

BS EN 61804-4:2016



BSI Standards Publication

Function blocks (FB) for process control and Electronic Device Description Language (EDDL)

Part 4: EDD interpretation

bsi.

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

National foreword

This British Standard is the UK implementation of EN 61804-4:2016. It is identical to IEC 61804-4:2015. It supersedes PD CLC/TR 61804-4:2007 which is withdrawn.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.

© The British Standards Institution 2016.

Published by BSI Standards Limited 2016

ISBN 978 0 580 81632 1

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 29 February 2016.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 61804-4

January 2016

ICS 25.040.40; 35.240.50

Supersedes CLC/TR 61804-4:2007

English Version

Function blocks (FB) for process control and Electronic Device Description Language (EDDL) - Part 4: EDD interpretation (IEC 61804-4:2015)

Blocs fonctionnels (FB) pour les procédés industriels et le
langage de description électronique de produit (EDDL) -
Partie 4: Interprétation EDD
(IEC 61804-4:2015)

Funktionsbausteine für die Prozessautomation und
elektronische Gerätebeschreibungssprache - Teil 4:
Interpretation von Gerätebeschreibungen
(IEC 61804-4:2015)

This European Standard was approved by CENELEC on 2015-11-11. 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 CEN-CENELEC Management Centre 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 CEN-CENELEC Management Centre 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, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

© 2016 CENELEC All rights of exploitation in any form and by any means reserved worldwide for CENELEC Members.

European foreword

The text of document 65E/465/FDIS, future edition 1 of IEC 61804-4, 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 approved by CENELEC as EN 61804-4:2016.

The following dates are fixed:

- latest date by which the document has to be (dop) 2016-08-11
implemented at national level by
publication of an identical national
standard or by endorsement
- latest date by which the national (dow) 2018-11-11
standards conflicting with the
document have to be withdrawn

This document supersedes CLC/TR 61804-4:2007.

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

Endorsement notice

The text of the International Standard IEC 61804-4:2015 was approved by CENELEC as a European Standard without any modification.

Annex ZA

(normative)

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

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

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

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

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61784-1	-	Industrial communication networks - Profiles -- Part 1: Fieldbus profiles	EN 61784-1	-
IEC 61784-2	-	Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3	EN 61784-2	-
IEC 61804-2	-	Function Blocks (FB) for process control -- Part 2: Specification of FB concept	EN 61804-2	-
IEC 61804-3	-	Function blocks (FB) for process control and EDDL - Part 3: EDDL specification and communication profiles	EN 61804-3	-
IEC 61804-5	-	Function blocks (FB) for process control and EDDL - Part 5: EDDL Built-in library	EN 61804-5	-
ISO/IEC 10918	series	Information technology - Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)	-	series
ISO/IEC 15948	-	Information technology - Computer graphics and image processing - Portable Network Graphics (PNG) - Functional specification	-	-

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references	11
3 Terms, definitions, abbreviated terms, acronyms and conventions.....	11
3.1 General terms and definitions	12
3.2 Terms and definitions related to modular devices.....	12
3.3 Abbreviated terms and acronyms	13
3.4 Conventions.....	13
4 EDDL user interface description	13
4.1 Overview.....	13
4.2 Menu conventions for handheld applications	14
4.3 Menu conventions for PC-based applications	14
4.3.1 Overview	14
4.3.2 Online Root Menus	14
4.3.3 Offline Root Menu.....	15
4.3.4 Example of EDD menu structure	15
4.3.5 User interface	20
4.4 Containers and contained items	23
4.4.1 Overview	23
4.4.2 Containers	23
4.4.3 Contained items.....	26
4.5 Layout rules	30
4.5.1 Overview	30
4.5.2 Layout rules for WIDTH and HEIGHT	30
4.5.3 Layout rules for COLUMNBREAK and ROWBREAK.....	31
4.5.4 Layout examples	37
4.5.5 Conditional user interface	45
4.6 Graphical elements	46
4.6.1 Overview	46
4.6.2 Graph and chart.....	47
4.6.3 Common attributes	47
4.6.4 CHART	48
4.6.5 GRAPH.....	56
4.6.6 AXIS.....	65
4.6.7 IMAGE.....	66
4.6.8 GRID	67
5 EDDL data description	69
5.1 Variables	69
5.1.1 VARIABLE TYPEs	69
5.1.2 VARIABLE CLASS.....	70
5.1.3 VARIABLE ACTIONS.....	70
5.2 EDDL application stored device data.....	70
5.2.1 Overview	70
5.2.2 FILE	71
5.2.3 LIST	73

5.3	Exposing data items outside the EDD application.....	80
5.4	Initialization of EDD instances.....	80
5.4.1	Overview	80
5.4.2	Initialization support	80
5.4.3	TEMPLATE.....	80
5.5	Device model mapping.....	81
5.5.1	BLOCK_A	81
5.5.2	BLOCK_B	82
6	EDDL METHOD programming and usage of Builtins	82
6.1	Builtin MenuDisplay	82
6.2	Division by zero and undetermined floating values	85
6.2.1	Integer and unsigned integer values	85
6.2.2	Floating-point values	85
7	Modular devices	85
7.1	Overview.....	85
7.2	EDD identification	86
7.3	Instance object model	86
7.4	Offline configuration.....	87
7.5	Online configuration.....	87
7.6	Simple modular device example.....	87
7.6.1	General	87
7.6.2	Separate EDD file example with direct EDD referencing	88
7.6.3	Separate EDD file example with classification EDD referencing and interfaces	89
7.6.4	One EDD file example	92
7.6.5	Combination of single and separate modular device example	93
7.7	COMPONENT_RELATION	93
7.7.1	General	93
7.7.2	NEXT_COMPONENT usage	93
7.7.3	REQUIRED_RANGES and ADDRESSING usage	93
7.8	Upload and download for modular devices	93
7.9	Diagnostic.....	94
7.10	Reading modular device topology	95
7.10.1	SCAN	95
7.10.2	Detect module type.....	96
7.11	Configuration check	97
8	Edit session.....	98
8.1	Data management.....	98
8.1.1	Overview	98
8.1.2	General rules.....	99
8.1.3	Data caching for dialogs and windows	99
8.1.4	Data caching for METHODS.....	100
8.2	UI aspects of editing sessions.....	102
8.3	User roles	103
9	Offline and online configuration	103
9.1	Overview.....	103
9.2	Offline dataset	104
9.3	Offline configuration.....	104
9.4	Online dataset	104

9.5	Online configuration	104
9.6	Upload and download	105
9.6.1	Overview	105
9.6.2	Error recovery.....	106
9.6.3	Upload procedure	106
9.6.4	Download procedure	107
10	EDDL communication description	109
10.1	COMMAND	109
10.1.1	General	109
10.1.2	OPERATION.....	109
10.1.3	TRANSACTION	110
10.1.4	Command addressing	113
10.2	Parsing data received from the device	114
10.2.1	General	114
10.2.2	Parsing complex data items.....	114
10.2.3	FOUNDATION Fieldbus	114
10.2.4	HART	115
10.2.5	PROFIBUS and PROFINET	115
10.3	FOUNDATION Fieldbus communication model.....	115
11	EDD development.....	119
11.1	Dictionaries.....	119
11.2	Reserved	119
Annex A	(normative) Device simulation.....	120
Annex B	(informative) Predefined identifiers	121
Figure 1	– EDD example of root menus.....	20
Figure 2	– Example of an EDD application for diagnostics	20
Figure 3	– Example of an EDD application for process variables.....	21
Figure 4	– Example of an EDD application for primary variables	21
Figure 5	– Example of an EDD application for process-related device features	22
Figure 6	– Example of an EDD application for device features	22
Figure 7	– Example of an EDD application for maintenance features	23
Figure 8	– Usage of COLLECTION MEMBERS in MENUs of STYLE GROUP	26
Figure 9	– Displaying single bits of BIT_ENUMERATED	27
Figure 10	– Displaying multiple bits of BIT_ENUMERATED.....	28
Figure 11	– Example of an EDD application for a variable of type BIT_ENUMERATED	28
Figure 12	– EDD source code for layout for protruding elements example.....	32
Figure 13	– Layout for protruding elements	32
Figure 14	– EDD source code for layout for partially filled rows example.....	33
Figure 15	– Layout for partially filled rows	33
Figure 16	– EDD source code for layout for partially filled rows example.....	34
Figure 17	– Layout for partially filled rows.....	34
Figure 18	– EDD source code for layout for oversized elements example.....	35
Figure 19	– Layout for oversized elements.....	35
Figure 20	– EDD source code example for a layout for columns in stacked group	36
Figure 21	– Layout for columns in stacked group	

