

DIN EN IEC 61804-4



ICS 35.060; 35.240.50

Einsprüche bis 2021-01-13
Vorgesehen als Ersatz für
DIN EN 61804-4:2016-08

Entwurf

**Funktionsbausteine für die Prozessautomation und elektronische
Gerätebeschreibungssprache –
Teil 4: Interpretation von Gerätebeschreibungen
(IEC 65E/633/CDV:2018);
Englische Fassung prEN IEC 61804-4:2018**

Function blocks (FB) for process control and electronic device description
language (EDDL) –

Part 4: EDD interpretation

(IEC 65E/633/CDV:2018);

English version prEN IEC 61804-4:2018

Blocs fonctionnels (FB) pour les procédés industriels et le langage de description
électronique de produit (EDDL) –

Partie 4: Interprétation EDD

(IEC 65E/633/CDV:2018);

Version anglaise prEN IEC 61804-4:2018

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2020-11-13 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

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

Stellungnahmen werden erbeten

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

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

Gesamtumfang 139 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Die englische Fassung des europäischen Dokuments prEN IEC 61804-4:2018 „Funktionsbausteine für die Prozessautomation und elektronische Gerätebeschreibungssprache – Teil 4: Interpretation von Gerätebeschreibungen“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 65E/633/CDV:2018 „Function blocks (FB) for process control and electronic device description language (EDDL) – Part 4: EDD interpretation“ unverändert in den Entwurf prEN IEC 61804-4:2018 übernommen worden.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und unterliegt dem Copyright der IEC . Das Dokument wurde den nationalen Komitees zur Stellungnahme vorgelegt.

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

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

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DKE (www.dke.de) und DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

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

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

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

Da sich die Benutzer des vorliegenden Norm-Entwurfs der englischen Sprache als Fachsprache bedienen, wird die englische Fassung der IEC 65E/633/CDV veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt und deren Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Norm-Entwurfs hinzugefügt. Für einige der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Dieser Norm-Entwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

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

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. IEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

Änderungen

Gegenüber DIN EN 61804-4:2016-08 wurden folgende Änderungen vorgenommen:

- a) Die Kommunikationsprofile ISA100 und GPE wurden hinzugefügt;
- b) eine Beschreibung der Regeln für ein Layout mit optimierter Spaltenbreite wurde hinzugefügt;
- c) die Beschreibung der Verkettung von Labels und Hilfe wurde hinzugefügt;
- d) Farbbänder für Diagramme vom Typ Meter wurden hinzugefügt.

Nationaler Anhang NA (informativ)

Auszugsweise Übersetzung

Reihenfolge und Inhalt der folgenden Abschnitte sind identisch mit denen in den jeweiligen Abschnitten der Englischen Fassung.

Einleitung

Dieser Teil von IEC 61804

- enthält eine Übersicht über die Anwendung von EDDL;
- gibt Beispiele, die die Anwendung der EDDL-Konstrukte zeigen;
- zeigt, wie die Anwendungsfälle erfüllt werden; und
- zeigt für jedes Beispiel die richtige Interpretation der EDD-Anwendung.

Dieser Teil von IEC 61804 ist keine EDDL-Anleitung und ist nicht dazu vorgesehen, die EDDL-Spezifikation zu ersetzen.

Es werden Anweisungen für die EDD-Anwendung gegeben, die beschreiben, was ausgeführt wird, ohne die für die Host-Implementierung anzuwendende Technologie vorzuschreiben. So beschreibt z. B. das FILE-Konstrukt Daten, die von der EDD-Anwendung im Namen der EDD gespeichert werden. Das FILE-Konstrukt legt nicht fest, wie die Daten gespeichert werden. Die EDD-Anwendung kann je nach Wahl eine Datenbank, eine flache Datei (en: flat file) oder irgendeine andere Implementierung nutzen.

Die EDDL-Funktionen sind für jede der Kommunikationstechnologien durch das Profil begrenzt. Die Beschreibungen in diesem Teil von IEC 61804 beziehen sich auf diese Merkmale in einem allgemeinen Sinn, und nicht alle Kommunikationstechnologien unterstützen alle beschriebenen Merkmale. Um die von den einzelnen Kommunikationstechnologien unterstützten Merkmale zu verstehen, ist auf die Profildefinitionen in IEC 61804-3 zu verweisen.

1 Anwendungsbereich

In diesem Teil von IEC 61804 wird die EDD-Interpretation für EDD-Anwendungen und EDDS zur Unterstützung der EDD-Interoperabilität festgelegt. Dieses Dokument soll sicherstellen, dass Feldgeräteentwickler die EDDL-Konstrukte in gleichbleibender Weise nutzen und dass die EDD-Anwendungen die EDD in gleicher Weise interpretieren. Es ergänzt die EDDL-Spezifikation, um die Interoperabilität von EDDL-Anwendungen zu fördern und die Übertragbarkeit der EDD zwischen EDDL-Anwendungen zu verbessern.

