKS (FB) FOR PROCESS CONTROL –

Part 3: Electronic Device Description Language (EDDL)

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 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.

IEC take no position concerning the evidence, validity and scope of this patent right. The holder of this patent right has assured the IEC that he is willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC.

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

This International Standard has been prepared by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This first edition cancels and replaces the EDDL specification given in the first edition of IEC 61804-2, published in 2004¹.

The text of this standard is based on the following documents:

CDV	Report on voting
65C/406/CDV	65C/421/RVC

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

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

The list of all the parts of the IEC 61804 series, under the general title *Function Blocks (FB) for process control*, can be found on the IEC website.

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

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

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

¹ The historical background to the EDDL specification is given in Annex E.

INTRODUCTION

The EDDL fills the gap between the conceptual FB 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.

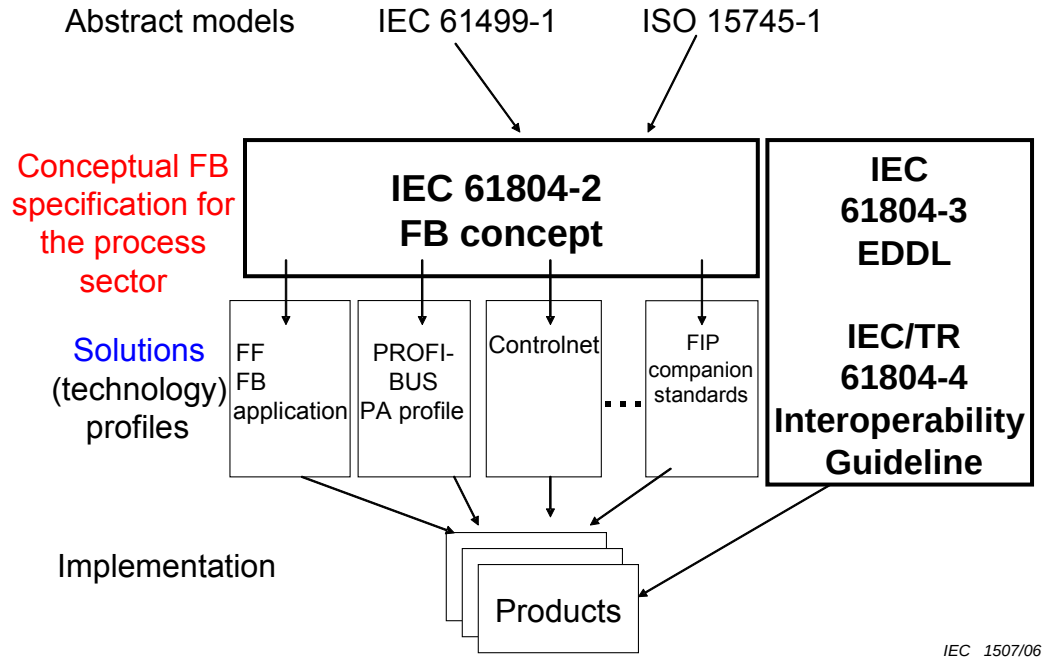


Figure 1 – Position of the IEC 61804 series related to other standards and products

FUNCTION BLOCKS (FB) FOR PROCESS CONTROL –

Part 3: Electronic Device Description Language (EDDL)

1 Scope

This part of IEC 61804 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 standard 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 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.

NOTE 1 The EDDL may also be used for the description of product properties in other domains.

The EDDL and the device-related EDD is applicable to industrial automation.

NOTE 2 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.

This International Standard satisfies the requirements of Clause 9 of IEC 61804-1.

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 61804-1:2003, *Function blocks (FB) for process control – Part 1: Overview of system aspects*

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*

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

ISO 639, *Codes for the representation of names of languages*

ISO 3166, *Codes for the representation of names of countries and their subdivisions*

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

3 Terms, definitions, abbreviated terms and acronyms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions, some of which have been compiled from the referenced documents, apply.

NOTE The terms in IEC 60050-351 apply partially.

3.2

algorithm

finite set of well-defined rules for the solution of a problem in a finite number of operations

3.3

attribute

description of an externally visible characteristic or feature of an object

NOTE The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behaviour of an object. Attributes are divided into class attributes and instance attributes.

3.4

Builtin

predefined subroutine for communication and display executed by the EDD application

3.5

configuration (of a system or device)

step in system design: selecting functional units, assigning their locations and defining their interconnections

[IEC 61499-1]

3.6

data

representation of facts, concepts or instructions in a formalized manner suitable for communication, interpretation or processing by human beings or by automatic means

[ISO/AFNOR Dictionary of Computer Science]

3.7

data connection

association established between functional units for conveyance of data

[IEC 61499-1]

3.8

data input

interface of an FB which receives data from a data connection

[IEC 61499-1]

3.9

data output

interface of an FB, which supplies data to a data connection

[IEC 61499-1]

3.10

data type

set of values together with a set of permitted operations

[ISO 2382 series]

3.11

device

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

[IEC 61499-1]