Figure 22 – EDD source code for layout for columns with GRAPHS in stacked group example	37
Figure 23 – Layout for columns with GRAPHS in stacked group	37
Figure 24 – Example of an EDD for an overview menu	37
Figure 25 – Example of an EDD application for an overview window	38
Figure 26 – Example of an EDD using COLUMNBREAK	38
Figure 27 – Example of an EDD application for an overview window	39
Figure 28 – EDD example for an overview window	39
Figure 29 – Example of an EDD application for an overview window	40
Figure 30 – Example of an EDD for in-line graphs and charts	40
Figure 31 – Example of an EDD application for an in-line graph	41
Figure 32 – Example of an EDD for full-width graphs and charts	41
Figure 33 – Example of an EDD application for a full-width graph	42
Figure 34 – Example of an EDD for nested containers	43
Figure 35 – Example of an EDD application for nested containers	43
Figure 36 – Example of an EDD for EDIT_DISPLAYS	44
Figure 37 – Example of an EDD application for EDIT_DISPLAYS	44
Figure 38 – Example of an EDD for images	45
Figure 39 – Example of an EDD application for images	45
Figure 40 – HEIGHT and WIDTH attributes for CHART and GRAPH	47
Figure 41 – EMPHASIS attribute to differentiate one or more SOURCES or WAVEFORMs	48
Figure 42 – Example of a chart with one curve in a dialog	50
Figure 43 – Example of a chart with two SOURCES	51
Figure 44 – Displaying example of a chart with two SOURCES	52
Figure 45 – Example of a chart with three horizontal bars	53
Figure 46 – Displaying example of a chart with three horizontal bars	54
Figure 47 – Example of a chart in a dialog	56
Figure 48 – A graph and the visual elements	57
Figure 49 – Example of a graph	60
Figure 50 – Multiple used axes	61
Figure 51 – EDD with device-supported zooming and scrolling	65
Figure 52 – EDD example of an IMAGE	66
Figure 53 – EDD example of an IMAGE with the LINK attribute	66
Figure 54 – EDD example of a GRID	68
Figure 55 – Result of the EDD example	68
Figure 56 – Wrong usage of a BIT_ENUMERATED variable	69
Figure 57 – Usage of ENUMERATED instead of BIT_ENUMERATED	69
Figure 58 – Example of a file declaration	72
Figure 59 – Example of comparing valve signatures	73
Figure 60 – Example of more complex file declaration	74
Figure 61 – Example of reviewing the stored radar signals	75
Figure 62 – Example of an EDD that inserts, replaces, or compares radar signals	80
Figure 63 – Example of TEMPLATE usage	81

Figure 64 – Example of a BLOCK_A	82
Figure 65 – Example of a wizard	84
Figure 66 – The different relations of a module	87
Figure 67 – Components and possible configuration of the modular devices	87
Figure 68 – Separate EDD file example with direct EDD referencing	88
Figure 69 – EDD example for module1	89
Figure 70 – EDD example for module2	89
Figure 71 – EDD example for modular device	90
Figure 72 – EDD example for module1	91
Figure 73 – EDD example for module2	91
Figure 74 – EDD example for module2	93
Figure 75 – NEXT_COMPONENT usage	93
Figure 76 – REQUIRED_RANGES usage	93
Figure 77 – Upload/download order of a modular device	94
Figure 78 – Example of a SCAN METHOD	96
Figure 79 – Example of a DETECT METHOD	97
Figure 80 – Example of a CHECK_CONFIGURATION METHOD	98
Figure 81 – Data caching for an offline session	98
Figure 82 – Data caching for an online session	99
Figure 83 – Sub dialogs or windows using a shared edit cache	100
Figure 84 – Sub dialogs or windows using separate edit caches	100
Figure 85 – Data caching for nested METHODS	101
Figure 86 – Data caching for a METHOD invoked within a dialog	101
Figure 87 – Data caching for a METHOD invoking a dialog using an edit cache	101
Figure 88 – Data caching for a METHOD invoking a dialog	102
Figure 89 – Data flow for download to the device	105
Figure 90 – Data flow for upload from the device	105
Figure 91 – Example of a single item mask	111
Figure 92 – Mapping example with a single item mask	111
Figure 93 – Multiple item masks	111
Figure 94 – Mapping example with a multiple item mask	112
Figure 95 – INFO qualifier	112
Figure 96 – INDEX qualifier	113
Figure 97 – INFO and INDEX qualifier	113
Figure 98 – Example device with 2 unique BLOCK_A definitions	116
Figure 99 – Example EDD for a device with 2 unique BLOCK_A definitions	117
Figure 100 – BLOCK_A example with PARAMETER_LISTS	118
Figure 101 – Example EDD for a BLOCK_A with PARAMETER_LISTS	119
Table 1 – List of defined root menu identifiers for handhelds	14
Table 2 – List of defined root menu identifiers for PC-based devices	14
Table 3 – Fall back alternatives for online root menus	15
Table 4 – Fall back alternatives for offline root menus	15
Table 5 – Permitted contained items and default STYLES	24

Table 6 – WIDTH and HEIGHT span and applicability	31
Table 7 – Image formats	67
Table 8 – String handling	69
Table 9 – Examples of floating-point results	85
Table 10 – Usages of COMPONENT_PATH	86
Table 11 – Diagnostic classifications	95
Table 12 – Builtins for method cache controlling	102
Table 13 – List of defined upload menu identifiers	106
Table 14 – List of defined download menu identifiers	107
Table 15 – PROFIBUS and PROFINET communication mapping.....	110
Table B.1 – Predefined identifiers	121

INTERNATIONAL ELECTROTECHNICAL COMMISSION

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

Part 4: EDD interpretation

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61804-4 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This first edition cancels and replaces IEC TR 61804-4 published in 2006. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- New paragraph:
 - EDDL data description
 - EDDL METHOD programming and usage of builtins
 - Edit session
 - Offline and online configuration

- EDDL communication description
- Enhancements in paragraph EDDL user interface descriptions

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

FDIS	Report on voting
65E/465/FDIS	65E/481/RVD

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

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

A list of all parts in the IEC 61804 series, published under the general title *Function blocks (FB) for process control and electronic device description language (EDDL)*, can be found on the IEC website.

Future standards in this series will carry the new general title as cited above. Titles of existing standards in this series will be updated at the time of the next edition.

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

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

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

INTRODUCTION

This part of IEC 61804 was developed using material from FDI Cooperation LLC (Foundation[™] Fieldbus¹, HART^{®2} Communication Foundation (HCF), PROFIBUS^{™3} Nutzerorganisation e.V. (PNO)), OPC Foundation (OPCF) and FDT Group. IEC 61804 has the general title "Function blocks (FB) for process control and Electronic Device Description Language (EDDL)".

This edition does reflect many of the various rules defined by the different communication foundations, however it is not a complete representation of those rules defined by each of the communication foundations today. Therefore, an EDD application and EDD developer will need to rely on both IEC 61804-4 and the respective communication foundation documents (e.g. specifications, test requirements, test cases) to develop a conformant application that will meet foundation registration requirements.

Conformity assessment of an EDD application is the responsibility of the respective communication foundations. In cases of any ambiguity, the rules of the respective communication foundations apply.

This part of IEC 61804

- contains an overview of the use of EDDL;
- provides examples demonstrating the use of the EDDL constructs;
- shows how the use cases are fulfilled; and
- shows the proper EDD application interpretation for each example.

This part of IEC 61804 is not an EDDL tutorial and is not intended to replace the EDDL specification.

Instructions are provided for the EDD application, which describe what will be performed without prescribing the technology used in the host implementation. For example, the FILE construct describes data that is stored by the EDD application on behalf of the EDD. The FILE construct does not specify how the data is stored. The EDD application can use a database, a flat file, or any other implementation it chooses.

¹ FOUNDATION[™] Fieldbus is the trademark of the Fieldbus Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

² HART[®] is the registered trademark of the HART Communication Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

³ PROFIBUS and PROFINET are the trademarks of the PROFIBUS Nutzerorganisation e.V. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

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

Part 4: EDD interpretation

1 Scope

This part of IEC 61804 specifies EDD interpretation for EDD applications and EDDs to support EDD interoperability. This document is intended to ensure that field device developers use the EDDL constructs consistently and that the EDD applications have the same interpretations of the EDD. It supplements the EDDL specification to promote EDDL application interoperability and improve EDD portability between EDDL applications.

