

BS EN 61804-3:2011



BSI Standards Publication

# Function Blocks (FB) for process control

Part 3: Electronic Device Description  
Language (EDDL)

**bsi.**

...making excellence a habit.  
**AVA** Documents™  
Document Sharing Hub

### National foreword

This British Standard is the UK implementation of EN 61804-3:2011. It is identical to IEC 61804-3:2010. It supersedes BS EN 61804-3:2007 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Industrial communications: process measurement and control, including fieldbus.

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

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

© BSI 2011

ISBN 978 0 580 70029 3

ICS 25.040.40; 35.240.50

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2011.

### Amendments issued since publication

| Amd. No. | Date | Text affected |
|----------|------|---------------|
|----------|------|---------------|

EUROPEAN STANDARD  
NORME EUROPÉENNE  
EUROPÄISCHE NORM

# EN 61804-3

March 2011

ICS 25.040.40; 35.240.50

Supersedes EN 61804-3:2007

English version

## Function Blocks (FB) for process control - Part 3: Electronic Device Description Language (EDDL) (IEC 61804-3:2010)

Blocs Fonctionnels (FB) pour le contrôle  
de processus industriel - Partie 3:  
Langage de description d'un équipement  
électronique (EDDL)  
(CEI 61804-3:2010)

Funktionsbausteine für die  
Prozessautomation -  
Teil 3: Elektronische  
Gerätebeschreibungssprache (EDDL)  
(IEC 61804-3:2010)

This European Standard was approved by CENELEC on 2011-01-03. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

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

# CENELEC

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

**Management Centre: Avenue Marnix 17, B - 1000 Brussels**

© 2011 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Ref. No. EN 61804-3:2011 E

 **Documents**  
Document Sharing Hub

## Foreword

The text of document 65E/162/FDIS, future edition 2 of IEC 61804-3, prepared by SC 65E, Devices and integration in enterprise systems, of IEC TC 65, Industrial-process measurement, control and automation, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 61804-3 on 2011-01-03.

This European Standard supersedes EN 61804-3:2007.

The main changes with respect to EN 61804-3:2007 are listed below <sup>1)</sup>.

### Additions

- Language elements to support modular devices, see 7.9, 7.10, 7.36.1, 7.36.2, 7.36.3, 7.36.13 and 7.36.15;
- Language elements to support offline configuration, 7.31;
- Usage of UTF-8, see A.2.3;
- Various BuiltIns;
- BLOCK\_A referencing. Updates to support access to multiple blocks, see 7.4.1, 7.38.18 up to 7.38.26;
- Add VALIDITY attribute to various lexical structures;
- Support of multi-language images, see 7.18.2.1;
- In 7.23.1 deleted the restriction of the MENU item list;
- Syntactical limitation on conditionals in 7.23.2.1 to restrict the MENU layout;
- Add LIST and delete VARIABLE\_LIST of the MENU items in 7.23.2.1;
- Additional return value data types for METHOD TYPE, see 7.24.2.3;
- Replace reference by a context specific specification in Table 170;
- Clarification on KEY\_POINTS behaviour, see 7.35.2.4;
- Add TRANSPARENT to the lexical structure to make the list of attributes consistent, see 7.36.11;
- Clarification on file behaviour, see 7.36.10;
- Add in Table A.5 the new key-words;
- Add the formal EDDL syntax in A.6.

### Corrections

- Deleted in A.6 all non-needed constructs using the auxiliary ...\_listR. These were created by a non-perfect syntax-checking tool.
- Deleted in A.6 all non-needed "stmt1:" and "stmt2:". These were created by a nonperfect syntax-checking tool.
- Made A.6 consistent about using a colon at the end of a term by amending colons in a consistent way.
- Deleted not used references.
- Deleted in several lexical structures the brackets and "<exp>".

---

<sup>1)</sup> The historical background to the EDDL specification is given in Annex E.

- Deleted in lexical structures the “[ ... <expr>]” where the conditional expression was not possible.
- Exchanged the attribute WIDTH by HEIGHT in the subclause specifying HEIGHT.
- Spelling errors like GUAGE exchanged to GAUGE.
- Syntactical limitation on conditionals in ACTIONS (for example see 7.14.2.3 and 7.14.2.4) to support conditionals only in the METHODS.
- Clarification on file behaviour in 7.15.
- Add GRID and IMAGES to the attribute list, see 7.19.
- Deleted a duplication of element list in 7.23.2.1.
- Defining and calling METHODS with parameter and return value, see 7.24.
- Restriction on METHOD CLASS, see 7.24.2.2.
- Clarification on SCALING\_FACTOR behaviour, see Table 165.
- Deleted EDDL operators from EDDL keyword list, see Table A.5.
- Clarification on TIME\_VALUE coding providing the absolute basis, see Table D.16 and Table D.17.

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

The following dates were fixed:

- |                                                                                                                                                |       |            |
|------------------------------------------------------------------------------------------------------------------------------------------------|-------|------------|
| – latest date by which the EN has to be implemented<br>at national level by publication of an identical<br>national standard or by endorsement | (dop) | 2011-10-03 |
| – latest date by which the national standards conflicting<br>with the EN have to be withdrawn                                                  | (dow) | 2014-01-03 |

Annex ZA has been added by CENELEC.

---

### Endorsement notice

The text of the International Standard IEC 61804-3:2010 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

|                |                                    |
|----------------|------------------------------------|
| IEC 61131-3    | NOTE Harmonized as EN 61131-3.     |
| IEC 61804-2    | NOTE Harmonized as EN 61804-2.     |
| IEC/TR 61804-4 | NOTE Harmonized as CLC/TR 61804-4. |

---

## **Annex ZA** (normative)

### **Normative references to international publications with their corresponding European publications**

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.

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

| <u>Publication</u> | <u>Year</u> | <u>Title</u>                                                                                                                              | <u>EN/HD</u>  | <u>Year</u> |
|--------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------|
| IEC 61499-1        | 2005        | Function blocks -<br>Part 1: Architecture                                                                                                 | EN 61499-1    | 2005        |
| IEC/TS 61804-1     | 2003        | Function blocks (FB) for process control -<br>Part 1: Overview of system aspects                                                          | -             | -           |
| ISO/IEC 2022       | -           | Information technology - Character code<br>structure and extension techniques                                                             | -             | -           |
| ISO/IEC 2375       | 2003        | Information technology - Procedure for<br>registration of escape sequences and coded<br>character sets                                    | -             | -           |
| ISO/IEC 7498-1     | -           | Information technology - Open Systems<br>Interconnection - Basic Reference Model: The<br>Basic Model                                      | -             | -           |
| ISO/IEC 8859-1     | 1998        | Information technology - 8-bit single-byte<br>coded graphic character sets -<br>Part 1: Latin alphabet No.1                               | -             | -           |
| ISO/IEC 9899       | -           | Programming languages - C                                                                                                                 | -             | -           |
| ISO/IEC 10646-1    | 2000        | Information technology - Universal Multiple-<br>Octet Coded Character Set (UCS) -<br>Part 1: Architecture and Basic Multilingual<br>Plane | -             | -           |
| ISO 639            | -           | Codes for the representation of names of<br>languages                                                                                     | -             | -           |
| ISO 3166-1         | 2006        | Codes for the representation of names of<br>countries and their subdivisions -<br>Part 1: Country codes                                   | EN ISO 3166-1 | 2006        |
| IEEE 754           | 1985        | Binary Floating-Point Arithmetic (R1990)                                                                                                  | -             | -           |
| RFC 3629           | 2003        | UTF-8, User Datagram Protocol, available at<br><a href="http://www.ietf.org/rfc/rfc0768.txt">http://www.ietf.org/rfc/rfc0768.txt</a>      | -             | -           |

## CONTENTS

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| INTRODUCTION .                                                            | 14  |
| 1 Scope .                                                                 | 15  |
| 2 Normative references .                                                  | 15  |
| 3 Terms, definitions, abbreviated terms and acronyms .                    | 16  |
| 3.1 Terms and definitions .                                               | 16  |
| 3.2 Abbreviated terms and acronyms .                                      | 21  |
| 4 Conformance statement .                                                 | 22  |
| 5 Conventions for lexical structures .                                    | 22  |
| 6 EDD and EDDL model .                                                    | 24  |
| 6.1 Overview of EDD and EDDL .                                            | 24  |
| 6.2 EDD architecture .                                                    | 24  |
| 6.3 Concepts of EDD .                                                     | 24  |
| 6.4 Principles of the EDD development process .                           | 24  |
| 6.5 Interrelations between the lexical structure and formal definitions . | 25  |
| 6.6 Builtins .                                                            | 25  |
| 6.7 Profiles .                                                            | 26  |
| 7 Electronic Device Description Language .                                | 26  |
| 7.1 Overview .                                                            | 26  |
| 7.2 EDD identification information .                                      | 34  |
| 7.3 AXIS .                                                                | 37  |
| 7.4 BLOCK .                                                               | 39  |
| 7.5 CHART .                                                               | 49  |
| 7.6 COLLECTION .                                                          | 51  |
| 7.7 COMMAND .                                                             | 52  |
| 7.8 COMPONENT .                                                           | 58  |
| 7.9 COMPONENT_FOLDER .                                                    | 62  |
| 7.10 COMPONENT_REFERENCE .                                                | 62  |
| 7.11 COMPONENT_RELATION .                                                 | 63  |
| 7.12 CONNECTION .                                                         | 65  |
| 7.13 DOMAIN .                                                             | 66  |
| 7.14 EDIT_DISPLAY .                                                       | 66  |
| 7.15 FILE .                                                               | 69  |
| 7.16 GRAPH .                                                              | 69  |
| 7.17 GRID .                                                               | 70  |
| 7.18 IMAGE .                                                              | 71  |
| 7.19 IMPORT .                                                             | 72  |
| 7.20 INTERFACE .                                                          | 87  |
| 7.21 LIKE .                                                               | 87  |
| 7.22 LIST .                                                               | 88  |
| 7.23 MENU .                                                               | 89  |
| 7.24 METHOD .                                                             | 96  |
| 7.25 PROGRAM .                                                            | 98  |
| 7.26 RECORD .                                                             | 99  |
| 7.27 REFERENCE_ARRAY .                                                    | 99  |
| 7.28 Relations .                                                          | 100 |