3.12

EDD application

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

3.13

EDDL processor

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

3.14

EDDL profile

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

3.15

Electronic Device Description Language

EDDL

methodology for describing parameter(s) of an automation system component

3.16

Electronic Device Description

EDD

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

NOTE The Electronic Device Description is created using the Electronic Device Description Language (EDDL)

3.17

Electronic Device Description Source

EDDS

ASCII file containing a specific device description

3.18

Electronic Device Description Technology

EDDT

technology which includes the EDD development process, the EDD usage and the involved tool chain

3.19

Electronic Device Description Checker

test tool which checks the Lexical Structure and partly the semantic of EDD sources to guarantee compliance of EDD sources with the EDD language

3.20

Electronic Device Description Compiler

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

3.21

Electronic Device Description Interpreter

EDDI

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

3.22

entity

particular thing, such as a person, place, process, object, concept, association, or event

[IEC 61499-1]

3.23

event

instantaneous occurrence that is significant to scheduling the execution of an algorithm

[IEC 61499-1]

NOTE The execution of an algorithm may make use of variables associated with an event.

3.24

exception

event that causes suspension of normal execution

[IEC 61499-1]

3.25

function

specific purpose of an entity or its characteristic action

[IEC 61499-1]

3.26

functional unit

entity of hardware or software, or both, capable of accomplishing a specified purpose

[ISO/AFNOR Dictionary of Computer Science]

3.27

function block (function block instance)

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

[IEC 61499-1]

NOTE Typical operations of a FB include modification of the values of the data in its associated data structure.

3.28

hardware

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

[ISO/AFNOR Dictionary of Computer Science]

3.29

implementation

development phase in which the hardware and software of a system become operational

[IEC 61499-1]

3.30

input variable

variable whose value is supplied by a data input, and which may be used in one or more operations of a FB

NOTE An input parameter of an FB, as defined in IEC 61131-3, is an input variable.

[IEC 61499-1]

3.31

instance

functional unit comprising an individual, named entity with the attributes of a defined type

[IEC 61499-1]

3.32

instance name

identifier associated with, and designating, an instance

[IEC 61499-1]

3.33

instantiation

creation of an instance of a specified type

[IEC 61499-1]

3.34

interface

shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate

[IEC 60050-351:1998, 11-19]

3.35

internal variable

variable whose value is used or modified by one or more operations of an FB but is not supplied by a data input or to a data output

[IEC 61499-1]

3.36

invocation

process of initiating the execution of the sequence of operations specified in an algorithm

[IEC 61499-1]

3.37

mapping

set of values having defined correspondence with the quantities or values of another set

[ISO/AFNOR Dictionary of Computer Science]

3.38

model

representation of a real world process, device, or concept

[IEC 61499-1]

3.39

operation

well-defined action that, when applied to any permissible combination of known entities, produces a new entity

[ISO/AFNOR Dictionary of Computer Science]

3.40

output variable

variable whose value is established by one or more operations of a FB, and is supplied to a data output

NOTE An output parameter of an FB, as defined in IEC 61131-3, is an output variable.

[IEC 61499-1]

3.41

parameter

variable that is given a constant value for a specified application and that may denote the application

[ISO/AFNOR Dictionary of Computer Science]

3.42

preprocessor

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

3.43

preprocessor directives

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

NOTE For example, a preprocessor directive providing the facility to define names for constants or to write macros to make code easier to read

3.44

resource

functional unit contained within a device which has independent control of its operation, and which provides various services to applications, including the scheduling and execution of algorithms

NOTE 1 The RESOURCE defined in IEC 61131-3 is a programming language element corresponding to the resource defined above.

NOTE 2 A device contains one or more resources.

3.45

resource management application

application whose primary function is the management of a single resource

[IEC 61499-1]

3.46

service

functional capability of a resource, which can be modelled by a sequence of service primitives

[IEC 61499-1]

3.47

software

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

[IEC 61499-1]

3.48

system

set of interrelated elements considered in a defined context as a whole and separated from its environment

[IEV 350-11-01:1998]

NOTE 1 Such elements may be both material objects and concepts as well as the results thereof (for example, forms of organization, mathematical methods, and programming languages).

NOTE 2 The system is considered to be separated from the environment and other external systems by an imaginary surface, which can cut the links between them and the considered system.

3.49 text dictionary

collection of multilingual or other texts within the EDD

NOTE References within an EDD are used to select an appropriate text dictionary.

3.50 type

software element, which specifies the common attributes shared by all instances of the type

[IEC 61499-1]

3.51 type name

identifier associated with, and designating, a type

[IEC 61499-1]

3.52 variable

software entity that may take different values, one at a time

NOTE 1 The values of a variable are usually restricted to a certain data type.

NOTE 2 Variables are described as input variables, output variables, and internal variables.