2 Normative references

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

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

IEC 61804-2, *Function blocks (FB) for process control – Part 2: Specification of FB concept*

IEC 61804-3⁴, *Function blocks (FB) for process control and Electronic device description language (EDDL) – Part 3: EDDL syntax and semantics*

IEC 61804-5⁵, *Function blocks (FB) for process control and Electronic device description language (EDDL) – Part 5: EDDL Builtin library*

ISO/IEC 10918 (all parts), *Information technology – Digital compression and coding of continuous-tone still images*

ISO/IEC 15948, *Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification*

3 Terms, definitions, abbreviated terms, acronyms and conventions

For the purposes of this document, the terms and definitions given in IEC 61804-3 as well as the following apply.

⁴ To be published.

⁵ To be published.

3.1 General terms and definitions

3.1.1

EDD developer

individual or team that develops an EDD

3.1.2

container

user interface elements that contain other user interface elements

Note 1 to entry: Containers can include menus, windows, dialogs, tables, pages, groups, and other containers.

3.1.3

contained item

user interface elements that can be contained in containers

Note 1 to entry: Contained items can include variables, methods, graphs, charts, images, static text.

3.1.4

device developer

individual or team that develops a device and an EDD that describes the device

3.1.5

handheld

device with limited display resolution that restricts EDD applications user interface

3.2 Terms and definitions related to modular devices

3.2.1

channel

connection to a process that is being measured or controlled

3.2.2

component

software or hardware item contained within the modular device concept

Note 1 to entry: A component cannot function separately from a modular device hosting it. A component may support one or more types of modular devices.

3.2.3

interface

basic declarations of basic constructs

Note 1 to entry: An interface defines all public parts that components may use.

3.2.4

modal window

a child window that requires users to interact with it before they can return to operating the parent application, thus preventing the workflow on the application main window

3.2.5

modular device

device that can contain a variety of software and or hardware components

3.3 Abbreviated terms and acronyms

CP	Communication Profile
CPF	Communication Profile Family
EDD	Electronic Device Description
EDDL	Electronic Device Description Language
PC	Personal computer
PI	PROFIBUS and PROFINET International
PI PROFILE PA	PI specific profile for Process Automation field devices

3.4 Conventions

There exists some differences using and interpreting EDDL based on the used communication network and device model. The different communication networks used within this part of IEC 61804 are:

- HART (according to IEC 61784-1 CPF 9)
- FOUNDATION fieldbus (according to IEC 61784-1 CPF 1)
- PROFIBUS (according to IEC 61784-1 CPF 3)
- PROFINET (according to IEC 61784-2 CPF 3)

EDD examples in this standard show parts of EDD implementations and are incomplete.

EDDL keywords are written in upper case letters. If more than one basic construct item is meant the keyword is written in uppercase letters added with a lower case 's' for example VARIABLES.

The capitalized word Builtin refers to functions specified in IEC 61804-5.

EXAMPLE Builtin MenuDisplay refers to the Builtin named MenuDisplay specified in IEC 61804-5.

4 EDDL user interface description

4.1 Overview

Most EDD applications can be characterized as either a PC application or a handheld application. Due to the relatively small screen of a handheld device, handheld applications can only display a small amount of information at any given time. On the other hand, PC applications can provide a much more beneficial user interface, largely due to their larger screen size.

To support the capabilities of PC applications, the MENU construct has been extended in IEC 61804-3 compared to previous definitions in IEC 61804-2. Due to the differences in the user interfaces of PC applications and handheld applications, it is expected that many devices will define two MENU hierarchies – one for handheld applications and the other for PC applications. Some MENUs may be used in both hierarchies. Therefore, the entire hierarchy does not need to be specified twice.

Different menu structures for different classes of applications are possible. This standard shall be used to create menu structures in an EDD that are interpreted by applications in an unambiguous way. To provide interoperability across applications, this standard shall be followed.

4.2 Menu conventions for handheld applications

EDD applications use specific menus in the EDDs to show the user interface of the device (see Table 1). In addition user interface items can be described for handhelds and PCs at once. For handhelds strings can have beside a language code, a specific country code zz to specify shorter strings or lower resolution images (see IEC 61804-3).

Table 1 – List of defined root menu identifiers for handhelds

Menu identifier	Default STYLE	Short description
root_menu	TABLE	Handheld root menu for HART devices. This is mandatory for all HART EDDs
Menu_Top*	MENU	Handheld root menu prefix for BLOCK_A MENU_ITEMS for FOUNDATION fieldbus devices e.g. Menu_Top_TB

4.3 Menu conventions for PC-based applications

4.3.1 Overview

EDD applications use special menus in the EDDs to show the user interface of the device. Such menus are defined for diagnostic, process variables, device features, and offline configuration. Table 2 defines identifiers for the different root menus with its default STYLES. The default STYLE is used by the EDD application if the STYLE attribute is not defined in the MENU. The “Usage” column in Table 2 defines how the EDD application has online or offline access to the device parameter (see Clause 8).

Table 2 – List of defined root menu identifiers for PC-based devices

MENU identifier	Default STYLE	Usage	Short description
device_root_menu	MENU	Online	Device feature views for set up
diagnostic_root_menu	MENU	Online	Diagnostic views
maintenance_root_menu	MENU	Online	Maintenance feature views
offline_root_menu	TABLE	Offline	Offline configuration
process_variables_root_menu	MENU	Online	Process variable views

For HART EDDs the EDD application shall display offline_root_menu with a default STYLE WINDOW if STYLE is not defined.

PROFIBUS and PROFINET allow in submenus to define different parameter access by defining ACCESS ONLINE or ACCESS OFFLINE.

The EDDs should contain the online root menus from Table 2. The online root menus are optional for PROFIBUS, PROFINET and FOUNDATION fieldbus. The offline root menu is optional for FOUNDATION fieldbus. HART EDDs shall have all root menus in Table 2.

4.3.2 Online Root Menus

4.3.2.1 General

If none of the online root menu exists then communication profile specific entry points shall be used (see Table 3).

Table 3 – Fall back alternatives for online root menus

Communication profile	Description of fall back alternatives
FOUNDATION fieldbus	Menus declared in the BLOCK_A MENU_ITEMS attribute, e.g. device_root_menu_aiblock. If block based menus are not defined, then a MENU of style TABLE shall be generated from the BLOCK_A PARAMETERS attribute.
HART	root_menu
PROFIBUS, PROFINET	Menu_Main_Specialist if exists or Menu_Main_Maintenance. These menus include online and offline sub-menus.

4.3.2.2 Diagnostic Root Menu

The diagnostic_root_menu includes views that show the device state, detailed diagnostic information and may include graphical views that show, for example, a valve signature.

4.3.2.3 Process variable Root Menu

The process_variable_root_menu includes views that show process measurements and set points with their quality and important information for process operators, for example, ranges.

4.3.2.4 Device Root Menu

The device_root_menu includes features for a device. These features can be split into process-related and device-specific features. This structure is not required if the number of features is too small for splitting. Submenus that represent any structuring are allowed on the device feature menu. In case of such submenus, the menus that are underneath can be split into process-related and device-specific features.

4.3.2.5 Maintenance Root Menu

The maintenance_root_menu should contain features for maintaining the device during the runtime phase and e.g. showing information about last maintenance inspection.

4.3.3 Offline Root Menu

The offline_root_menu is a menu hierarchy including e.g. data items and methods for offline configuration. It contains, in particular, all of the application-specific parameters of the device and may also contain important read-only and writeable variables. For more information about application-specific parameters, see 9.3. The menu can have offline methods, for example, configuration assistants.

If the offline root menu does not exist, the communication profile specific entry point shall be used (see Table 4).