|                                                                                          |     |
|------------------------------------------------------------------------------------------|-----|
| 7.29 RESPONSE_CODES.....                                                                 | 101 |
| 7.30 SOURCE.....                                                                         | 102 |
| 7.31 TEMPLATE.....                                                                       | 103 |
| 7.32 VALUE_ARRAY.....                                                                    | 104 |
| 7.33 VARIABLE.....                                                                       | 105 |
| 7.34 VARIABLE_LIST.....                                                                  | 121 |
| 7.35 WAVEFORM.....                                                                       | 121 |
| 7.36 Common attributes.....                                                              | 127 |
| 7.37 Conditional expression.....                                                         | 134 |
| 7.38 Referencing.....                                                                    | 135 |
| 7.39 Strings.....                                                                        | 144 |
| 7.40 Expression.....                                                                     | 146 |
| 7.41 Text dictionary.....                                                                | 150 |
| Annex A (normative) EDDL formal definition.....                                          | 152 |
| Annex B (normative) EDDL Builtin library.....                                            | 238 |
| Annex C (informative) EDD example.....                                                   | 388 |
| Annex D (normative) Profiles of EDDL and Builtins.....                                   | 402 |
| Annex E (informative) Historical background.....                                         | 434 |
| Bibliography.....                                                                        | 435 |
| Figure 1 – Position of the IEC 61804 series related to other standards and products..... | 14  |
| Figure 2 – EDD generation process.....                                                   | 25  |
| Figure 3 – BLOCK_A.....                                                                  | 27  |
| Figure 4 – CHART.....                                                                    | 27  |
| Figure 5 – COLLECTION.....                                                               | 28  |
| Figure 6 – COMMAND.....                                                                  | 28  |
| Figure 7 – DOMAIN.....                                                                   | 28  |
| Figure 8 – EDIT_DISPLAY.....                                                             | 29  |
| Figure 9 – FILE.....                                                                     | 29  |
| Figure 10 – GRAPH.....                                                                   | 29  |
| Figure 11 – GRID.....                                                                    | 29  |
| Figure 12 – IMAGE.....                                                                   | 29  |
| Figure 13 – LIKE.....                                                                    | 30  |
| Figure 14 – LIST.....                                                                    | 30  |
| Figure 15 – MENU.....                                                                    | 30  |
| Figure 16 – PROGRAM.....                                                                 | 31  |
| Figure 17 – RECORD.....                                                                  | 31  |
| Figure 18 – REFERENCE_ARRAY.....                                                         | 31  |
| Figure 19 – REFRESH.....                                                                 | 32  |
| Figure 20 – UNIT.....                                                                    | 32  |
| Figure 21 – WRITE_AS_ONE.....                                                            | 32  |
| Figure 22 – SOURCE.....                                                                  | 32  |
| Figure 23 – VALUE_ARRAY.....                                                             | 32  |
| Figure 24 – VARIABLE.....                                                                | 33  |
| Figure 25 – VARIABLE_LIST.....                                                           | 33  |

|                                                            |     |
|------------------------------------------------------------|-----|
| Figure 26 – WAVEFORM. ....                                 | 33  |
| Figure 27 – EDDL import mechanisms. ....                   | 73  |
| Figure 28 – MENU activation. ....                          | 96  |
| Figure 29 – Time for read-and-write operation. ....        | 120 |
| Figure C.1 – Example of an operator screen using EDD. .... | 388 |
|                                                            |     |
| Table 1 – Field attribute descriptions. ....               | 23  |
| Table 2 – DD_REVISION attribute. ....                      | 35  |
| Table 3 – DEVICE_REVISION attribute. ....                  | 35  |
| Table 4 – DEVICE_TYPE attribute. ....                      | 35  |
| Table 5 – EDD_PROFILE attribute. ....                      | 36  |
| Table 6 – EDD_VERSION attribute. ....                      | 36  |
| Table 7 – MANUFACTURER attribute. ....                     | 36  |
| Table 8 – MANUFACTURER_EXT attribute. ....                 | 37  |
| Table 9 – AXIS attributes. ....                            | 37  |
| Table 10 – MAX_VALUE, MIN_VALUE attribute. ....            | 38  |
| Table 11 – SCALING attribute. ....                         | 38  |
| Table 12 – BLOCK_A attributes. ....                        | 39  |
| Table 13 – CHARACTERISTIC attribute. ....                  | 40  |
| Table 14 – PARAMETER attributes. ....                      | 40  |
| Table 15 – AXIS_ITEMS attribute. ....                      | 41  |
| Table 16 – CHART_ITEMS attribute. ....                     | 41  |
| Table 17 – COLLECTION_ITEMS attribute. ....                | 41  |
| Table 18 – EDIT_DISPLAY_ITEMS attribute. ....              | 41  |
| Table 19 – FILE_ITEMS attribute. ....                      | 42  |
| Table 20 – GRAPH_ITEMS attribute. ....                     | 42  |
| Table 21 – GRID_ITEMS attribute. ....                      | 42  |
| Table 22 – IMAGE_ITEMS attribute. ....                     | 42  |
| Table 23 – LIST_ITEMS attribute. ....                      | 43  |
| Table 24 – MENU_ITEMS attribute. ....                      | 43  |
| Table 25 – METHOD_ITEMS attribute. ....                    | 43  |
| Table 26 – PARAMETER_LISTS attributes. ....                | 44  |
| Table 27 – REFERENCE_ARRAY_ITEMS attribute. ....           | 44  |
| Table 28 – REFRESH_ITEMS attribute. ....                   | 44  |
| Table 29 – SOURCE_ITEMS attribute. ....                    | 44  |
| Table 30 – UNIT_ITEMS attribute. ....                      | 45  |
| Table 31 – WAVEFORM_ITEMS attribute. ....                  | 45  |
| Table 32 – WRITE_AS_ONE_ITEMS attribute. ....              | 45  |
| Table 33 – CHARTS attribute. ....                          | 46  |
| Table 34 – LISTS attribute. ....                           | 46  |
| Table 35 – GRAPHS attribute. ....                          | 46  |
| Table 36 – GRIDS attribute. ....                           | 47  |
| Table 37 – MENUS attribute. ....                           | 47  |

|                                                    |    |
|----------------------------------------------------|----|
| Table 38 – METHODS attribute . . . . .             | 47 |
| Table 39 – BLOCK_B attributes . . . . .            | 48 |
| Table 40 – NUMBER attributes . . . . .             | 48 |
| Table 41 – TYPE attributes . . . . .               | 49 |
| Table 42 – CHART attributes . . . . .              | 49 |
| Table 43 – CYCLE_TIME attribute . . . . .          | 50 |
| Table 44 – LENGTH attribute . . . . .              | 50 |
| Table 45 – TYPE attribute . . . . .                | 50 |
| Table 46 – COLLECTION attributes . . . . .         | 51 |
| Table 47 – item-type . . . . .                     | 51 |
| Table 48 – COMMAND attributes . . . . .            | 52 |
| Table 49 – OPERATION attribute . . . . .           | 53 |
| Table 50 – TRANSACTION attributes . . . . .        | 54 |
| Table 51 – REPLY and REQUEST attributes . . . . .  | 54 |
| Table 52 – INDEX attribute . . . . .               | 55 |
| Table 53 – BLOCK_B attribute . . . . .             | 56 |
| Table 54 – NUMBER attribute . . . . .              | 56 |
| Table 55 – SLOT attribute . . . . .                | 56 |
| Table 56 – SUB_SLOT attribute . . . . .            | 57 |
| Table 57 – CONNECTION attribute . . . . .          | 57 |
| Table 58 – HEADER attribute . . . . .              | 57 |
| Table 59 – MODULE attribute . . . . .              | 57 |
| Table 60 – COMPONENT attribute . . . . .           | 58 |
| Table 61 – CAN_DELETE attribute . . . . .          | 59 |
| Table 62 – CHECK_CONFIGURATION attribute . . . . . | 59 |
| Table 63 – COMPONENT_RELATIONS attribute . . . . . | 59 |
| Table 64 – DECLARATION attribute . . . . .         | 60 |
| Table 65 – DETECT attribute . . . . .              | 60 |
| Table 66 – EDD attribute . . . . .                 | 60 |
| Table 67 – CHECK_CONFIGURATION attribute . . . . . | 61 |
| Table 68 – REDUNDANCY attribute . . . . .          | 61 |
| Table 69 – SCAN attribute . . . . .                | 61 |
| Table 70 – SCAN_LIST attribute . . . . .           | 61 |
| Table 71 – COMPONENT_FOLDER attribute . . . . .    | 62 |
| Table 72 – COMPONENT_REFERENCE attribute . . . . . | 62 |
| Table 73 – COMPONENT_RELATION attribute . . . . .  | 63 |
| Table 74 – COMPONENTS attribute . . . . .          | 63 |
| Table 75 – RELATION_TYPE attribute . . . . .       | 64 |
| Table 76 – ADDRESSING attribute . . . . .          | 64 |
| Table 77 – MAXIMUM_NUMBER attribute . . . . .      | 65 |
| Table 78 – MINIMUM_NUMBER attribute . . . . .      | 65 |
| Table 79 – REQUIRED_INTERFACE attribute . . . . .  | 65 |
| Table 80 – CONNECTION attribute . . . . .          | 66 |