[IEC 61499-1]

3.53 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)
ASN.1	Abstract Lexical Structure Notation 1
DD	Device Description
EDD	Electronic Device Description
EDDI	Electronic Device Description Interpreter
EDDL	Electronic Device Description Language
EUC	Extended Unit Code (see ISO/IEC 2022)
FB	Function Block
HMI	Human Machine Interface
HTML	Hypertext Mark-up Language
I/O	Input/Output

ID	Identifier
OSI	Open Systems Interconnection
wao	Write as one

4 Conformance statement

A statement of conformance to a profile of this International Standard shall be stated² as either

- compliance to IEC 61804-3:2006 Annex D.n <Type>
- or
- compliance to IEC 61804-3/Ed1.0 Annex D.n <Type>

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

NOTE The Type may be, for example, the name of the communication protocol used like PROFIBUS, Foundation Fieldbus, or HART.

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

5 Conventions for lexical structures

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

ABC field1, field2

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

Field1 and field2 are fields of the lexical element ABC. Each field is mandatory and may have more than one attribute. If a field has attributes, the presence of the attributes is specified in a table. A comma separates Field1 and 2. The comma is a lexical element and is not coded explicitly.

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

Table 1 – Field attribute descriptions

Usage	Attribute	Description
m	yyy	The presence of this attribute is mandatory

² In accordance with the ISO/IEC Directives.

Usage	Attribute	Description
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

The characters in the Usage column have the following meanings.

- m: this attribute is mandatory and shall be present.
- o: this attribute is optional and need not be present.
- s: this attribute is a selection. One, and only one, of the fields marked with s (Z1 or Z2) shall be present.
- c: this attribute is conditional. The condition is described in the Description column.

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

ABC field1+

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

ABC field2*

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

ABC [field1, field2]+

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

ABC field1, (field2, field3)<exp>

<exp> indicates that field2 and field3 are used in conjunction with the conditional expression, (conditional constructs are specified in 7.32). The expression <exp> refers only to the fields within the preceding curved brackets (). Usage of conditional expressions is optional.

6 EDD and EDDL model

6.1 Overview of EDD and EDDL

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

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

6.2 EDD architecture

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

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

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

6.3 Concepts of EDD

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

- description of the device parameters;
- description of parameter dependencies;
- logical grouping of the device parameters;
- selection and execution of supported device functions;
- description of the transferred data sets.

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

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

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

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

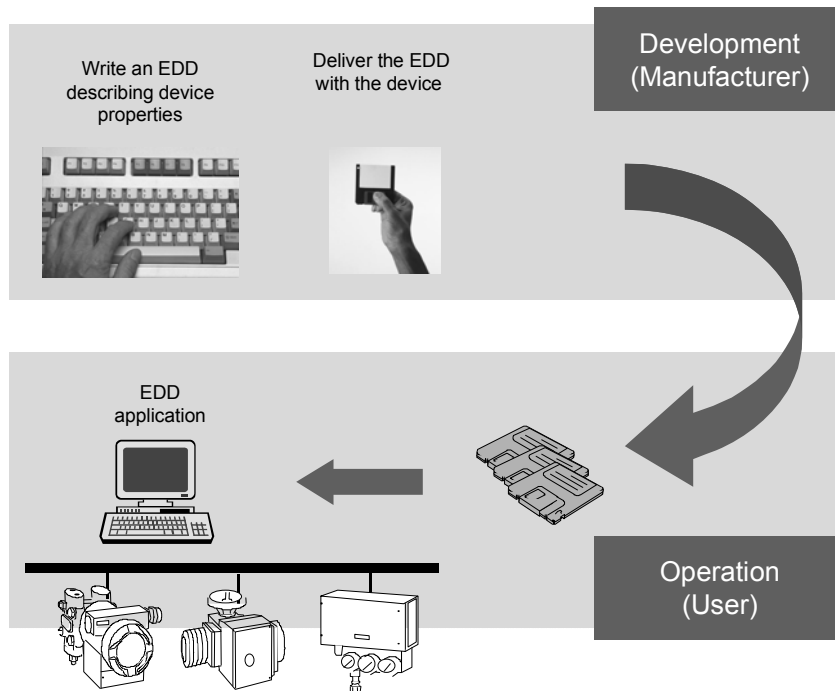
6.4 Principles of the EDD development process

6.4.1 General

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

6.4.2 EDD source generation

The EDD generation process is shown in Figure 2.



IEC 1511/06

Figure 2 – EDD generation process

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 Annex B.

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).

NOTE 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 7.36).