Table 4 – Fall back alternatives for offline root menus

Communication profile	Description of fall back alternatives
FOUNDATION fieldbus	BLOCK_A PARAMETERS
HART	MENU upload_variables
PROFIBUS, PROFINET	Table_Main_Specialist if exists or Table_Main_Maintenance

4.3.4 Example of EDD menu structure

The EDD example in Figure 1 includes additional menus that can be added to an EDD for PC applications. These menus are additional to the existing menus for the applications. This example is specific for a device with an interface according to HART. Examples for devices with an interface according FOUNDATION fieldbus, PROFIBUS and PROFINET would be very similar.

```

MENU diagnostic_root_menu
{
    LABEL "Diagnostics";
    STYLE MENU;                                /* not required to define STYLE in this case, */
                                                /* because of default STYLE of the root menu */

    ITEMS
    {
        status_window,                        /* menu: style=window */
        self_test                            /* method */
    }
}

MENU status_window
{
    LABEL "Status";
    STYLE WINDOW;
    ITEMS
    {
        standard_diagnostics_page,           /* menu: style=page */
        devspec_diagnostics_page            /* menu: style=page */
    }
}

MENU standard_diagnostics_page
{
    LABEL "Standard";
    STYLE PAGE;
    ITEMS
    {
        device_status                        /* variable */
    }
}

MENU devspec_diagnostics_page
{
    LABEL "Device Specific";
    STYLE PAGE;
    ITEMS
    {
        xmtr_specific_status_1,             /* variable */
        xmtr_specific_status_2             /* variable */
    }
}

METHOD self_test
{
    LABEL "Self Test";
    DEFINITION
    {
        /* elided */
    }
}

MENU maintenance_root_menu
{
    LABEL "Maintenance";
    STYLE MENU;                                /* not required to define STYLE in this case, */
                                                /* because of default STYLE of the root menu */

    ITEMS
    {
        device_mode_dialog,                 /* menu: style=dialog */
        teach_in                            /* method */
    }
}

MENU device_mode_dialog
{
    LABEL "Device Mode";
    STYLE DIALOG;
    ITEMS
    {
        mode_page                            /* menu: style=page */
    }
}

MENU mode_page
{
    LABEL "Process Variables";
    STYLE PAGE;

```

```

ITEMS
{
    transducer_group,      /* menu: style=group */
    function_group         /* menu: style=group */
}
}

MENU transducer_group
{
    LABEL "Transducer";
    STYLE GROUP;
    ITEMS
    {
        trans_target_mode, /* variable */
        trans_actual_mode  /* variable */
    }
}

MENU function_group
{
    LABEL "Function";
    STYLE GROUP;
    ITEMS
    {
        func_target_mode, /* variable */
        func_actual_mode  /* variable */
    }
}

METHOD teach_in
{
    LABEL "Teach-in";
    DEFINITION
    {
        /* elided */
    }
}

MENU process_variables_root_menu
{
    LABEL "Process Variables";
    STYLE MENU;
    /* not required to define STYLE in this case, */
    /* because of default STYLE of the root menu */

    ITEMS
    {
        overview_window, /* menu: style=window */
        primary_vars_window /* menu: style=window */
    }
}

MENU overview_window
{
    LABEL "Overview";
    STYLE WINDOW;
    ITEMS
    {
        process_vars_page /* menu: style=page */
    }
}

MENU process_vars_page
{
    LABEL "Process Variables";
    STYLE PAGE;
    ITEMS
    {
        pressure_group, /* menu: style=group */
        temperature_group /* menu: style=group */
    }
}

MENU pressure_group
{
    LABEL "Pressure";
    STYLE GROUP;
    ITEMS
    {
        pv_digital_value, /* variable */

```

```

        pv_upper_range_value,      /* variable */
        pv_lower_range_value      /* variable */
    }
}

MENU temperature_group
{
    LABEL "Temperature";
    STYLE GROUP;
    ITEMS
    {
        sv_digital_value,          /* variable */
        sv_upper_range_value,      /* variable */
        sv_lower_range_value      /* variable */
    }
}

MENU primary_vars_window
{
    LABEL "Primary Variables";
    STYLE WINDOW;
    ITEMS
    {
        pressure_chart_page,      /* menu: style=page */
        temperature_chart_page    /* menu: style=page */
    }
}

MENU pressure_chart_page
{
    LABEL "Pressure";
    STYLE PAGE;
    ITEMS
    {
        pressure_chart            /* chart */
    }
}

CHART pressure_chart
{
    /* elided */
}

MENU temperature_chart_page
{
    LABEL "Temperature";
    STYLE PAGE;
    ITEMS
    {
        temperature_chart        /* chart */
    }
}

CHART temperature_chart
{
    /* elided */
}

MENU device_root_menu
{
    LABEL "Device";
    STYLE MENU;                      /* not required to define STYLE in this case, */
                                      /* because of default STYLE of the root menu */

    ITEMS
    {
        process_related_window,    /* menu: style=window */
        device_specific_window,    /* menu: style=window */
        master_reset               /* method */
    }
}

MENU process_related_window
{
    LABEL "Process Related";
    STYLE WINDOW;
    ITEMS
    {
        identification_page,      /* menu: style=page */

```

```

        output_info_page          /* menu: style=page */
    }
}

MENU identification_page
{
    LABEL "Identification";
    STYLE PAGE;
    ITEMS
    {
        tag,                      /* variable */
        manufacturer,             /* variable */
        device_type,              /* variable */
        device_revision,          /* variable */
        descriptor,               /* variable */
        message                   /* variable */
    }
}

MENU output_info_page
{
    LABEL "Output Information";
    STYLE PAGE;
    ITEMS
    {
        range_values_group,       /* menu: style=group */
        sensor_limits_group      /* menu: style=group */
    }
}

MENU range_values_group
{
    LABEL "Range Values";
    STYLE GROUP;
    ITEMS
    {
        pv_units,                /* variable */
        pv_urv,                  /* variable */
        pv_lrv                   /* variable */
    }
}

MENU sensor_limits_group
{
    LABEL "Sensor Limits";
    STYLE GROUP;
    ITEMS
    {
        sensor_units,            /* variable */
        upper_sensor_limit,       /* variable */
        lower_sensor_limit        /* variable */
    }
}

MENU device_specific_window
{
    LABEL "Device Specific";
    STYLE WINDOW;
    ITEMS
    {
        identification_page,     /* menu: style=page */
        calibration_page         /* menu: style=page */
    }
}

MENU calibration_page
{
    LABEL "Calibration";
    STYLE PAGE;
    ITEMS
    {
        sensor_limits_group,     /* menu: style=group */
        sensor_trim_group        /* menu: style=group */
    }
}

MENU sensor_trim_group
{

```

```

    LABEL "Sensor Trim";
    STYLE GROUP;
    ITEMS
    {
        upper_sensor_trim_point, /* variable */
        lower_sensor_trim_point, /* variable */
        sensor_trim               /* method */
    }
}

METHOD master_reset
{
    LABEL "Master Reset";
    DEFINITION
    {
        /* elided */
    }
}

```

Figure 1 – EDD example of root menus

4.3.5 User interface

4.3.5.1 Diagnostics

Figure 2 shows an example of an EDD application for diagnostics.

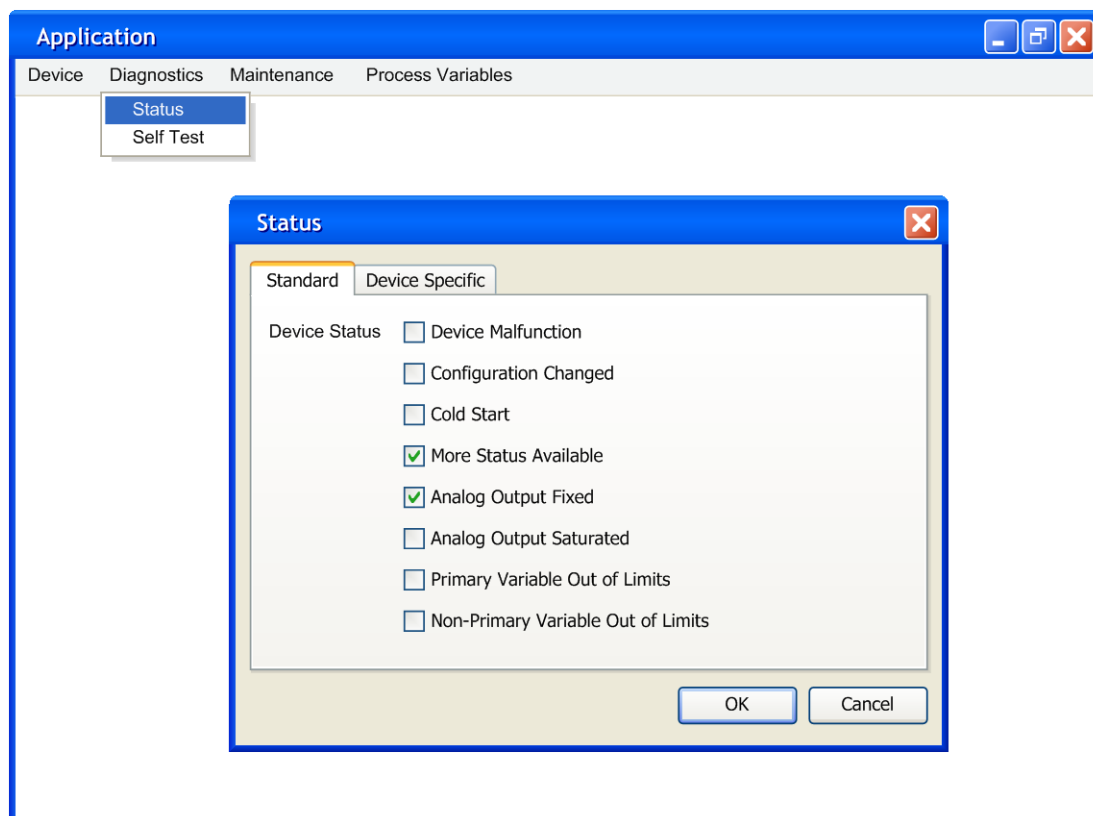


Figure 2 – Example of an EDD application for diagnostics

4.3.5.2 Process variables

Figure 3 and Figure 4 show examples of EDD applications for process variables.