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| Table 81 – APPINSTANCE attribute . . . . .                                  | 66 |
| Table 82 – DOMAIN attributes . . . . .                                      | 66 |
| Table 83 – EDIT_DISPLAY attributes . . . . .                                | 67 |
| Table 84 – EDIT_ITEMS attribute . . . . .                                   | 67 |
| Table 85 – DISPLAY_ITEM attributes . . . . .                                | 68 |
| Table 86 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS attribute . . . . .          | 68 |
| Table 87 – FILE attributes . . . . .                                        | 69 |
| Table 88 – GRAPH attributes . . . . .                                       | 69 |
| Table 89 – CYCLE_TIME attribute . . . . .                                   | 70 |
| Table 90 – X_AXIS attribute . . . . .                                       | 70 |
| Table 91 – GRID attributes . . . . .                                        | 70 |
| Table 92 – VECTORS attribute . . . . .                                      | 71 |
| Table 93 – ORIENTATION attribute . . . . .                                  | 71 |
| Table 94 – IMAGE attributes . . . . .                                       | 72 |
| Table 95 – PATH attribute . . . . .                                         | 72 |
| Table 96 – LINK attribute . . . . .                                         | 72 |
| Table 97 – Importing Device Description . . . . .                           | 73 |
| Table 98 – Redefinition attributes . . . . .                                | 75 |
| Table 99 – Redefinition rules for AXIS attributes . . . . .                 | 75 |
| Table 100 – Redefinition rules for BLOCK_A attributes . . . . .             | 75 |
| Table 101 – Redefinition rules for BLOCK_B attributes . . . . .             | 76 |
| Table 102 – Redefinition rules for CHART attributes . . . . .               | 77 |
| Table 103 – Redefinition rules for COLLECTION attributes . . . . .          | 77 |
| Table 104 – Redefinition rules for COMMAND attributes . . . . .             | 77 |
| Table 105 – Redefinition rules for COMPONENT attributes . . . . .           | 78 |
| Table 106 – Redefinition rules for COMPONENT_FOLDER attributes . . . . .    | 78 |
| Table 107 – Redefinition rules for COMPONENT_REFERENCE attributes . . . . . | 79 |
| Table 108 – Redefinition rules for COMPONENT_RELATION attributes . . . . .  | 79 |
| Table 109 – Redefinition rules for CONNECTION attributes . . . . .          | 79 |
| Table 110 – Redefinition rules for DOMAIN attributes . . . . .              | 80 |
| Table 111 – Redefinition rules for EDIT_DISPLAY attributes . . . . .        | 80 |
| Table 112 – Redefinition rules for FILE attributes . . . . .                | 80 |
| Table 113 – Redefinition rules for GRAPH attributes . . . . .               | 81 |
| Table 114 – Redefinition rules for GRID attributes . . . . .                | 81 |
| Table 115 – Redefinition rules for IMAGE attributes . . . . .               | 81 |
| Table 116 – Redefinition rules for INTERFACE attributes . . . . .           | 82 |
| Table 117 – Redefinition rules for LIST attributes . . . . .                | 82 |
| Table 118 – Redefinition rules for MENU attributes . . . . .                | 82 |
| Table 119 – Redefinition rules for METHOD attributes . . . . .              | 83 |
| Table 120 – Redefinition rules for PROGRAM attributes . . . . .             | 83 |
| Table 121 – Redefinition rules for RECORD attributes . . . . .              | 83 |
| Table 122 – Redefinition rules for REFERENCE_ARRAY attributes . . . . .     | 83 |
| Table 123 – Redefinition rules for RESPONSE_CODES attributes . . . . .      | 84 |

|                                                                                                                                             |     |
|---------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Table 124 – Redefinition rules for SOURCE attributes .                                                                                      | 84  |
| Table 125 – Redefinition rules for TEMPLATE attributes .                                                                                    | 84  |
| Table 126 – Redefinition rules for VALUE_ARRAY attributes .                                                                                 | 85  |
| Table 127 – Redefinition rules for VARIABLE attributes .                                                                                    | 85  |
| Table 128 – Redefinition rules for VARIABLE_LIST attributes .                                                                               | 86  |
| Table 129 – Redefinition rules for WAVEFORM attributes .                                                                                    | 86  |
| Table 130 – INTERFACE attribute .                                                                                                           | 87  |
| Table 131 – DECLARATION attribute .                                                                                                         | 87  |
| Table 132 – LIKE attributes .                                                                                                               | 88  |
| Table 133 – LIST attributes .                                                                                                               | 88  |
| Table 134 – TYPE attribute .                                                                                                                | 88  |
| Table 135 – CAPACITY, COUNT attribute .                                                                                                     | 89  |
| Table 136 – MENU attribute .                                                                                                                | 89  |
| Table 137 – ITEMS attribute .                                                                                                               | 90  |
| Table 138 – ACCESS attribute .                                                                                                              | 91  |
| Table 139 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS, POST_READ_ACTIONS,<br>PRE_READ_ACTIONS, POST_WRITE_ACTIONS, PRE_WRITE_ACTIONS attributes . | 91  |
| Table 140 – PURPOSE attribute .                                                                                                             | 93  |
| Table 141 – ROLE attribute .                                                                                                                | 94  |
| Table 142 – STYLE attribute .                                                                                                               | 94  |
| Table 143 – METHOD attributes .                                                                                                             | 97  |
| Table 144 – ACCESS attributes .                                                                                                             | 97  |
| Table 145 – TYPE attributes .                                                                                                               | 98  |
| Table 146 – PROGRAM attributes .                                                                                                            | 98  |
| Table 147 – ARGUMENTS attribute .                                                                                                           | 99  |
| Table 148 – RECORD attributes .                                                                                                             | 99  |
| Table 149 – REFERENCE_ARRAY attribute .                                                                                                     | 99  |
| Table 150 – ELEMENTS attribute .                                                                                                            | 100 |
| Table 151 – REFRESH attributes .                                                                                                            | 100 |
| Table 152 – UNIT attributes .                                                                                                               | 101 |
| Table 153 – WRITE_AS_ONE attribute .                                                                                                        | 101 |
| Table 154 – RESPONSE_CODES attributes .                                                                                                     | 101 |
| Table 155 – SOURCE attributes .                                                                                                             | 102 |
| Table 156 – Y_AXIS attribute .                                                                                                              | 103 |
| Table 157 – TEMPLATE attributes .                                                                                                           | 104 |
| Table 158 – DEFAULT_VALUES attribute .                                                                                                      | 104 |
| Table 159 – VALUE_ARRAY attributes .                                                                                                        | 104 |
| Table 160 – NUMBER_OF_ELEMENTS attribute .                                                                                                  | 105 |
| Table 161 – TYPE attribute .                                                                                                                | 105 |
| Table 162 – VARIABLE attributes .                                                                                                           | 105 |
| Table 163 – CLASS attributes .                                                                                                              | 106 |
| Table 164 – TYPE attributes .                                                                                                               | 107 |
| Table 165 – DOUBLE, FLOAT, INTEGER, UNSIGNED_INTEGER attributes .                                                                           | 109 |

|                                                                                                                                                                 |     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Table 166 – DATE, DATE_AND_TIME, DURATION, TIME, TIME_VALUE attributes .                                                                                        | 111 |
| Table 167 – BIT_ENUMERATED attributes .                                                                                                                         | 112 |
| Table 168 – status–class attributes .                                                                                                                           | 113 |
| Table 169 – ALL, AO, DV, TV attributes .                                                                                                                        | 114 |
| Table 170 – Enumerated types attributes .                                                                                                                       | 114 |
| Table 171 – Index type attributes .                                                                                                                             | 115 |
| Table 172 – Object reference type attribute .                                                                                                                   | 115 |
| Table 173 – DEFAULT_REFERENCE attributes .                                                                                                                      | 115 |
| Table 174 – String types attributes .                                                                                                                           | 117 |
| Table 175 – CONSTANT_UNIT attribute .                                                                                                                           | 117 |
| Table 176 – DEFAULT_VALUE attribute .                                                                                                                           | 118 |
| Table 177 – INITIAL_VALUE attribute .                                                                                                                           | 118 |
| Table 178 – POST_EDIT_ACTIONS, PRE_EDIT_ACTIONS, POST_READ_ACTIONS,<br>PRE_READ_ACTIONS, POST_WRITE_ACTIONS, PRE_WRITE_ACTIONS,<br>REFRESH_ACTIONS attributes . | 118 |
| Table 179 – READ/WRITE_TIMEOUT attributes .                                                                                                                     | 120 |
| Table 180 – STYLE attribute .                                                                                                                                   | 121 |
| Table 181 – VARIABLE_LIST attributes .                                                                                                                          | 121 |
| Table 182 – WAVEFORM attributes .                                                                                                                               | 121 |
| Table 183 – TYPE attribute .                                                                                                                                    | 122 |
| Table 184 – XY attribute .                                                                                                                                      | 122 |
| Table 185 – YT attribute .                                                                                                                                      | 123 |
| Table 186 – HORIZONTAL attribute .                                                                                                                              | 124 |
| Table 187 – VERTICAL attribute .                                                                                                                                | 124 |
| Table 188 – EXIT_ACTIONS, INIT_ACTIONS, REFRESH_ACTIONS attribute .                                                                                             | 125 |
| Table 189 – KEY_POINTS attribute .                                                                                                                              | 125 |
| Table 190 – X_VALUES, Y_VALUES attribute .                                                                                                                      | 126 |
| Table 191 – Y_AXIS attribute .                                                                                                                                  | 126 |
| Table 192 – CLASSIFICATION attribute .                                                                                                                          | 127 |
| Table 193 – COMPONENT_PARENT attribute .                                                                                                                        | 128 |
| Table 194 – COMPONENT_PATH attribute .                                                                                                                          | 128 |
| Table 195 – DEFINITION attributes .                                                                                                                             | 129 |
| Table 196 – EMPHASIS attribute .                                                                                                                                | 129 |
| Table 197 – HANDLING attribute .                                                                                                                                | 130 |
| Table 198 – HEIGHT/WIDTH attribute .                                                                                                                            | 130 |
| Table 199 – HELP attribute .                                                                                                                                    | 130 |
| Table 200 – LABEL attribute .                                                                                                                                   | 131 |
| Table 201 – LINE_COLOR attribute .                                                                                                                              | 131 |
| Table 202 – LINE_TYPE attribute .                                                                                                                               | 131 |
| Table 203 – MEMBERS attributes .                                                                                                                                | 132 |
| Table 204 – PROTOCOL attribute .                                                                                                                                | 133 |
| Table 205 – RESPONSE_CODES attribute .                                                                                                                          | 133 |
| Table 206 – SUPPLIED_INTERFACE attribute .                                                                                                                      | 133 |
| Table 207 – VALIDITY attributes .                                                                                                                               | 134 |