3 Begriffe, Abkürzungen, Akronyme und Konventionen

3.1 Allgemeine Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC 61804-3 und die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Verwendung in der Normung unter den folgenden Adressen bereit:

- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>
- ISO Online Browsing Platform: verfügbar unter <http://www.iso.org/obp>

3.1.1

EDD-Entwickler

(en: EDD developer)

Einzelperson oder Arbeitsgruppe, die eine EDD entwickelt

3.1.2

Container

(en: container)

Benutzerschnittstellenelemente, die andere Benutzerschnittstellenelemente enthalten

Anmerkung 1 zum Begriff: Zu den Containern können Menüs, Fenster, Dialoge, Tabellen, Seiten, Gruppen und andere Container gehören.

3.1.3

enthaltenes Objekt

(en: contained item)

Benutzerschnittstellenelemente, die in Containern enthalten sein können

Anmerkung 1 zum Begriff: Zu den enthaltenen Objekten können Variablen, Methoden, Graphen, Diagramme, Bilder und statischer Text gehören.

3.1.4

Geräteentwickler

(en: device developer)

Einzelperson oder Arbeitsgruppe, die ein Gerät und eine EDD zur Beschreibung dieses Geräts entwickelt

3.1.5

Organizer

(en: handheld)

Gerät mit begrenzter Anzeigenauflösung, die eine Beschränkung für die Benutzerschnittstelle von EDD-Anwendungen darstellt

3.2 Begriffe zu modularen Geräten

3.2.1

Kanal

(en: channel)

Verbindung zu einem Prozess, der gemessen oder gesteuert bzw. geregelt wird

3.2.2

Komponente

(en: component)

Software- oder Hardware-Objekt, das im Konzept des modularen Geräts enthalten ist

Anmerkung 1 zum Begriff: Eine Komponente kann nicht getrennt von dem sie hostenden modularen Gerät funktionieren. Eine Komponente darf einen oder mehrere Typen modularer Geräte unterstützen.

3.2.3

Schnittstelle

(en: interface)

grundlegende Erklärungen von Basiskonstrukten

Anmerkung 1 zum Begriff: Eine Schnittstelle legt alle öffentlichen Teile fest, die von den Komponenten genutzt werden dürfen.

3.2.4

modales Fenster

(en: modal window)

Nachfolgerfenster, mit dem die Benutzer interagieren müssen, bevor sie zum Betrieb der Hauptanwendung zurückkehren können, wodurch der Arbeitsfluss auf dem Anwendungshauptfenster unterbrochen wird

3.2.5

modulares Gerät

(en: modular device)

Gerät, das eine Vielzahl von Software- und Hardware-Komponenten enthalten kann

3.3 Abkürzungen und Akronyme

CP	Kommunikationsprofil (en: Communication Profile)
CPF	Kommunikationsprofilfamilie (en: Communication Profile Family)
CS	Kommunikationsserver (en: Communication Server)
EDD	elektronische Gerätebeschreibung (en: Electronic Device Description)
EDDL	elektronische Gerätebeschreibungssprache (en: Electronic Device Description Language)
FDI	Feldgeräteintegration (en: field device integration)
FF	Foundtion Fieldbus (en: Foundtion fieldbus)
GPE	Allgemeine Protokoll-Erweiterungen (en: Generic Protocol Extensions)
HART	Highway Addressable Remote Transducer (en: Highway Addressable Remote Transducer)

ISA100	ISA100 (en: ISA100_Wireless)
PB	PROFIBUS (en: PROFIBUS)
PC	Arbeitsplatzrechner (en: Personal Computer)
PI	PROFIBUS und PROFINET International (en: PROFIBUS and PROFINET International)
PI PROFILE PA	PI-spezifisches Profil für zur Prozessautomation dienende Feldgeräte (en: PI specific profile for Process Automation field devices)
PN	PROFINET (en: PROFINET)

3.4 Konventionen

Je nachdem, welches Kommunikationsnetz genutzt und welches Gerätemodell angewendet wird, gibt es einige Unterschiede bei der Anwendung und Interpretation von EDDL. Die verschiedenen Kommunikationsnetze, die bei Anwendung dieses Teils von IEC 61804 genutzt werden sind:

- HART (nach IEC 61784-1, CPF9);
- FOUNDATION Fieldbus (nach IEC 61784-1, CPF1);
- PROFIBUS (nach IEC 61784-1, CPF3);
- PROFINET (nach IEC 61784-2, CPF3).
- Kommunikationsserver
 - IEC 62769-4 Feldgeräteintegration (FDI) – Teil 4: FDI-Packages
 - IEC 62769-7 Feldgeräteintegration (FDI) – Teil 7: FDI-Kommunikationsgeräte
- ISA100_Wireless (nach IEC 62734)
- GPE (Allgemeine Protokoll-Erweiterungen)
 - IEC 62769-4 Feldgeräteintegration (FDI) – Teil 4: FDI-Packages
 - IEC 62769-7 Feldgeräteintegration (FDI) – Teil 7: FDI-Kommunikationsgeräte

Die in diesem Dokument angegebenen EDD-Beispiele zeigen Teile von EDD-Implementierungen und sind nicht vollständig.