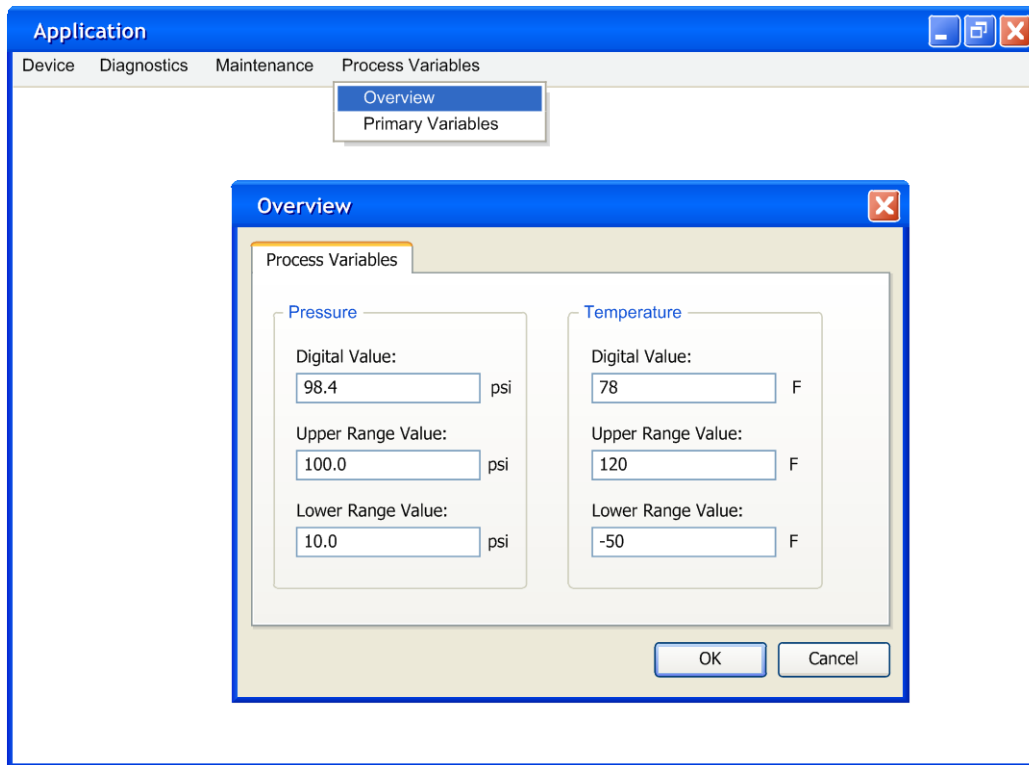


Figure 3 – Example of an EDD application for process variables

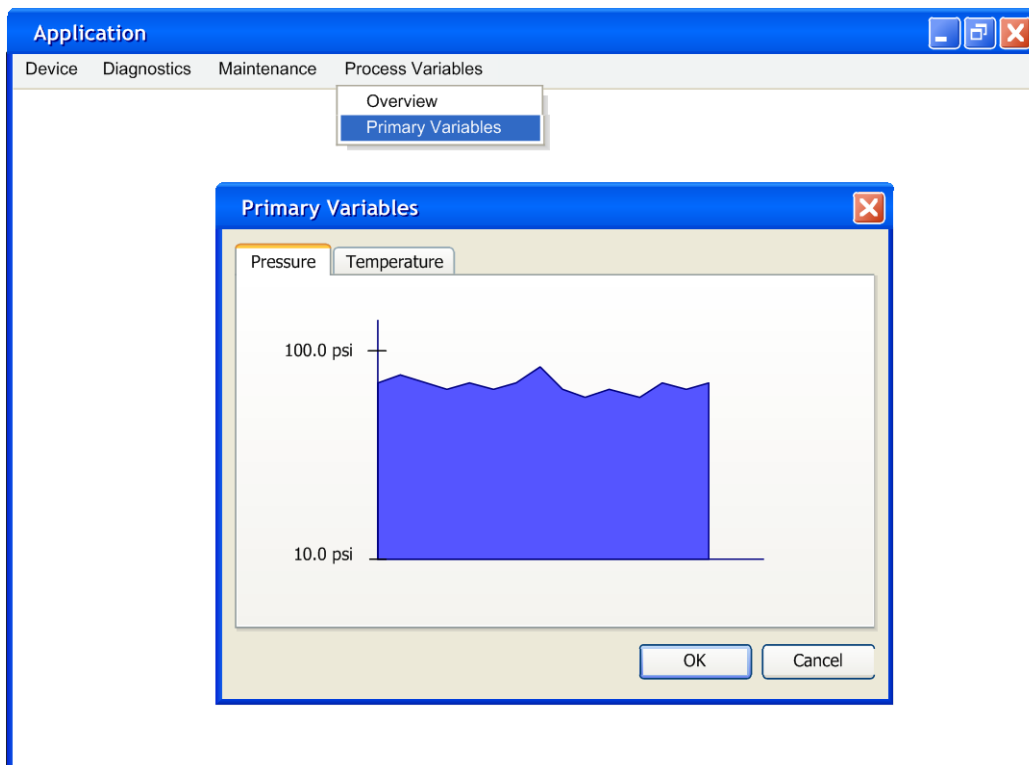


Figure 4 – Example of an EDD application for primary variables

4.3.5.3 Device features

Figure 5, Figure 6 and Figure 7 show examples of EDD applications for device features.

The screenshot shows a software interface with a main window titled 'Application' and a sub-dialog box titled 'Process Related'. The main window has tabs for 'Device', 'Diagnostics', 'Maintenance', and 'Process Variables'. The 'Process Variables' tab is active, and a dropdown menu shows 'Process Related', 'Device Specific', and 'Master Reset'. The 'Process Related' dialog box has two tabs: 'Identification' and 'Output Information'. The 'Output Information' tab is active, showing two sections: 'Range Values' and 'Sensor Limits'. Both sections have a 'Units' dropdown set to 'psi'. The 'Range Values' section has 'Upper Range Value' set to 100.0 and 'Lower Range Value' set to 10.0. The 'Sensor Limits' section has 'Upper Sensor Limit' set to 250 and 'Lower Sensor Limit' set to 0. 'OK' and 'Cancel' buttons are at the bottom.

Figure 5 – Example of an EDD application for process-related device features

The screenshot shows the same 'Application' window, but the 'Device Specific' dialog box is open. The 'Device Specific' dialog box has two tabs: 'Identification' and 'Calibration'. The 'Calibration' tab is active, showing two sections: 'Sensor Limits' and 'Sensor Trim'. The 'Sensor Limits' section has 'Units' set to 'psi', 'Upper Sensor Limit' set to 250, and 'Lower Sensor Limit' set to 0. The 'Sensor Trim' section has 'Upper Trim Point' set to 250 and 'Lower Trim Point' set to 0, with a 'Trim Sensor' button below. 'OK' and 'Cancel' buttons are at the bottom.