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

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

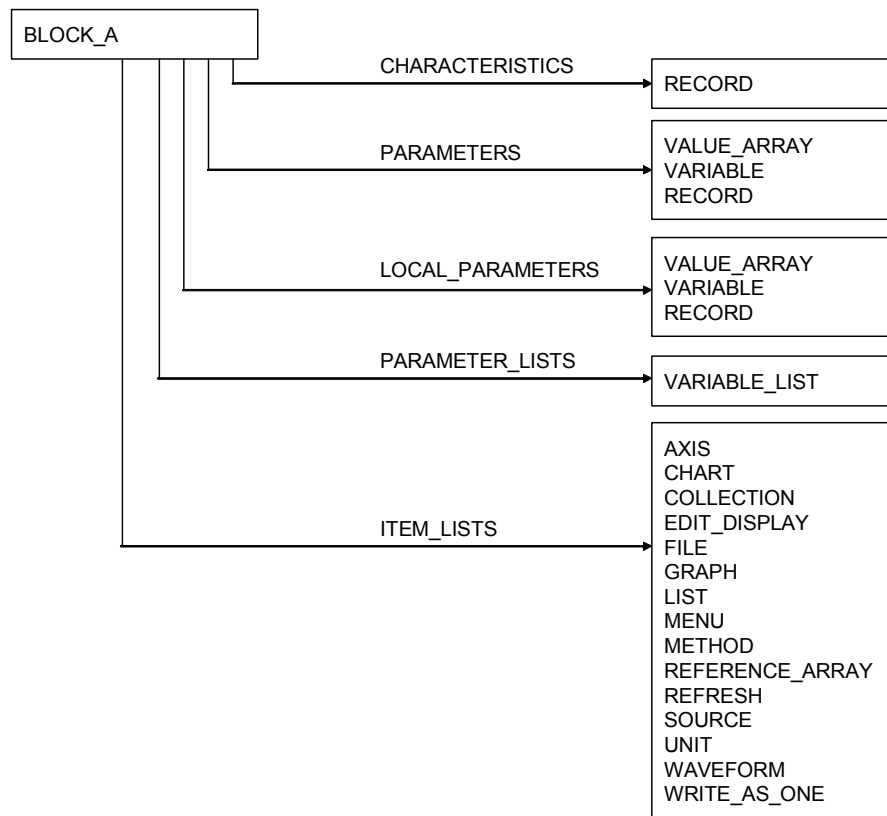
Some constructs have similar names and functions while differing in their detailed specification. This additional variety has been included to ensure compatibility with several existing description languages. Appropriate mapping cross-references are given by profiles.

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.

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

Some basic constructs refer to other basic constructs. In the following list, the basic constructs as well as the relationships between the basic constructs are shown:

- **AXIS** describes the axis of a CHART or GRAPH (see 7.3).
- **BLOCK_A** is a logical grouping of CHARACTERISTICS, PARAMETERS, PARAMETER_LISTS, and ITEM_LISTS, see Figure 3. To access one item of BLOCK_A, the instance of the block should be used (see 7.4.1 and Figure 3).



IEC 1512/06

Figure 3 – BLOCK_A

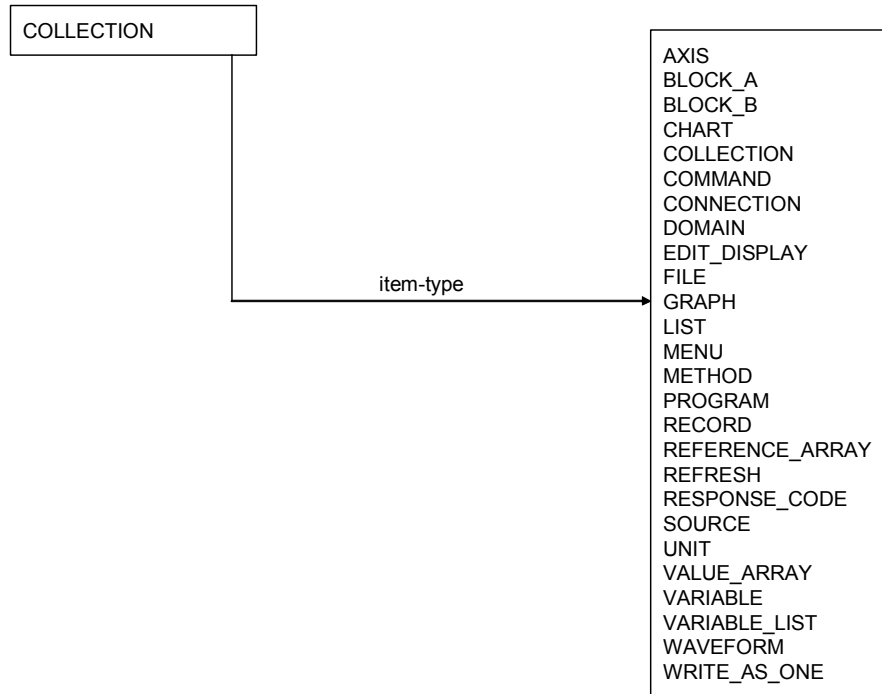
- **BLOCK_B** is used to structure variables in instances of logical block types. For accessing a VARIABLE within a BLOCK_B construct, the basic construct COMMAND is used to provide the relative addressing (see 7.4.2).
- **CHART** describes a chart used to display data from a device in the EDD application (see 7.5 and Figure 4).