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| Table 208 – IF, SELECT conditional .                                    | 135 |
| Table 209 – Referencing an EDD instance .                               | 135 |
| Table 210 – Referencing elements of VARIABLE .                          | 136 |
| Table 211 – Referencing elements of RECORD .                            | 136 |
| Table 212 – Referencing elements of VALUE_ARRAY .                       | 136 |
| Table 213 – Referencing members of COLLECTION .                         | 137 |
| Table 214 – Referencing members of REFERENCE_ARRAY .                    | 137 |
| Table 215 – Referencing members of VARIABLE_LISTS .                     | 137 |
| Table 216 – Referencing members of a BLOCK_A PARAMETERS .               | 137 |
| Table 217 – Referencing members of BLOCK_A PARAMETER_LISTS .            | 138 |
| Table 218 – Referencing members of BLOCK_A LOCAL_PARAMETER .            | 138 |
| Table 219 – Referencing BLOCK_A CHARACTERISTICS .                       | 138 |
| Table 220 – Referencing members of FILE .                               | 139 |
| Table 221 – Referencing elements of LIST .                              | 139 |
| Table 222 – Referencing members of CHART .                              | 139 |
| Table 223 – Referencing members of GRAPH .                              | 139 |
| Table 224 – Referencing members of SOURCE .                             | 140 |
| Table 225 – Referencing AXIS of a GRAPH. SOURCE, WAVEFORM .             | 140 |
| Table 226 – Referencing PARAMETERS of specific BLOCK_A instance .       | 140 |
| Table 227 – Referencing LOCAL_PARAMETERS of specific BLOCK_A instance . | 141 |
| Table 228 – Referencing CHARACTERISTICS of specific BLOCK_A instance .  | 141 |
| Table 229 – Referencing CHARTS of specific BLOCK_A instance .           | 141 |
| Table 230 – Referencing LISTS of specific BLOCK_A instance .            | 142 |
| Table 231 – Referencing GRAPHS of specific BLOCK_A instance .           | 142 |
| Table 232 – Referencing GRIDS of specific BLOCK_A instance .            | 142 |
| Table 233 – Referencing MENUS of specific BLOCK_A instance .            | 143 |
| Table 234 – Referencing METHODS of specific BLOCK_A instance .          | 143 |
| Table 235 – Referencing a COMPONENT instance .                          | 143 |
| Table 236 – Referencing a COMPONENT type .                              | 144 |
| Table 237 – string as a string literal .                                | 144 |
| Table 238 – string as a string variable .                               | 144 |
| Table 239 – String as an enumeration value .                            | 145 |
| Table 240 – String as a dictionary reference .                          | 145 |
| Table 241 – Referencing HELP and LABEL attributes of EDD instances .    | 145 |
| Table 242 – String operation .                                          | 146 |
| Table 243 – Format specifier .                                          | 146 |
| Table 244 – Primary expressions .                                       | 146 |
| Table 245 – Attribute values of VARIABLES .                             | 147 |
| Table 246 – AXIS Attribute Values .                                     | 147 |
| Table 247 – LIST Attribute Values .                                     | 148 |
| Table 248 – Unary expressions .                                         | 148 |
| Table 249 – Multiplicative operators .                                  | 148 |
| Table 250 – Additive operators .                                        | 149 |

|                                              |     |
|----------------------------------------------|-----|
| Table 251 – Shift operators .....            | 149 |
| Table 252 – Relational operators .....       | 149 |
| Table 253 – Equality operators .....         | 150 |
| Table 254 – Text dictionary attributes ..... | 151 |

## INTRODUCTION

This part of the IEC 61804 series specifies the Electronic Device Description Language (EDDL) technology.

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.

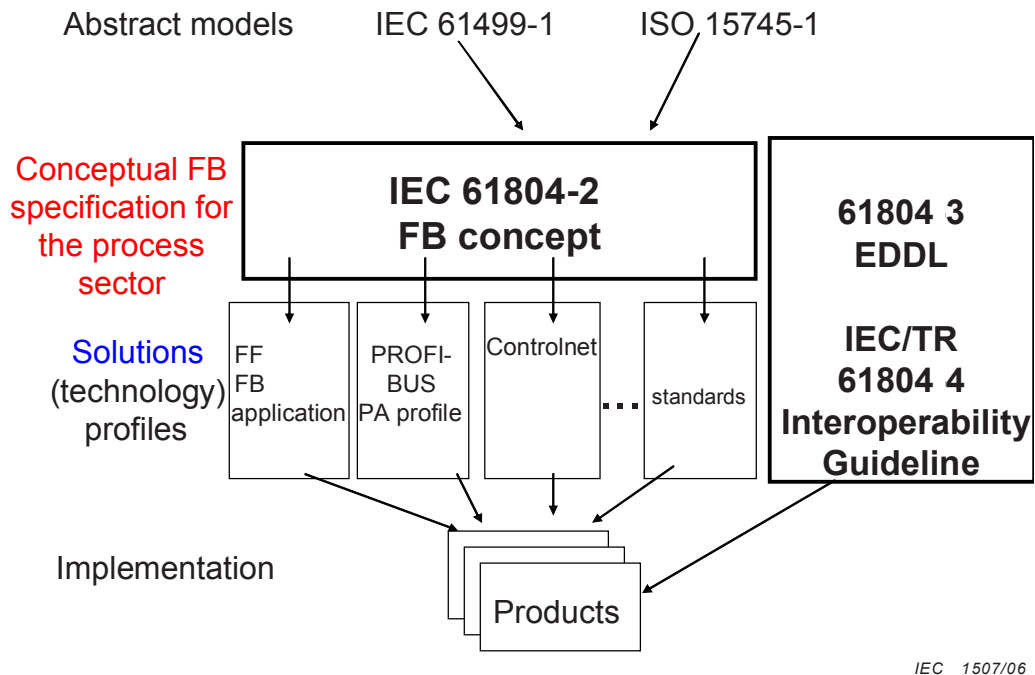


Figure 1 – Position of the IEC 61804 series related to other standards and products

## 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/TS 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/TS 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, *Code for the representation of names of languages*

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

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

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

### 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.

##### 3.1.1

##### **algorithm**

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

##### 3.1.2

##### **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.

##### 3.1.3

##### **Builtin**

predefined subroutine for communication and display executed by the EDD application

##### 3.1.4

##### **configuration** (of a system or device)

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

[IEC 61499-1]

##### 3.1.5

##### **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]

##### 3.1.6

##### **data connection**

association established between functional units for conveyance of data

[IEC 61499-1]

**3.1.7****data input**

interface of an FB which receives data from a data connection

[IEC 61499-1]

**3.1.8****data output**

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

[IEC 61499-1]

**3.1.9****data type**

set of values together with a set of permitted operations

[ISO 2382 series]

**3.1.10****device**

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

[IEC 61499-1]

**3.1.11****EDD application**

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

**3.1.12****EDDL processor**

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

**3.1.13****EDDL profile**

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

**3.1.14****Electronic Device Description Language****EDDL**

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

**3.1.15****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).

**3.1.16****Electronic Device Description Source****EDDS**

ASCII file containing a specific device description

**3.1.17****Electronic Device Description Technology****EDDT**

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

**3.1.18****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

**3.1.19****Electronic Device Description Compiler**

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

**3.1.20****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

**3.1.21****entity**

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

[IEC 61499-1]

**3.1.22****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.

**3.1.23****exception**

event that causes suspension of normal execution

[IEC 61499-1]

**3.1.24****function**

specific purpose of an entity or its characteristic action

[IEC 61499-1]

**3.1.25****functional unit**

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

[ISO/AFNOR Dictionary of Computer Science]

**3.1.26****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.

**3.1.27****hardware**

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

[ISO/AFNOR Dictionary of Computer Science]

**3.1.28****implementation**

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

[IEC 61499-1]

**3.1.29****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]

**3.1.30****instance**

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

[IEC 61499-1]

**3.1.31****instance name**

identifier associated with, and designating, an instance

[IEC 61499-1]

**3.1.32****instantiation**

creation of an instance of a specified type

[IEC 61499-1]

**3.1.33****interface**

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

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

**3.1.34****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]

**3.1.35****invocation**

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

[IEC 61499-1]

**3.1.36****mapping**

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

[ISO/AFNOR Dictionary of Computer Science]

### 3.1.37

#### **model**

representation of a real world process, device, or concept

[IEC 61499-1]

### 3.1.38

#### **operation**

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

[ISO/AFNOR Dictionary of Computer Science]

### 3.1.39

#### **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]

### 3.1.40

#### **parameter**

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

[ISO/AFNOR Dictionary of Computer Science]

### 3.1.41

#### **preprocessor**

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

### 3.1.42

#### **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

### 3.1.43

#### **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.

### 3.1.44

#### **resource management application**

application whose primary function is the management of a single resource

[IEC 61499-1]

### 3.1.45

#### **service**

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

[IEC 61499-1]

### 3.1.46

#### **software**

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

[IEC 61499-1]

### 3.1.47

#### **system**

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

[IEC 60050-351:IEV 351-21-20:2006, modified]

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.

### 3.1.48

#### **text dictionary**

collection of multilingual or other texts within the EDD

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

### 3.1.49

#### **type**

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

[IEC 61499-1]

### 3.1.50

#### **type name**

identifier associated with, and designating, a type

[IEC 61499-1]

### 3.1.51

#### **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]

## 3.2 Abbreviated terms and acronyms

|        |                                                                                           |
|--------|-------------------------------------------------------------------------------------------|
| ANSI   | American National Standard Institute                                                      |
| ANSI C | American National Standard Institute for the programming language C<br>(see ISO/IEC 9899) |
| ASCII  | American Standard Code for Information Interchange (see ISO/IEC 10646-1)                  |
| ASN.1  | Abstract Lexical Structure Notation 1                                                     |
| CP     | Communication Profile                                                                     |
| CPF    | Communication Profile Family                                                              |
| 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                                 |
| FIFO | First In First Out                             |
| HCF  | HART® <sup>2</sup> Communication Foundation    |
| HMI  | Human Machine Interface                        |
| HTML | Hypertext Mark-up Language                     |
| I/O  | Input/Output                                   |
| ID   | Identifier                                     |
| OSI  | Open Systems Interconnection                   |
| PI   | PROFIBUS & PROFINET <sup>3</sup> International |
| PNO  | PROFIBUS Nutzerorganisation                    |
| wao  | Write as one                                   |

#### 4 Conformance statement

A statement of conformance to a profile of this International Standard shall be stated<sup>4</sup> as either

- compliance to IEC 61804-3:2010 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**

<sup>2</sup> 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.

<sup>3</sup> 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.

<sup>4</sup> In accordance with the ISO/IEC Directives.

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.                                                                                                                                          |
| 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 7.37). 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.