Figure 6 – Example of an EDD application for device features

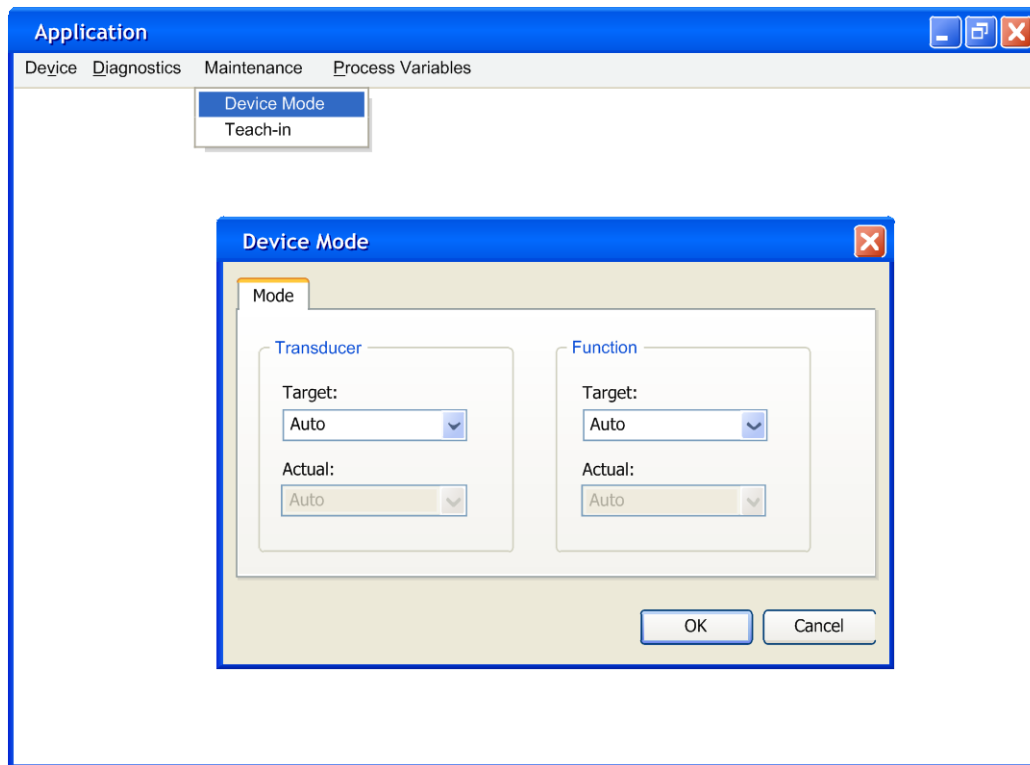


Figure 7 – Example of an EDD application for maintenance features

4.4 Containers and contained items

4.4.1 Overview

The user interface extensions are based on a simple user interface model. The model consists of two concepts:

- containers; and
- contained items.

Containers are so named because they contain other user interface elements. Containers may include: menus, windows, dialogs, tables, pages, groups, and other containers. Containers correspond to a MENU. They are distinguished from one another via the STYLE attribute. This STYLE attribute indicates how the MENU will be displayed.

Contained items include e.g. variables, methods, edit displays, graphs, charts, images, static text.

4.4.2 Containers

4.4.2.1 Permitted and default STYLES

Table 5 defines permitted user interface items in containers, substitutes and default STYLES are not defined. If the style of a menu is not defined, then a default style may be used depending on the menu where the menu is contained and the content of the menu.

The default style of a menu that is not contained in a menu has the style TABLE. The HART root_menu is shown as a parameter table.

General rules:

- If the ACCESS online or offline of a contained MENU is different to the access of the container, the EDD application shall use STYLE WINDOW in a window otherwise shall use STYLE DIALOG.

Table 5 – Permitted contained items and default STYLES

Container	Permitted contained items															Default STYLE of contained items, used if STYLE is not defined
	MENU	WINDOW, DIALOG	PAGE	GROUP	TABLE	EDIT_DISP LAY	VARIABLE	COLLECTIO N, RECORD	ARRAY, LIST	IMAGE, CHART, GRAPH, GRID	static text	METHOD	PLUGIN	COLUMNBR EAK, ROWBREAK	SEPARATO R	
MENU	X	X	D	D	2	X	7	2	2	2	2	X	X	6	6	MENU of STYLE MENU, if no data item is in the menu. DIALOG, if any data item is in the menu.
DIALOG, WINDOW	X	4	X	X	X	4	X	X	X	X	X	4	4	X	8	PAGE
PAGE	X	4	D	X	X	4	X	X	X	X	X	4	4	X	3	GROUP
GROUP	X	4	D	1	D	4	X	1	X	X	X	4	4	X	3	GROUP if the parent of the parent GROUP is not of STYLE GROUP. DIALOG rendered as button if the parent of the parent GROUP is of STYLE GROUP.
TABLE	4	D	D	D	X	4	X	X	X	5	X	4	4	6	6	TABLE
Key																
X Allowed.																
D Default STYLE shall be used (see column “Default STYLE of submenu if not defined”).																
1 Only one sublevel is allowed e.g. GROUP may contain a GROUP but this GROUP shall not contain GROUPs, COLLECTIONs or RECORDs.																
2 Shall be rendered in a dialog.																
3 Shall be interpreted as COLUMNBREAK.																
4 Shall be rendered as button or hyperlink.																
5 Shall be rendered as button, hyperlink or inline.																
6 Shall be rendered as a line or ignored.																
7 Shall be rendered as button, hyperlink or ignored.																
8 SEPARATOR shall be treated as COLUMNBREAK.																

4.4.2.2 Menu

A menu of STYLE MENU takes the form of a drop-down menu, a pop-up menu or a navigation bar.

4.4.2.3 Window

A menu of STYLE WINDOW takes the form of a modeless window.

4.4.2.4 Dialog

A MENU of STYLE DIALOG shall be a modal window. If a dialog contains a subordinated window, the EDD application shall manage the window as a modal subdialog.

4.4.2.5 Table

A menu of STYLE TABLE shall be rendered as a table. Each item shall be represented as one or multiple rows, e.g. ARRAYS. The submenus build a hierarchy, which shall be shown.

The STYLE attribute (STYLE = TABLE) should be defined only on the root menu of a hierarchy.

4.4.2.6 Page

A menu of STYLE PAGE within a WINDOW or DIALOG shall be rendered as a tab.

The EDD application shall interpret ROWBREAKs or other items between pages as horizontal separations.

COLUMNBREAKs and SEPARATORs between pages shall be ignored.

4.4.2.7 Group

A menu of STYLE GROUP takes the form of a group box. In order to maintain a clear arrangement on the user interface, the following restricting rules apply.

- The nesting level of GROUP declarations is restricted to two. In other words, a GROUP may contain further GROUPs but these shall not contain further GROUPs. When a second layer GROUP contains a GROUP, this GROUP shall be rendered as either a button or a hyperlink. The LABEL shall appear on the button or as the hyperlink text. When the button or hyperlink is pressed, a corresponding dialog shall be opened on top of the calling dialog or window.

Within a GROUP, usually logically related parameters and methods are grouped. The title of the GROUP indicates this relation, for example, "AnalogInput 1". Often the same string is also used in the labels of the GROUP items, for example, "AnalogInput1_StaticRevision" or "Analog Input1_Blockmode". In order to avoid this duplication of information, the EDD developer may assemble the items of a GROUP in a COLLECTION in order to override the original labels. If the parameters are to be referred to in a GROUP the parameters can be referenced using the COLLECTION; in other cases, especially without additional semantic context, the parameters can be referenced directly.

Figure 8 shows an EDD example with COLLECTION MEMBERS within a MENU of STYLE GROUP.

```
VARIABLE ail_strev
{
    LABEL "Analog Input 1: Static Revision";
    TYPE INTEGER(4);
}

VARIABLE ail_bloMo
{
    LABEL "Analog Input 1: Block Mode";
    TYPE INTEGER(1);
}

MENU ail_group_double                                /* Leads to double display of */
{
    LABEL "Analog Input 1";
    STYLE GROUP;
    ITEMS
    {
        ail_strev,
        ail_bloMo
    }
}

COLLECTION ail_groupcol
{
    MEMBERS
    {
        strev,ail_strev,"Static Revision";
        bloMo,ail_bloMo,"Block Mode";
    }
}

MENU ail_group_single                                /* Leads to single display of */
{
    LABEL "Analog Input 1";
    STYLE GROUP;
    ITEMS
    {
        ail_groupcol.strev,
        ail_groupcol.bloMo
    }
}
```

Figure 8 – Usage of COLLECTION MEMBERS in MENUs of STYLE GROUP

4.4.3 Contained items

4.4.3.1 Overview

How a container renders contained items is described in 4.4.3. All containers render contained items in the same way. Contained items, such as methods, variables and others, have a LABEL attribute. In order not to disturb the layout of the EDD application, for example, with oversized buttons for methods, the EDD developer should not use very long strings for LABELs. For this reason, some EDD applications may truncate these labels.

4.4.3.2 Methods

A method should be rendered as a button or a hyperlink. The LABEL of the corresponding METHOD should appear on the button or as the hyperlink text. When the button or hyperlink is pressed, the corresponding METHOD should be executed.

4.4.3.3 Variables

4.4.3.3.1 General

The label, value and, if defined, the units of the variable should be displayed in a manner consistent with the definition of the corresponding VARIABLE.

The general handling of variables are as follows.