IEC 1513/06

Figure 4 – CHART

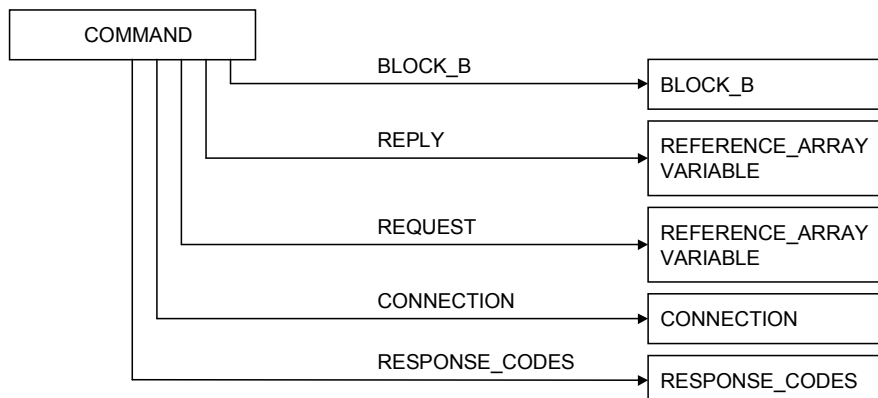
- **COLLECTION** describes logical groupings of data (see 7.6 and Figure 5).



IEC 1514/06

Figure 5 – COLLECTION

- **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

- **CONNECTION** defines multiple applications in a device (see 7.8).

- **DOMAIN** can be used to download or upload moderately large amounts of data to or from a device (see 7.9 and Figure 7).

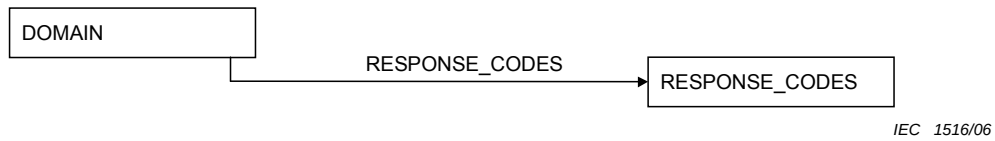


Figure 7 – DOMAIN

- **EDIT_DISPLAY** describes how the data will be represented to a user by a display device (see 7.10 and Figure 8).

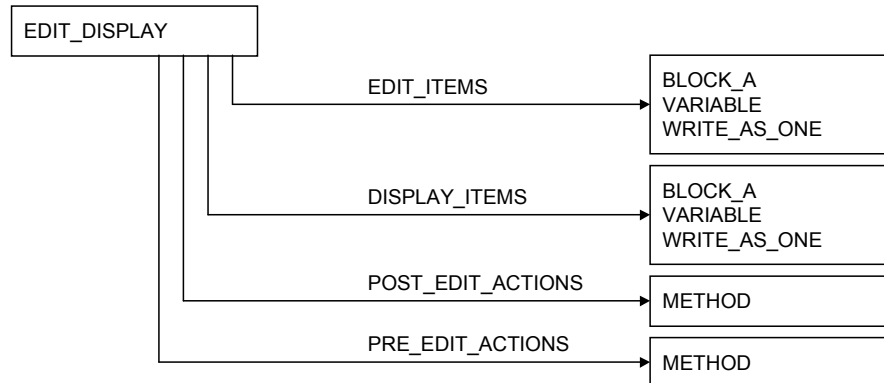


Figure 8 – EDIT_DISPLAY

IEC 1517/06

- **FILE** describes an area of persistent storage (see 7.11 and Figure 9).

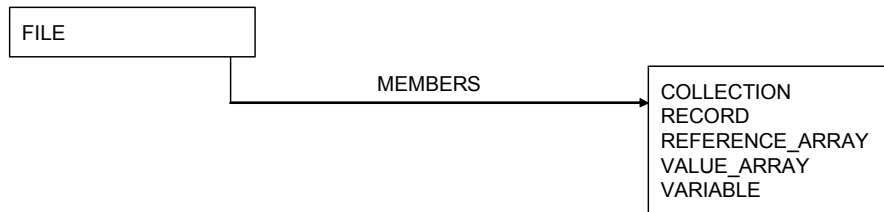


Figure 9 – FILE

IEC 1518/06

- **GRAPH** describes a graph used to display data from a device in the EDD application (see 7.12 and Figure 10).

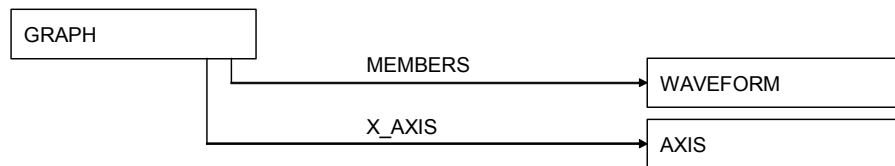
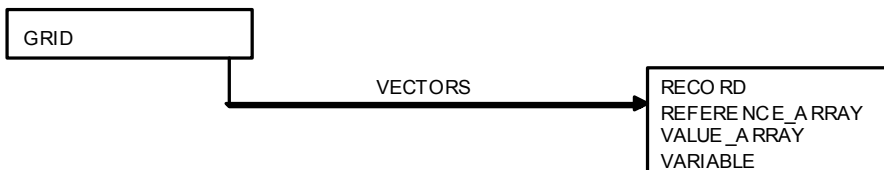


Figure 10 – GRAPH

IEC 1519/06

- **GRID** describes a matrix used to display data from a device in the EDD application (see 7.13 and Figure 11).