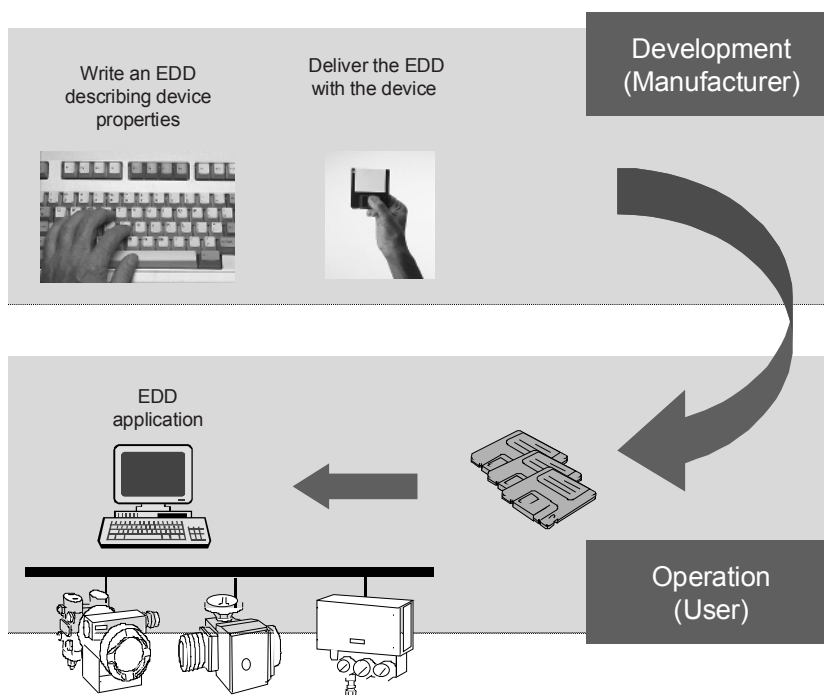
### 6.4 Principles of the EDD development process

#### 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 7.41).

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

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

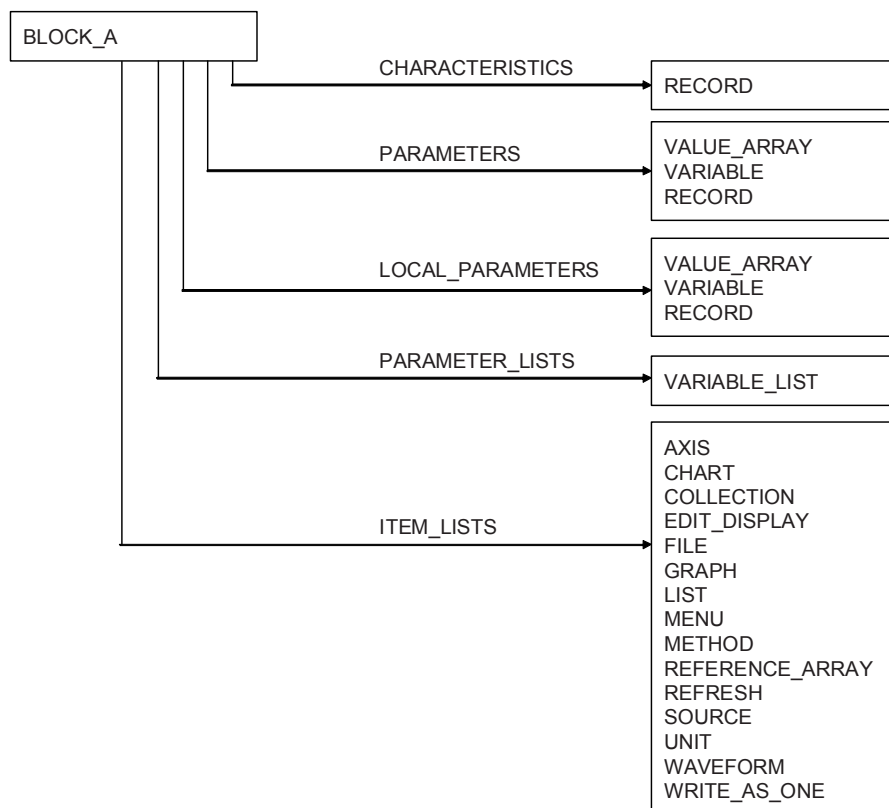
Some constructs have similar names and functions while differing in their detailed specification. This additional variety has been included to ensure compatibility with several existing description languages. Appropriate mapping cross-references are given by profiles.

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

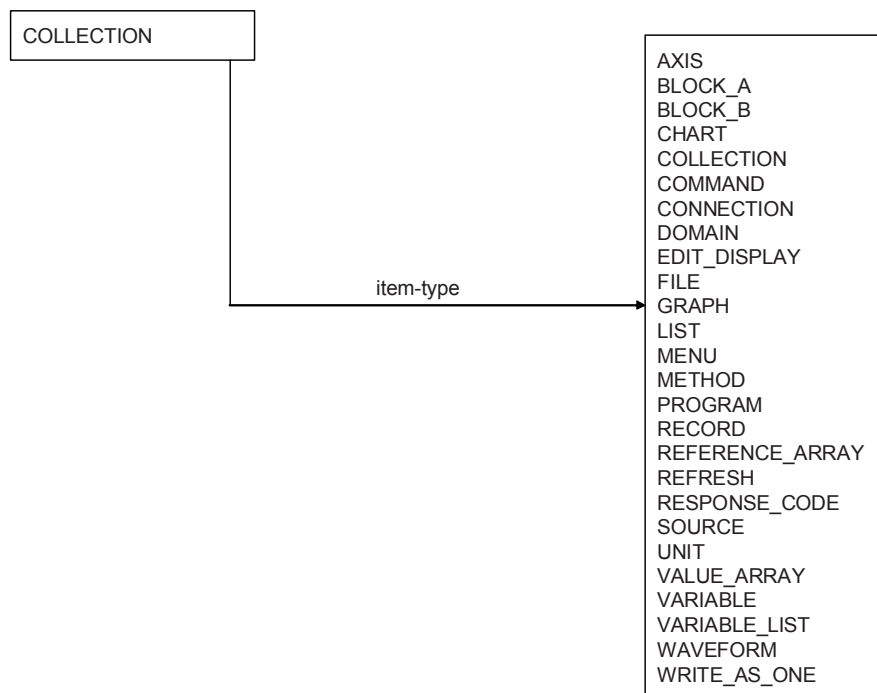
- **BLOCK\_B** is used to structure variables in instances of logical block types. For accessing a VARIABLE within a BLOCK\_B construct, the basic construct COMMAND is used to provide the relative addressing (see 7.4.2).
- **CHART** describes a chart used to display data from a device in the EDD application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



IEC 1513/06

Figure 4 – CHART

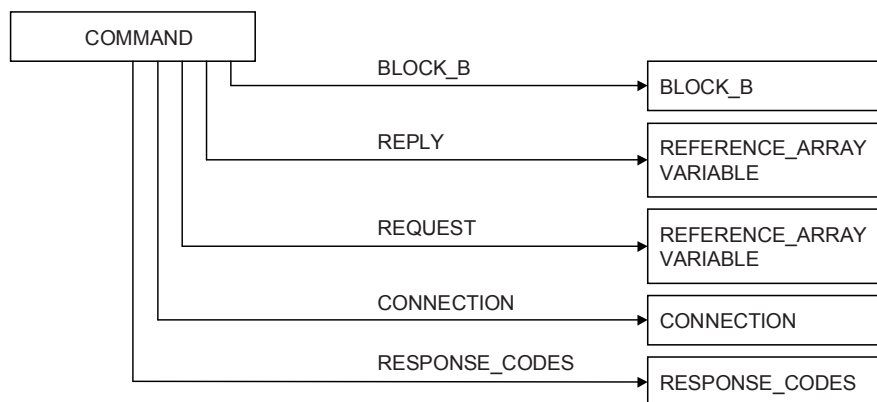
- **COLLECTION** describes logical groupings of data (see 7.6 and Figure 5).



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 7.13 and Figure 7).



IEC 1516/06

Figure 7 – DOMAIN

- **EDIT\_DISPLAY** describes how the data will be represented to a user by a display device (see 7.14 and Figure 8).

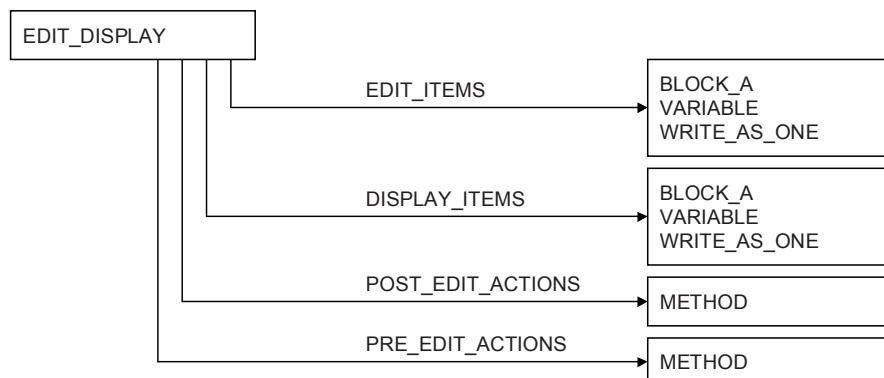


Figure 8 – EDIT\_DISPLAY

IEC 1517/06

- **FILE** describes an area of persistent storage (see 7.15 and Figure 9).

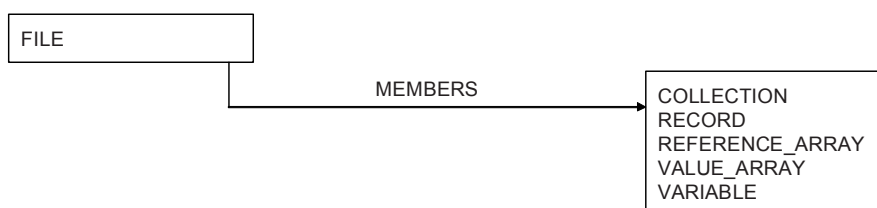


Figure 9 – FILE

IEC 1518/06

- **GRAPH** describes a graph used to display data from a device in the EDD application (see 7.16 and Figure 10).