- HANDLING = READ or the menu item attribute is READ_ONLY: the value of the variable should not be editable.
- CLASS = DYNAMIC: the value should be updated continuously with current read values from the device.

4.4.3.3.2 Variable of TYPE BIT_ENUMERATED

Multiple bits can be packaged in a single-bit enumerated variable. Each bit could have a distinct meaning. It is possible to display each bit separately and, in addition, to display the whole BIT_ENUMERATED variable. BIT_ENUMERATED variables may be displayed as checkboxes.

In case of showing a bit of a BIT_ENUMERATED variable, the variable reference should be extended with a mask in square brackets and the LABEL of the variable is not shown.

Figure 9 shows an EDD example for displaying single bits of a BIT_ENUMERATED variable.

```
VARIABLE var1
{
    LABEL "Var 1";
    TYPE BIT_ENUMERATED
    {
        { 0x1, "Bit 0"},
        { 0x2, "Bit 1"},
        { 0x4, "Bit 2"}
    }
}

MENU diagnostic_info
{
    LABEL "Diagnostic";
    ITEMS
    {
        var1[0x1],    // Bit 0
        var1[0x2],    // Bit 1
        var1[0x4]     // Bit 2
    }
}
```

Figure 9 – Displaying single bits of BIT_ENUMERATED

Figure 10 shows an EDD example that shows the differences between the full display of a BIT_ENUMERATED variable and how it may be displayed if all bits are explicitly addressed.

```
VARIABLE var1
{
    LABEL "Var 1";
    TYPE BIT_ENUMERATED
    {
        { 0x1, "Bit 0"},
        { 0x2, "Bit 1"},
        { 0x4, "Bit 2"}
    }
}

MENU var1_group
{
    LABEL var1.LABEL;
    STYLE GROUP;
    ITEMS
    {
        var1[0x1],    // Bit 0
        var1[0x2],    // Bit 1
        var1[0x4]     // Bit 2
    }
}

MENU diagnostic_info
{
    LABEL "Diagnostic";
    STYLE PAGE;
    ITEMS
    {
        var1,          // all bit should be shown
        COLUMNBREAK,
        var1[0x1],      // Bit 0
        var1[0x2],      // Bit 1
        var1[0x4],      // Bit 2
        COLUMNBREAK,
        var1_group
    }
}
```

Figure 10 – Displaying multiple bits of BIT_ENUMERATED

Figure 11 shows how the result of the Figure 10 EDD example in an EDD application may appear.

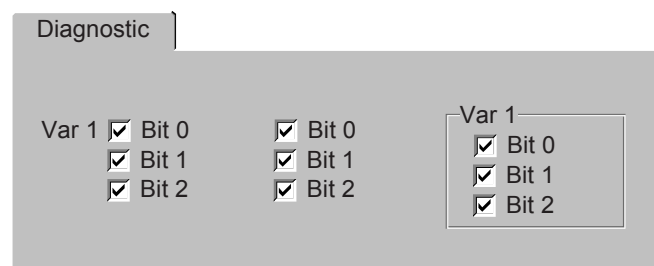


Figure 11 – Example of an EDD application for a variable of type BIT_ENUMERATED

4.4.3.3.3 Variable of TYPE INDEX

When a VARIABLE of type INDEX is presented to the user, the EDD application shall determine the text that shall be displayed by following rules:

- 1) if the description in the referenced ELEMENTS exist then this shall be displayed else;
- 2) if the LABEL of the variable exist then this shall be displayed else:
- 3) the LABEL of the array shall be displayed with the numeric value of the INDEX e.g. myarray[3].

If the INDEX value is out of range of the related array, a text should inform the user that the value is out of range.

If the variable is editable, it should be presented as a combo box.

4.4.3.4 Records and collections

RECORDs and COLLECTIONs shall be displayed as MENU with STYLE GROUP. COLLECTIONs can contain COLLECTIONs. This is handled like nested groups (see 4.4.2.7). The rules about nesting groups apply also in this case.

4.4.3.5 Arrays and lists

Value arrays, reference arrays and LISTs shall be rendered as a scrollable area within the container. The item LABEL shall always be visible.

NOTE HART does not support arrays and lists references in containers.

4.4.3.6 Edit displays

An EDIT_DISPLAY shall be rendered in the container as either a button or a hyperlink. The LABEL of the corresponding EDIT_DISPLAY shall appear on the button or as the hyperlink text.

When the button or hyperlink is activated, the corresponding EDIT_DISPLAY shall be opened as a modal window. The content of an EDIT_DISPLAY contains DISPLAY_ITEMS and EDIT_ITEMS. The EDD application shall display items from DISPLAY_ITEMS above of the items of EDIT_ITEMS. The DISPLAY_ITEMS shall be displayed as read only items and EDIT_ITEMS shall be handled like ITEMS from a MENU of STYLE DIALOG.

4.4.3.7 Graphs

A graph should be displayed in a manner that is consistent with the definition of the corresponding GRAPH. Refer to the layout rules defined in 4.5 and IEC 61804-3 for information on the affect of the WIDTH attribute on the layout of the GRAPH.

4.4.3.8 Charts

A chart should be displayed in a manner that is consistent with the definition of the corresponding CHART. Refer to the layout rules defined in 4.5 and IEC 61804-3 for information on the affect of the WIDTH attribute on the layout of the CHART.

4.4.3.9 Images

The images that are referenced by the MENU should be displayed.

The EDD application shall display an IMAGE in original size, if it is not referenced with a menu item INLINE qualifier. The EDD application shall insert a ROWBREAK above the IMAGE, if items exists before the IMAGE and insert a ROWBREAK below the IMAGE, if items exists after the IMAGE.

The EDD application shall scale down an IMAGE with a menu item INLINE qualifier if the IMAGE width is larger than the column width in which the IMAGE is contained.

The EDD application shall not scale up an IMAGE.

The EDD application shall keep original aspect ratio of IMAGES.

4.4.3.10 Static text

The text that is referenced by the MENU should be displayed. The text will be wrapped in multiple lines, if the text is longer than the width of dialog, window, page, group, or column.

4.4.3.11 Grid

The LABEL of the grid is shown at first and below the grid area with the width of the dialog, window, page, group, or column and the height that is needed to show the data or what is available on the screen. Scroll bars are used to scroll the grid area, if the width or height is too small to show the complete data.

4.5 Layout rules

4.5.1 Overview

Layout rules are rules that the EDD application shall follow for arranging the content of windows and dialogs on the display.

Each container defines the bounding box of the container. This is the area of the container where its contents shall be displayed. For example, the bounding box of a window or dialog is the entire window except the border and title bar.

The ITEMS of a container shall be displayed in the same order they appear in the EDD. The contents of the container should be organized starting at the upper left area of the container. The items shall be displayed vertically down the container. When a COLUMNBREAK is encountered, a new column shall be created. The following items shall be displayed in the next column starting at the top of the container. This process continues until all items have been displayed. Column breaks shall only be introduced when a COLUMNBREAK is encountered. The EDD application shall not introduce columns breaks on its own.

If static text contains carriage return and line feeds, then the text should be wrapped in lines if the text is long.

The ordering of items in unit and refresh relations cause or effect lists are not used for display purpose. If a WRITE_AS_ONE relation makes it necessary to complete the displayed items then these additional items shall appear in the order of the items in the WRITE_AS_ONE relation.

4.5.2 Layout rules for WIDTH and HEIGHT

VARIABLE, CHART, GRAPH and GRID have the attributes WIDTH and HEIGHT. The WIDTH attribute defines the number of columns of display. The HEIGHT attribute defines the vertical size of the display as a multiple of the minimum height for a standard input field, see Table 6. The default value of WIDTH and HEIGHT is MEDIUM for CHART, GRAPH, GRID; XXX_SMALL for VARIABLE height, and width.

The width of the EDD application should be split in up to a maximum of 5 columns depending on the number of COLUMNBREAKs and the WIDTH of the containers. Columns that would span to a column higher than the 5th column shall be moved to the next row through an implied ROWBREAK.

An implied ROWBREAK has the same behaviour as an explicitly defined ROWBREAK.

The WIDTH and HEIGHT attribute of a container defines how many columns the container spans and the height in multiples of minimum height possible for a simple field e.g. variable, constance string, menu and method reference, see Table 6.