IEC 1520/06

Figure 11 – GRID

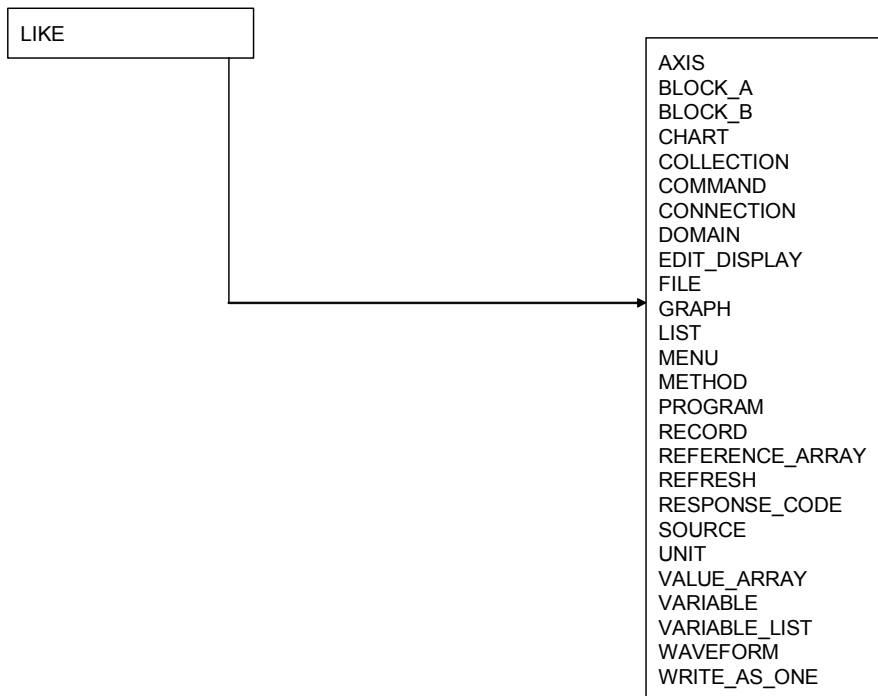
- **IMAGE** describes a visual image (see 7.13 and Figure 12).



IEC 1521/06

Figure 12 – IMAGE

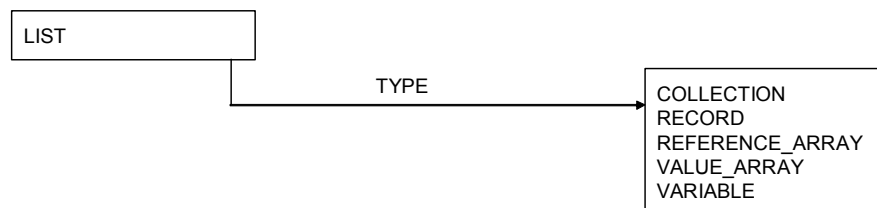
- **IMPORT** is used to import an EDD and to modify its existing definitions (see 7.13).
- **LIKE** creates a new instance of an existing instance of a basic construct. The attributes of the new instance may be redefined, added or deleted (see 7.16 and Figure 13).



IEC 1522/06

Figure 13 – LIKE

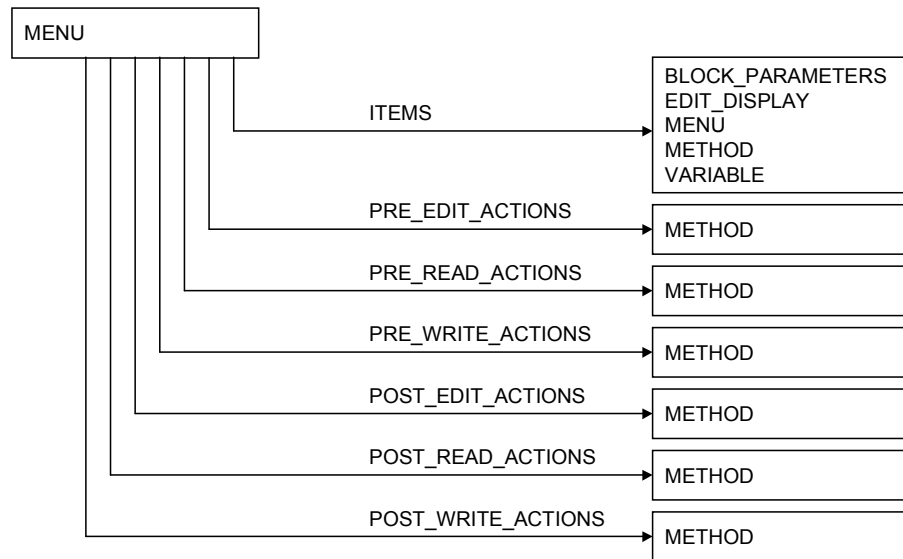
- **LIST** describes a sequence of elements (see 7.17 and Figure 14).



IEC 1523/06

Figure 14 – LIST

- **MENU** describes how the data will be presented to a user by an EDD application (see 7.18 and Figure 15).