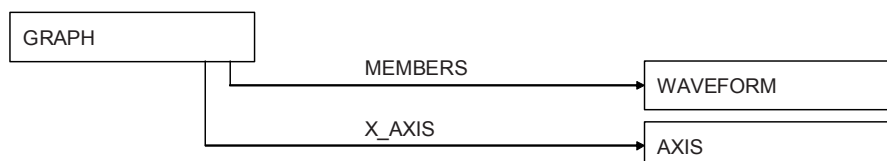
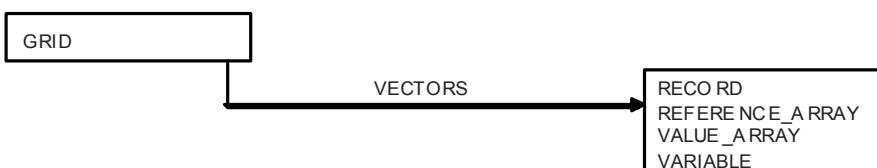


Figure 10 – GRAPH

IEC 1519/06

- **GRID** describes a matrix used to display data from a device in the EDD application (see 7.17 and Figure 11).



IEC 1520/06

Figure 11 – GRID

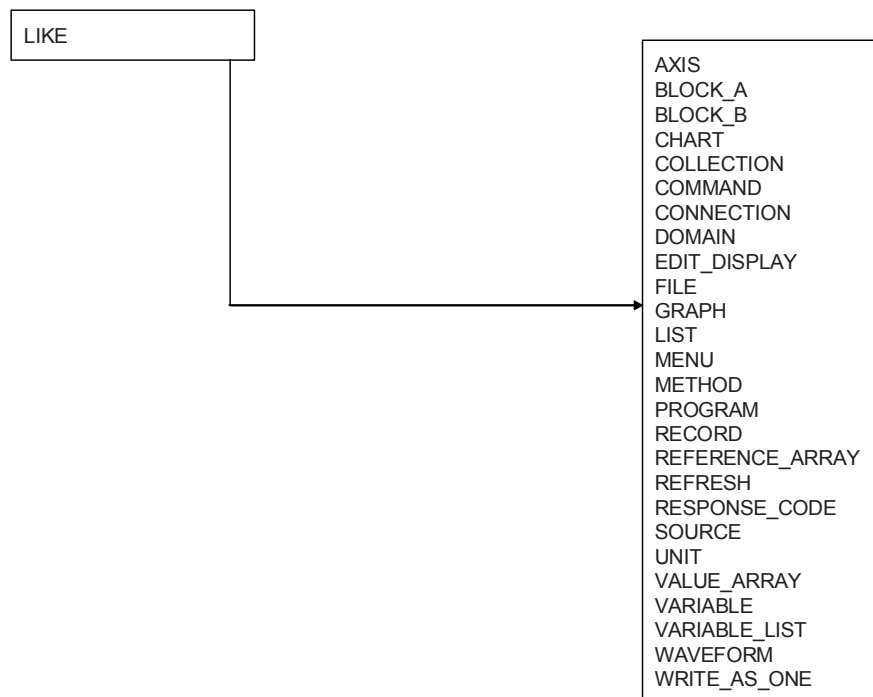
- **IMAGE** describes a visual image (see 0 and Figure 12).



Figure 12 – IMAGE

IEC 1521/06

- **IMPORT** is used to import an EDD and to modify its existing definitions (see 7.19).
- **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 7.20 and Figure 13).



IEC 1522/06

Figure 13 – LIKE

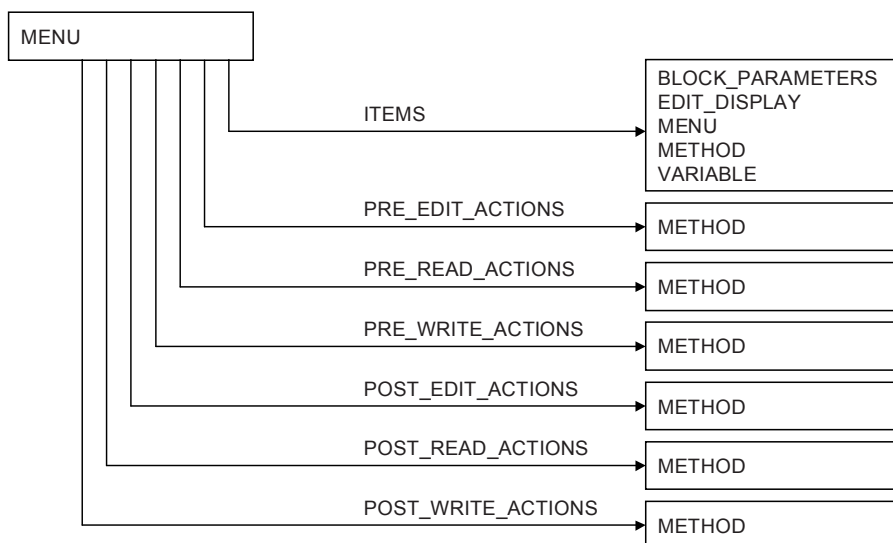
- **LIST** describes a sequence of elements (see 7.22 and Figure 14).



IEC 1523/06

Figure 14 – LIST

- **MENU** describes how the data will be presented to a user by an EDD application (see 7.23 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 7.24).
- **PROGRAM** specifies how device executable code can be initiated by an appropriate agent (see 7.25 and Figure 16).



Figure 16 – PROGRAM

- **RECORD** describes the data contained in the device (see 7.26 and Figure 17).

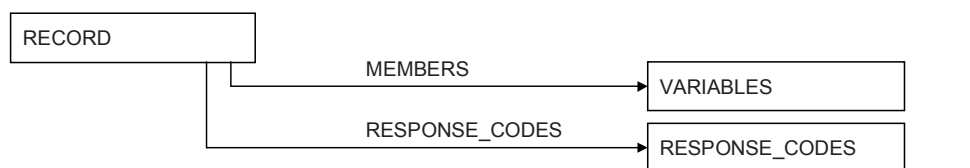


Figure 17 – RECORD

- **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 7.27 and Figure 18).

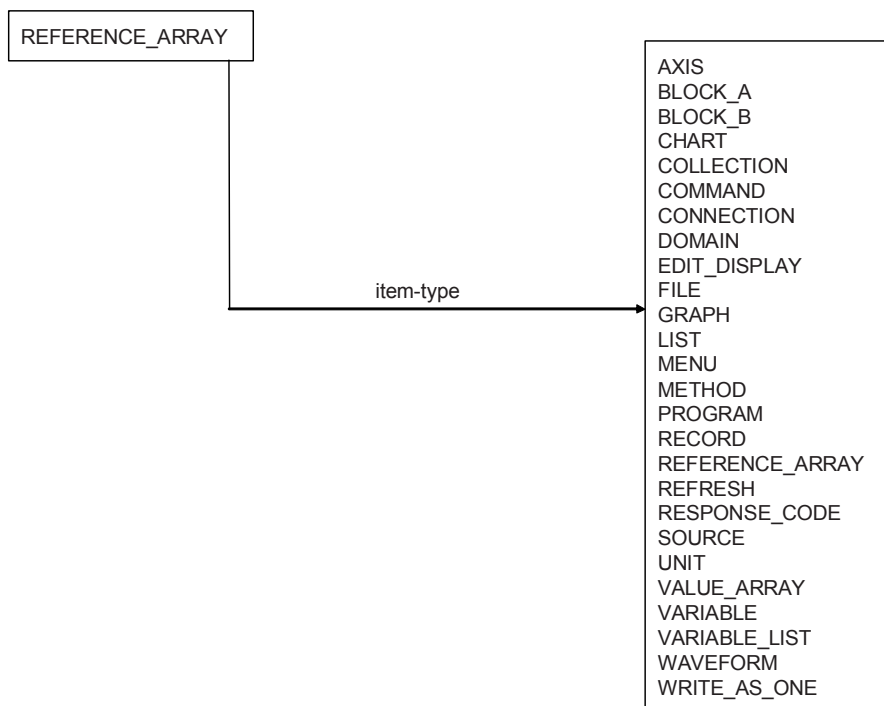


Figure 18 – REFERENCE\_ARRAY

- **Relations (see 7.28):** three types of relations are specified.
  - **REFRESH** describes relationships between VARIABLES, RECORDs, and VALUE\_ARRAYs (see 7.28.1 and Figure 19).



IEC 1528/06

**Figure 19 – REFRESH**

— UNIT describes relationships between VARIABLES, RECORDs, and VALUE\_ARRAYs (see 7.28.2 and Figure 20).



IEC 1529/06

**Figure 20 – UNIT**

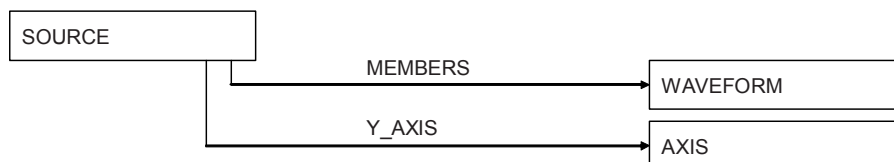
— WRITE\_AS\_ONE RELATION describes relationships between VARIABLES, RECORDs, and VALUE\_ARRAYs (see 7.28.3 and Figure 21).



IEC 1530/06

**Figure 21 – WRITE\_AS\_ONE**

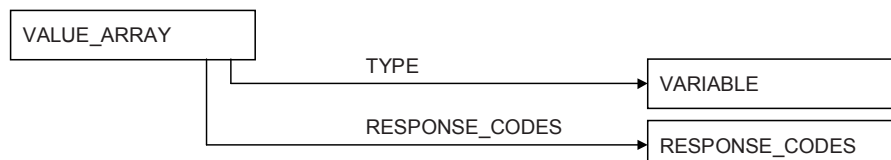
- **RESPONSE\_CODES** specifies the application specific response codes for use when managing a VARIABLE, RECORD, VALUE\_ARRAY, VARIABLE\_LIST, PROGRAM, or DOMAIN (see 7.29).
- **SOURCE** (see 7.30 and Figure 22).