Table 6 – WIDTH and HEIGHT span and applicability

WIDTH and HEIGHT Value	WIDTH	HEIGHT	Applicability
XXX_SMALL	1	1	VARIABLE
XX_SMALL	1	3	CHART GRAPH GRID VARIABLE
X_SMALL	1	5	CHART GRAPH GRID VARIABLE
SMALL	2	7	CHART GRAPH GRID VARIABLE
MEDIUM	3	8	CHART GRAPH GRID VARIABLE
LARGE	4	9	CHART GRAPH GRID VARIABLE
X_LARGE	5	11	CHART GRAPH GRID VARIABLE
XX_LARGE	5	13	CHART GRAPH GRID VARIABLE

4.5.3 Layout rules for COLUMNBREAK and ROWBREAK

4.5.3.1 Layout for protruding elements

The EDD application shall insert all the menu items in a single column one after the other until it encounters a COLUMNBREAK. The COLUMNBREAK will cause the next menu item to be inserted in the highest row in the next column. If there is a menu item occupying a region of space in the next column, then the menu item will be inserted in the row immediately below the occupying menu item. See Figure 12 for EDD source code example and Figure 13 for the corresponding layout.

```

MENU protruding_elements
{
    LABEL "protruding_elements";
    STYLE WINDOW;
    ITEMS
    {
        Element1,
        Element2,
        Element3,
        Element4,
        SmallGraph5,      //WIDTH SMALL (2 columns), HEIGHT SMALL (7 height units)
        Element6,
        Element7,
        Element8,
        COLUMNBREAK,
        Element9,
        Element10,
        Element11,
        COLUMNBREAK,
        Element12,
        Element13,
        Element14,
    }
}

```

Figure 12 – EDD source code for layout for protruding elements example

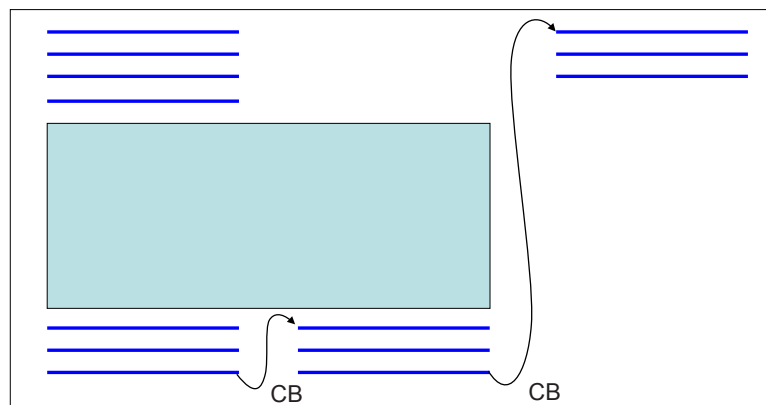


Figure 13 – Layout for protruding elements

4.5.3.2 Layout for partially filled rows

The EDD application shall insert all the menu items in a single column one after the other until it encounters a ROWBREAK. A ROWBREAK will cause the next menu item to be inserted in the lowest row in the 1st column (it acts as a carriage return). The ROWBREAK shall also produce a horizontal barrier so that any further COLUMNBREAK will prevent a menu item from being inserted above this line. See Figure 14 for EDD source code example and Figure 15 for the corresponding layout.

```

MENU Fig13
{
    LABEL "Fig13";
    STYLE WINDOW;
    ITEMS
    {
        Element1,
        Element2,
        Element3,
        COLUMNBREAK,
        Element4,
        Element5,
        Element6,
        Element7,
        ROWBREAK,
        SmallGraph8,    //WIDTH SMALL, HEIGHT SMALL, occupies 2 columns
        Element9,
        Element10,
        Element11,
        COLUMNBREAK,
        Element12,
        Element13,
        Element14,
        COLUMNBREAK,
        Element15,
        Element16,
        Element17,
    }
}
    
```

Figure 14 – EDD source code for layout for partially filled rows example

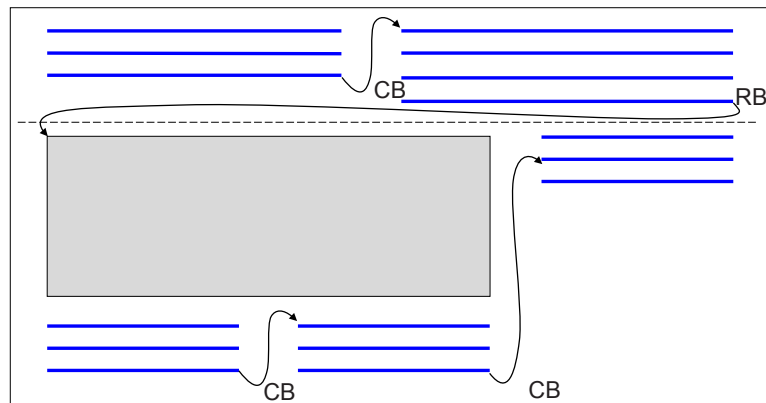


Figure 15 – Layout for partially filled rows

Figure 17 shows that menu items in the rows with two columns (rows 1 and 3) have expanded cell space available such that the row takes up the same width as the rows with three columns. Figure 16 shows the corresponding EDD source code example.

```

MENU partially_filled_rows
{
    LABEL "partially filled rows";
    STYLE WINDOW;
    ITEMS
    {
        Element1,
        Element2,
        COLUMNBREAK,
        Element3,
        Element4,
        ROWBREAK,
        Element5,
        Element6,
        COLUMNBREAK,
        Element7,
        Element8,
        COLUMNBREAK,
        Element9,
        Element10,
        ROWBREAK,
        Element11,
        Element12,
        COLUMNBREAK,
        Element13,
        Element14
    }
}

```

Figure 16 – EDD source code for layout for partially filled rows example

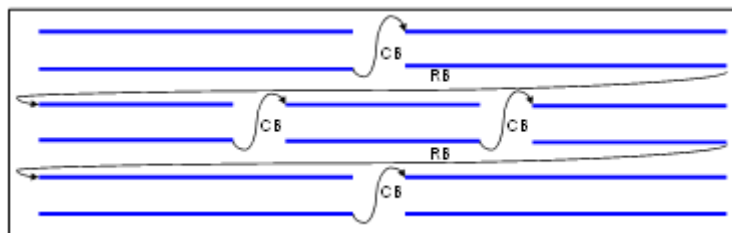


Figure 17 – Layout for partially filled rows

4.5.3.3 Layout for oversized elements

The EDD application shall insert all the menu items in a single column one after the other until it encounters a COLUMNBREAK. The COLUMNBREAK will cause the next menu item to be inserted in the highest row in the next column. If the menu item to be inserted is wider than the number of columns available, then the menu item will be inserted in the lowest row in the 1st column (i.e., implied ROWBREAK). See Figure 18 for EDD source code example and Figure 19 for the corresponding layout.

```

MENU layout_for_oversized_elements
{
    LABEL "layout for oversized elements";
    STYLE WINDOW;
    ITEMS
    {
        Element1,
        Element2,
        Element3,
        COLUMNBREAK,
        Element4,
        Element5,
        Element6,
        COLUMNBREAK,
        Element7,
        Element8,
        Element9,
        COLUMNBREAK, //Results in implied ROWBREAK
        MediumGraph10 //WIDTH MEDIUM, HEIGHT MEDIUM, occupies 3 columns
    }
}
    
```

Figure 18 – EDD source code for layout for oversized elements example

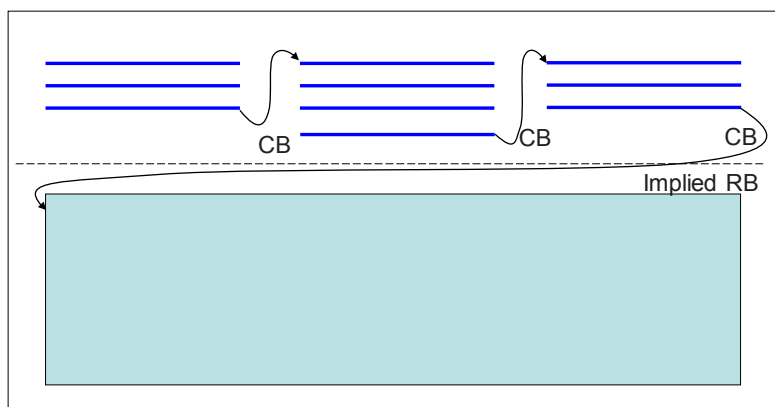


Figure 19 – Layout for oversized elements

4.5.3.4 Layout for columns in stacked group

The EDD application shall insert all the menu items in a single column one after the other until it encounters a COLUMNBREAK. The COLUMNBREAK will cause the next menu item to be inserted in the highest row in the next column. If the menu item resides in a nested menu (such as a GROUP), then the item will remain inside the parent menu. See Figure 20 for EDD source code example and Figure 21 for the corresponding layout.