IEC 1524/06

Figure 15 – MENU

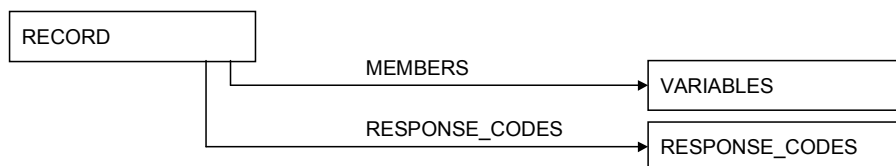
- **METHOD** describes the execution of complex sequence of event interactions that shall take place between system devices such as display units, configurators and field devices (see 7.19).
- **PROGRAM** specifies how device executable code can be initiated by an appropriate agent (see 7.20 and Figure 16).



IEC 1525/06

Figure 16 – PROGRAM

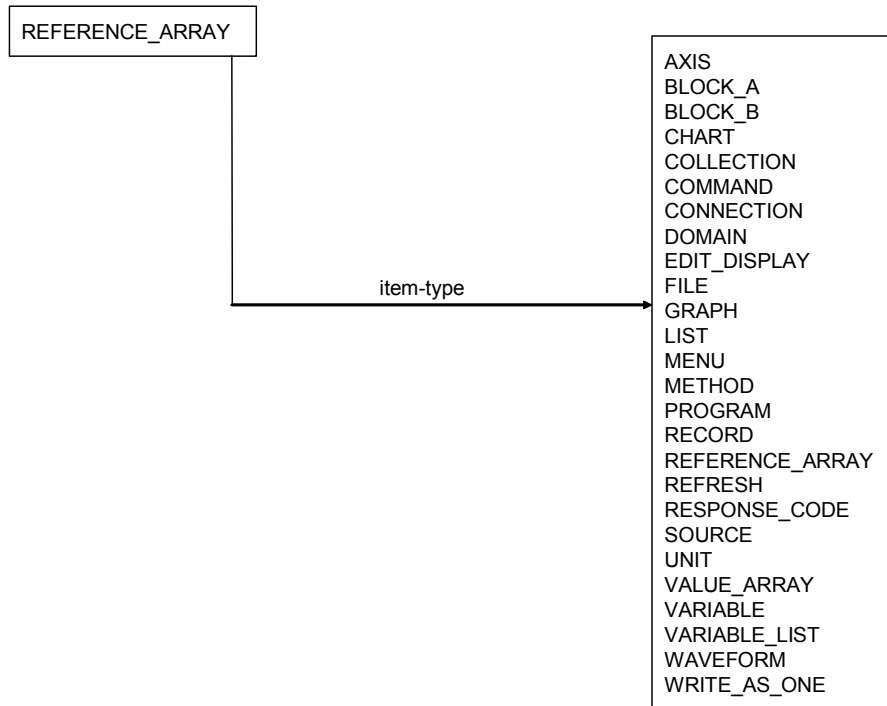
- **RECORD** describes the data contained in the device (see 7.21 and Figure 17).



IEC 1526/06

Figure 17 – RECORD

- **REFERENCE_ARRAY** is a set of EDD items of the same EDDL item type (for example, VARIABLE or MENU). An item can be referenced in the EDD via the REFERENCE_ARRAY name and the index associated with the item (see 7.22 and Figure 18).



IEC 1527/06

Figure 18 – REFERENCE_ARRAY

- **Relation:** three types of relations are specified.

REFRESH describes relationships between VARIABLES, RECORDs, and VALUE_ARRAYs (see 7.23.2 and Figure 19).



IEC 1528/06

Figure 19 – REFRESH

UNIT describes relationships between VARIABLES, RECORDs, and VALUE_ARRAYs (see 7.23.3 and Figure 20).



IEC 1529/06

Figure 20 – UNIT

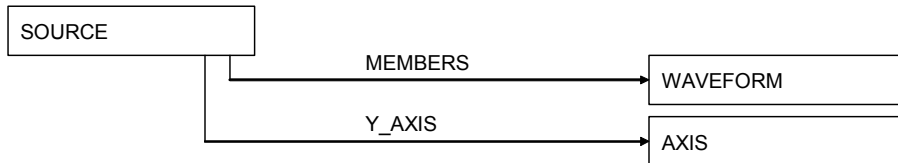
WRITE_AS_ONE RELATION describes relationships between VARIABLES, RECORDs, and VALUE_ARRAYs (see 7.23.4 and Figure 21).



IEC 1530/06

Figure 21 – WRITE_AS_ONE

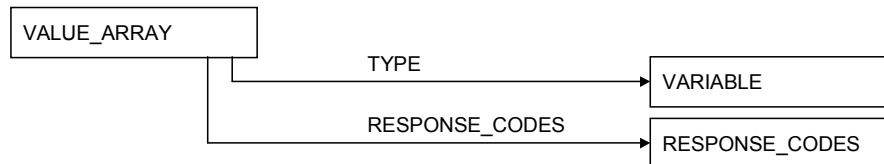
- **RESPONSE_CODES** specifies the application specific response codes for use when managing a VARIABLE, RECORD, VALUE_ARRAY, VARIABLE_LIST, PROGRAM, or DOMAIN (see 7.24).
- **SOURCE** (see 7.25 and Figure 22).