IEC 1531/06

**Figure 22 – SOURCE**

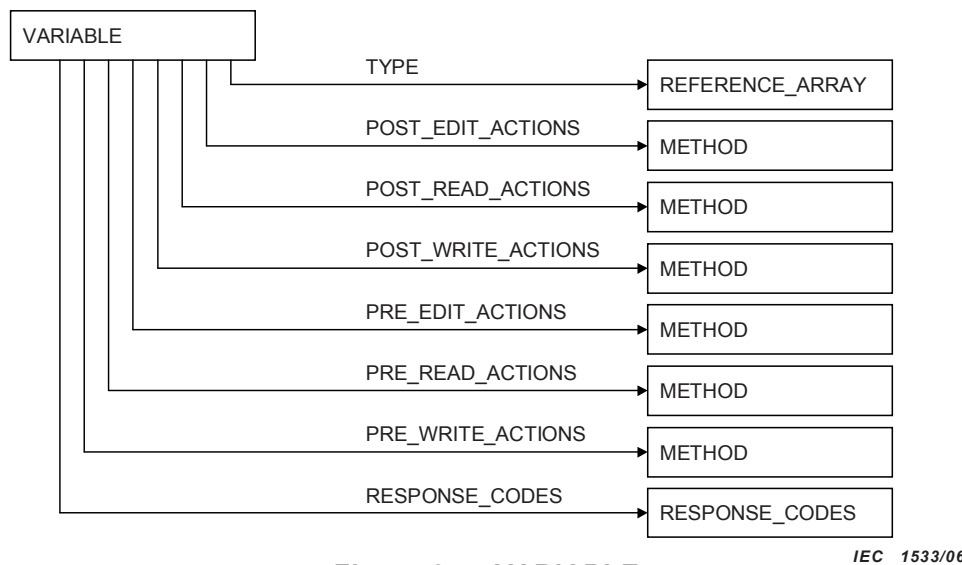
- **VALUE\_ARRAY** describes logical groups of EDDL variables (see 7.30.2.4 and Figure 23).



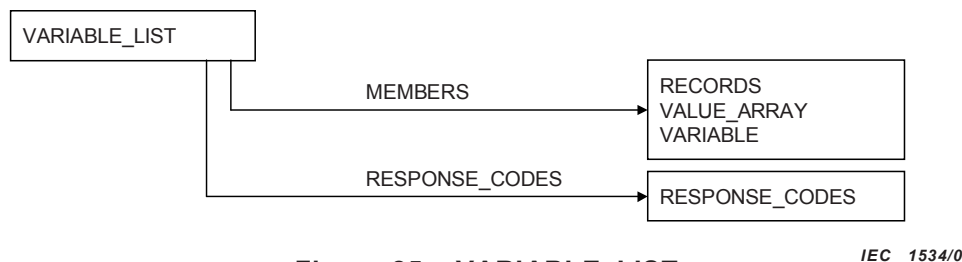
IEC 1532/06

**Figure 23 – VALUE\_ARRAY**

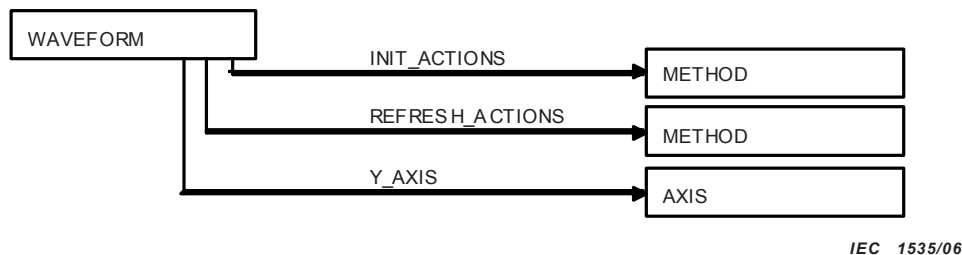
- **VARIABLE** describes the data contained in the device (see 7.33 and Figure 24).

**Figure 24 – VARIABLE**

- **VARIABLE\_LIST** describes logical groupings of data contained in the device that may be communicated as a list (see 7.34 and Figure 25).

**Figure 25 – VARIABLE\_LIST**

- **WAVEFORM** (see 7.35 and Figure 26).

**Figure 26 – WAVEFORM**

### 7.1.5 Common attributes

The following common attributes are used in many basic constructs:

- CLASSIFICATION
- COMPONENT\_PARENT
- COMPONENT\_PATH
- DEFINITION
- EMPHASIS
- HANDLING
- HEIGHT
- HELP
- LABEL

- LINE\_COLOR
- LINE\_TYPE
- MEMBERS
- PROTOCOL
- RESPONSE\_CODES
- SUPPLIED\_INTERFACE
- VALIDITY
- WIDTH

#### 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:

- **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

The DD\_REVISION 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   | A two-octet unsigned integer containing the code for a specific EDD file revision. |

### 7.2.2.2 DEVICE\_REVISION

#### Purpose

The DEVICE\_REVISION attribute specifies a unique code for the device revision of the field device, as defined by the manufacturer.

#### Lexical structure

DEVICE\_REVISION integer

The attribute of DEVICE\_REVISION is specified in Table 3.

**Table 3 – DEVICE\_REVISION attribute**

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

### 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.

NOTE 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. |

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

#### 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. |

#### 7.2.2.6 MANUFACTURER

##### Purpose

The MANUFACTURER attribute identifies the manufacturer. The code allocation should be managed by the specific organizations which are responsible for the different EDDL profiles. The combination of EDD\_PROFILE and MANUFACTURER shall be unique.

##### Lexical structure

MANUFACTURER (long integer, identifier)

The attribute of MANUFACTURER is specified in Table 7.

**Table 7 – MANUFACTURER attribute**

| Usage | Attribute    | Description                                                                                                                                                                                                                  |
|-------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| s     | long integer | <p>An unsigned long integer containing the code for a specific manufacturer.</p> <p>If all bits in the value of the integer are set to one, the lexical structure "MANUFACTURER_EXT" shall be present and identifies the</p> |

| Usage                                                                                                                                                                        | Attribute  | Description                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|----------------------------------------------------------------------------------------------|
|                                                                                                                                                                              |            | manufacturer.                                                                                |
| sm                                                                                                                                                                           | identifier | A reference to an unsigned long integer which contains the code for a specific manufacturer. |
| NOTE 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. |            |                                                                                              |

### 7.2.2.7 MANUFACTURER\_EXT

#### Purpose

The MANUFACTURER\_EXT attribute is reserved for future use. It is a placeholder for a string providing global identification for the MANUFACTURER.

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

#### Lexical structure

MANUFACTURER\_EXT string

The attribute of MANUFACTURER\_EXT is specified in Table 8.

**Table 8 – MANUFACTURER\_EXT attribute**

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

## 7.3 AXIS

### 7.3.1 General structure

#### Purpose

AXIS describes an axis of a CHART or GRAPH.

#### Lexical structure

AXIS identifier [CONSTANT\_UNIT , HELP, LABEL, MAX\_VALUE, MIN\_VALUE, SCALING]

The attributes of AXIS are specified in Table 9.

**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).  |
| 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). |

### 7.3.2 Specific attributes

#### 7.3.2.1 CONSTANT\_UNIT

##### Purpose

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

**Lexical structure**

CONSTANT\_UNIT (string)<exp>

The attribute of CONSTANT\_UNIT is specified in Table 175.

**7.3.2.2 MAX\_VALUE****Purpose**

The MAX\_VALUE attribute specifies the value associated with the last tick mark of the AXIS. The last tick mark of an X\_AXIS and Y\_AXIS is the right-most and top-most tick mark, respectively. By default, the value associated with the last tick mark of an AXIS without a MAX\_VALUE attribute is inferred from the WAVEFORMs on the GRAPH.

**Lexical structure**

MAX\_VALUE (reference, double, float, integer, unsigned\_integer)<exp>

The attributes of MAX\_VALUE are specified in Table 10.

**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. |
| 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.                                                                                                                               |

**7.3.2.3 MIN\_VALUE****Purpose**

The MIN\_VALUE attribute specifies the value associated with the first tick mark of the AXIS. The first tick mark of an X\_AXIS and Y\_AXIS is the left-most and bottom-most tick mark, respectively. By default, the value associated with the first tick mark of an AXIS without a MIN\_VALUE attribute is inferred from the WAVEFORMs on the GRAPH.

**Lexical structure**

MIN\_VALUE (reference, double, float, integer, unsigned\_integer)<exp>

The attributes of MIN\_VALUE are specified in Table 10.

**7.3.2.4 SCALING****Purpose**

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

**Lexical structure**

SCALING (LINEAR, LOGARITHMIC)<exp>

The attributes of SCALING are specified in Table 11.

**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 be scaled logarithmically (base 10). |

## 7.4 BLOCK

### 7.4.1 BLOCK\_A

#### 7.4.1.1 General structure

##### Purpose

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

NOTE This BLOCK\_A approach is optimized for access efficiency.

##### Lexical structure

BLOCK\_A identifier [CHARACTERISTICS, LABEL, PARAMETERS, AXIS\_ITEMS, CHART\_ITEMS, COLLECTION\_ITEMS, EDIT\_DISPLAY\_ITEMS, FILE\_ITEMS, GRAPH\_ITEMS, GRID\_ITEMS, HELP, IMAGE\_ITEMS, LIST\_ITEMS, MENU\_ITEMS, METHOD\_ITEMS, PARAMETER\_LISTS, REFERENCE\_ARRAY\_ITEMS, REFRESH\_ITEMS, SOURCE\_ITEMS, UNIT\_ITEMS, WAVEFORM\_ITEMS, WRITE\_AS\_ONE\_ITEMS, CHARTS, LISTS, GRAPHS, GRIDS, MENUS, METHODS]

The attributes of BLOCK\_A are specified in Table 12.

**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 elements (see 7.4.1.2.9).  |
| o     | HELP                  | Lexical element (see 7.36.8).      |
| o     | IMAGE_ITEMS           | Lexical elements (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).  |
| 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     | 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     | 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).  |

| Usage | Attribute | Description                       |
|-------|-----------|-----------------------------------|
| o     | MENUS     | Lexical element (see 7.4.1.2.26). |
| o     | METHODS   | Lexical element (see 7.4.1.2.27). |

### 7.4.1.2 Specific attributes

#### 7.4.1.2.1 CHARACTERISTICS

##### Purpose

The CHARACTERISTICS attribute specifies the external characteristics for a BLOCK\_A. CHARACTERISTICS may include class, function type, block type and execution time. The external characteristics of a BLOCK\_A are contained in a RECORD.

NOTE In general the BLOCK\_A CHARACTERISTICS fix the type of the BLOCK\_A.

##### Lexical structure

CHARACTERISTICS reference

The attribute of CHARACTERISTICS is specified in Table 13.

**Table 13 – CHARACTERISTIC attribute**

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

#### 7.4.1.2.2 PARAMETERS

##### Purpose

The PARAMETERS attribute contains a list of references to VARIABLE, VALUE\_ARRAY or RECORD instances. Each reference shall have an identifier and may have a description and/or a help text.

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

##### Lexical structure

PARAMETERS [identifier, reference, description, help]+

The attributes of PARAMETERS are specified in Table 14.

**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.                                                                                                                               |

#### 7.4.1.2.3 AXIS\_ITEMS

##### Purpose

The AXIS\_ITEMS attribute is a list of the AXIS instances associated with the BLOCK\_A.

##### Lexical structure

AXIS\_ITEMS reference+

The attribute of AXIS\_ITEMS is specified in Table 15.

**Table 15 – AXIS\_ITEMS attribute**

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

**7.4.1.2.4 CHART\_ITEMS****Purpose**

The CHART\_ITEMS attribute is a list of the CHART instances associated with the BLOCK\_A.

**Lexical structure**

CHART\_ITEMS reference+

The attribute of CHART\_ITEMS is specified in Table 16.

**Table 16 – CHART\_ITEMS attribute**

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

**7.4.1.2.5 COLLECTION\_ITEMS****Purpose**

The COLLECTION\_ITEMS attribute is a list of the COLLECTION instances associated with the BLOCK\_A.

**Lexical structure**

COLLECTION\_ITEMS reference+

The attribute of COLLECTION\_ITEMS is specified in Table 17.

**Table 17 – COLLECTION\_ITEMS attribute**

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

**7.4.1.2.6 EDIT\_DISPLAY\_ITEMS****Purpose**

The EDIT\_DISPLAY\_ITEMS attribute is a list of the EDIT\_DISPLAY instances associated with BLOCK\_A.

**Lexical structure**

EDIT\_DISPLAY\_ITEMS reference+

The attribute of EDIT\_DISPLAY\_ITEMS is specified in Table 18.

**Table 18 – EDIT\_DISPLAY\_ITEMS attribute**

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

**7.4.1.2.7 FILE\_ITEMS****Purpose**

The FILE\_ITEMS attribute is a list of the FILE instances associated with the BLOCK\_A.

**Lexical structure**

FILE\_ITEMS reference+

The attribute of FILE\_ITEMS is specified in Table 19.

**Table 19 – FILE\_ITEMS attribute**

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

#### 7.4.1.2.8 GRAPH\_ITEMS

##### Purpose

The GRAPH\_ITEMS attribute is a list of the GRAPH instances associated with the BLOCK\_A.

##### Lexical structure

GRAPH\_ITEMS reference+

The attribute of GRAPH\_ITEMS is specified in Table 20.

**Table 20 – GRAPH\_ITEMS attribute**

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

#### 7.4.1.2.9 GRID\_ITEMS

##### Purpose

The GRID\_ITEMS attribute is a list of the GRID instances associated with the BLOCK\_A.

##### Lexical structure

GRID\_ITEMS reference+

The attribute of GRID\_ITEMS is specified in Table 21.

**Table 21 – GRID\_ITEMS attribute**

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

#### 7.4.1.2.10 IMAGE\_ITEMS

##### Purpose

The IMAGE\_ITEMS attribute is a list of the IMAGE instances associated with BLOCK\_A.

##### Lexical structure

IMAGE\_ITEMS reference+

The attribute of IMAGE\_ITEMS is specified in Table 22.

**Table 22 – IMAGE\_ITEMS attribute**

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

#### 7.4.1.2.11 LIST\_ITEMS

##### Purpose

The LIST\_ITEMS attribute is a list of the LIST instances associated with BLOCK\_A.

##### Lexical structure

LIST\_ITEMS reference+

The attribute of LIST\_ITEMS is specified in Table 23.

**Table 23 – LIST\_ITEMS attribute**

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

#### 7.4.1.2.12 LOCAL\_PARAMETERS

##### Purpose

The LOCAL\_PARAMETERS attribute contains a list of references to VARIABLE, VALUE\_ARRAY or RECORD instances. Each reference shall have an identifier and may have a description and/or a help text.

NOTE The LOCAL\_PARAMETERS in a BLOCK\_A do not correspond to communication objects; they are parameters that are only used by the EDD application.

##### Lexical structure

LOCAL\_PARAMETERS [identifier, reference, description, help]+

The attributes of LOCAL\_PARAMETERS are specified in Table 14.

#### 7.4.1.2.13 MENU\_ITEMS

##### Purpose

The MENU\_ITEMS attribute specifies the MENUs associated with BLOCK\_A.

##### Lexical structure

MENU\_ITEMS reference+

The attribute of MENU\_ITEMS is specified in Table 24.

**Table 24 – MENU\_ITEMS attribute**

| Usage | Attribute | Description                     |
|-------|-----------|---------------------------------|
| m     | reference | A reference to a MENU instance. |

#### 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.

##### Lexical structure

REFRESH\_ITEMS reference+

The attribute of REFRESH\_ITEMS is specified in Table 28.

**Table 28 – REFRESH\_ITEMS attribute**

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

#### 7.4.1.2.18 SOURCE\_ITEMS

##### Purpose

The SOURCE\_ITEMS attribute specifies the SOURCES associated with the BLOCK\_A.

##### Lexical structure

SOURCE\_ITEMS reference+

The attribute of SOURCE\_ITEMS is specified in Table 29.

**Table 29 – SOURCE\_ITEMS attribute**

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

**7.4.1.2.19 UNIT\_ITEMS****Purpose**

The UNIT\_ITEMS attribute specifies the UNIT relations associated with the BLOCK\_A.

**Lexical structure**

UNIT\_ITEMS reference+

The attribute of UNIT\_ITEMS is specified in Table 30.

**Table 30 – UNIT\_ITEMS attribute**

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

**7.4.1.2.20 WAVEFORM\_ITEMS****Purpose**

The WAVEFORM\_ITEMS attribute specifies the WAVEFORMs associated with the BLOCK\_A.

**Lexical structure**

WAVEFORM\_ITEMS reference+

The attribute of WAVEFORM\_ITEMS is specified in Table 31.

**Table 31 – WAVEFORM\_ITEMS attribute**

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

**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) need 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) need 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.                       |

**7.4.1.2.24 GRAPHS****Purpose**

The GRAPHS attribute specifies the GRAPHS associated with the BLOCK\_A. Only those GRAPHS that will be referenced via a specific BLOCK\_A instance (see 7.38.23) need be specified.

**Lexical structure**

GRAPHS (identifier, reference, description, help)+

The attribute of GRAPHS is specified in Table 35.

**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.                       |

**7.4.1.2.25 GRIDS****Purpose**

The GRIDS attribute specifies the GRIDS associated with the BLOCK\_A. Only those GRIDS that will be referenced via a specific BLOCK\_A instance (see 7.38.24) need be specified.

**Lexical structure**

GRIDS (identifier, reference, description, help)+

The attribute of GRIDS is specified in Table 36.

**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.                       |

**7.4.1.2.26 MENUS****Purpose**

The MENUS attribute specifies the MENUS associated with the BLOCK\_A. Only those MENUS that will be referenced via a specific BLOCK\_A instance (see 7.38.25) need be specified.

**Lexical structure**

MENUS (identifier, reference, description, help)+

The attribute of MENUS is specified in Table 37.

**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.                       |

**7.4.1.2.27 METHODS****Purpose**

The METHODS attribute specifies the METHODS associated with the BLOCK\_A. Only those METHODS that will be referenced via a specific BLOCK\_A instance (see 7.38.26) need be specified.

**Lexical structure**

METHODS (identifier, reference, description, help)+

The attribute of METHODS is specified in Table 38.

**Table 38 – METHODS attribute**

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

**7.4.2 BLOCK\_B****7.4.2.1 General structure****Purpose**

The variables of a device or module respectively are structured in blocks corresponding to device components or functional parts. Three types of blocks are defined: PHYSICAL, TRANSDUCER and FUNCTION. More than one BLOCK\_B from the same TYPE can be

instantiated. For efficient access, each instance has its own NUMBER. For each of the block types there are different threads of NUMBER sequences, starting with 1.

When accessing a VARIABLE within a device, the BLOCK\_B construct is used in conjunction with the COMMAND to provide relative addressing.

NOTE 1 This BLOCK\_B approach is optimized for memory efficiency.

NOTE 2 The storage of a variable in a device is manufacturer specific and is represented by a device directory. A device directory contains a summary of the available BLOCK\_B entries in a device. Furthermore, the device directory contains for each BLOCK\_B a pointer to its physical address. An EDD application finds a single object by adding an offset to the BLOCK\_B physical address.

NOTE 3 Within a BLOCK\_B instance, relative indexing is used. This relative index is defined in the device profile.

### Lexical structure

BLOCK\_B identifier [NUMBER, TYPE]

The attributes of BLOCK\_B are specified in Table 39.

**Table 39 – BLOCK\_B attributes**

| Usage | Attribute | Description                      |
|-------|-----------|----------------------------------|
| m     | NUMBER    | Lexical element (see 7.4.2.2.1). |
| m     | TYPE      | Lexical element (see 7.4.2.2.2). |

## 7.4.2.2 Specific attributes

### 7.4.2.2.1 NUMBER

#### Purpose

The NUMBER attribute enumerates each BLOCK\_B of the same TYPE in the device. A device may contain several BLOCK\_B instances, which are identified with the NUMBER attribute.

#### Lexical structure

NUMBER (integer, expression)<exp>

The attribute of NUMBER is specified in Table 40.

**Table 40 – NUMBER attributes**

| Usage | Attribute  | Description                                                                                                                                                               |
|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| s     | integer    | The number of the BLOCK_B instance of the same TYPE.                                                                                                                      |
| s     | expression | An expression which shall be evaluated to a non-negative integer that is within the index range of the NUMBER attribute (for the description of expressions, see 7.39.7). |

### 7.4.2.2.2 TYPE

#### Purpose

The TYPE attribute is used to specify one of the three block types.

#### Lexical structure

TYPE [PHYSICAL, TRANSDUCER, FUNCTION]

The attributes of TYPE are specified in Table 41.