IEC 1531/06

Figure 22 – SOURCE

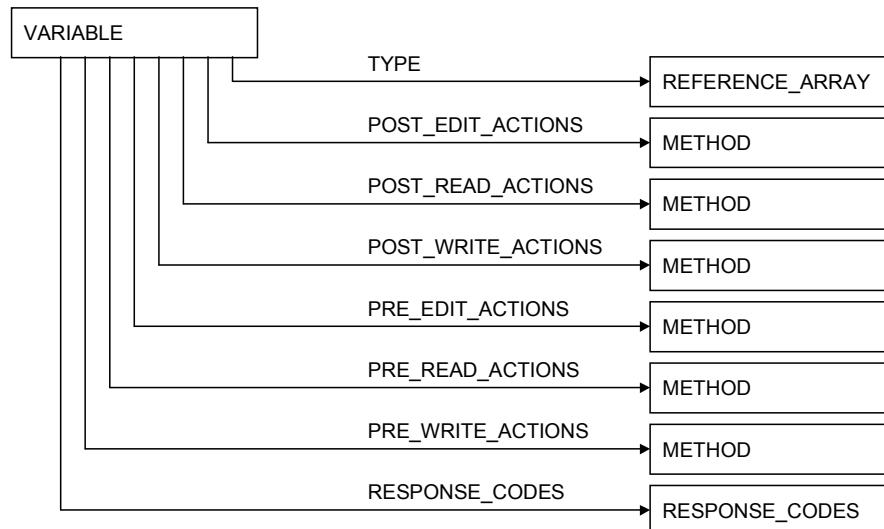
- **VALUE_ARRAY** describes logical groups of EDDL variables (see 7.26 and Figure 23).



IEC 1532/06

Figure 23 – VALUE_ARRAY

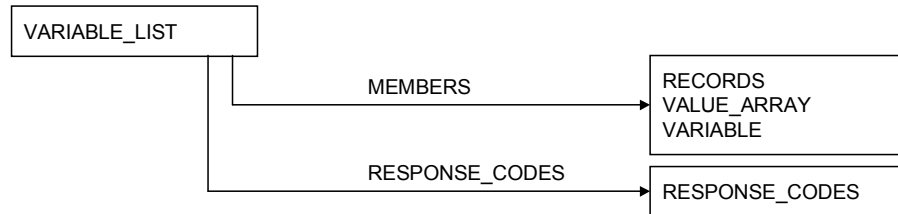
- **VARIABLE** describes the data contained in the device (see 7.27 and Figure 24).



IEC 1533/06

Figure 24 – VARIABLE

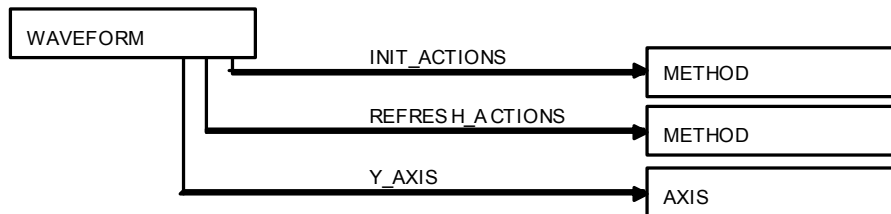
- **VARIABLE_LIST** describes logical groupings of data contained in the device that may be communicated as a list (see 7.28 and Figure 25).



IEC 1534/06

Figure 25 – VARIABLE_LIST

- **WAVEFORM** (see 7.29 and Figure 26).



IEC 1535/06

Figure 26 – WAVEFORM

7.1.5 Common attributes

The following common attributes are used in many basic constructs:

- DEFINITION
- MEMBERS
- LABEL
- HELP
- RESPONSE_CODES

7.1.6 Special elements

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

- **Output Redirection**
 - **OPEN** is used to create a separate output file for an EDD.
 - **CLOSE** is used to close an opened output file for an EDD.
- **Conditional Expressions** specify attributes values, which are dependent on the values of variables at run-time
- **Reference** is used throughout a device description to refer to other items within the same EDDL
- **Expression** is a logical or mathematical operation on variable values in order to modify attribute values of a basic construct at run-time

7.1.7 Rules for instances

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

7.1.8 Rules for list of VARIABLES

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

- BLOCK_A PARAMETERS
- COLLECTIONS

- RECORDS
- REFERENCE_ARRAY
- VALUE_ARRAY
- VARIABLE_LISTS

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

7.2 EDD identification information

7.2.1 General structure

Purpose

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

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

Lexical structure

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

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

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

7.2.2 Specific attributes

7.2.2.1 DD_REVISION

Purpose

This attribute specifies a unique code for the EDD revision of this device, as defined by the manufacturer.

NOTE 1 The manufacturer should change the revision number each time a new device description is issued for a device.

NOTE 2 There can be multiple EDDs for a particular revision of a particular device, in which case the DD_REVISION distinguishes them.

Lexical structure

DD_REVISION integer

The attribute of DD_REVISION is specified in Table 2.

Table 2 – DD_REVISION attribute

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