EDDL-Schlüsselwörter sind in Großbuchstaben geschrieben. Falls mehrere Grundkonstruktelemente gemeint sind, wird das Schlüsselwort in Großbuchstaben geschrieben und ein kleines „s“ angehängt, z.B. VARIABLES.

Das mit einem Großbuchstaben beginnende Wort Builtin bezieht sich auf in IEC 61804-5 festgelegte Funktionen.

BEISPIEL Builtin MenuDisplay bezieht sich auf das in IEC 61804-5 festgelegte Builtin mit dem Namen MenuDisplay.

– Leerseite –

CONTENTS

1	Scope	6
2	Normative references	6
3	Terms, definitions, abbreviated terms acronyms and conventions	6
3.1	General terms and definitions	6
3.2	Terms and definitions related to modular devices	7
3.3	Abbreviated terms and acronyms	7
3.4	Conventions	8
4	EDDL user interface description	8
4.1	Overview	8
4.2	Menu conventions for handheld applications	8
4.3	Menu conventions for PC-based applications	9
4.3.1	Overview	9
4.3.2	Online Root Menus	9
4.3.3	Offline Root Menu	10
4.3.4	Example of EDD menu structure	11
4.3.5	User interface	15
4.4	Label Concatenation for Indirect Variable References	18
4.4.1	General	18
4.4.2	Simple variable references	19
4.4.3	Complex variable references	19
4.5	Help Concatenation	21
4.5.1	General	21
4.5.2	Simple variable references	21
4.5.3	Complex variable references	21
4.6	Containers and contained items	22
4.6.1	Overview	22
4.6.2	Permitted and default STYLES	23
4.6.3	Containers	24
4.6.4	Contained items	26
4.7	Layout rules	32
4.7.1	Overview	32
4.7.2	Controlling the layout by LAYOUT_TYPE attribute	34
4.7.3	Layout rules for WIDTH and HEIGHT	37
4.7.4	Layout rules for COLUMNBREAK and ROWBREAK	40
4.7.5	Layout examples	46
4.7.6	Conditional user interface	60
4.8	Graphical elements	65
5	EDDL data description	69
5.1	EDDL application stored device data	69
5.1.1	Overview	69
5.1.2	FILE	69
5.1.3	LIST	71
5.2	Exposing data items outside the EDD application	78
5.3	Initialization of EDD instances	78
5.3.1	Overview	78
5.3.2	Initialization support	78

9.4	Online dataset	105
9.5	Online configuration	105
9.6	Upload and download	105
9.6.1	Overview	105
9.6.2	Error recovery.....	106
9.6.3	Upload procedure	107
9.6.4	Download procedure	108
10	EDDL communication description	111
10.1	General	111
10.2	Parsing data received from the device	111
10.3	Parsing complex data items.....	111
10.4	Foundation Fieldbus	112
10.5	ISA100_Wireless communication model	116
Annex A	(normative) Device simulation	120
Annex B	(informative) Predefined identifiers	121
Annex C	(informative) Description of EDDL Profiles	124
C.1	Communication Server (CS)	124
C.2	Foundation Fieldbus (FF)	124
C.3	Generic Protocol Extension (GPE).....	124
C.4	HART	124
C.5	ISA100	124
C.6	PROFIBUS (PB)	124
C.7	PROFINET (PN)	124
Annex D	(normative) Upload/Download Caching Model.....	125
Figure 1	– EDD example of root menus.....	15
Figure 2	– Example of an EDD application for diagnostics	15
Figure 3	– Example of an EDD application for process variables	16
Figure 4	– Example of an EDD application for primary variables	16
Figure 5	– Example of an EDD application for process-related device features	17
Figure 6	– Example of an EDD application for device features	17
Figure 7	– Example of an EDD application for maintenance features.....	18
Figure 8	– Usage of COLLECTION MEMBERS in MENUs of STYLE GROUP	26
Figure 9	– Displaying single bits of BIT_ENUMERATED	28
Figure 10	– Displaying multiple bits of BIT_ENUMERATED.....	28
Figure 11	– Example of an EDD application for a variable of type BIT_ENUMERATED	29
Figure 12	– EDD example with a "write-only" variable (HANDLING WRITE)	30
Figure 13	– Basic Layout elements	33
Figure 14	– Example of layout with equal column width.....	34
Figure 15	– Example of layout with optimized column width	35
Figure 16	– Cell body in a layout with optimized column width (Label to the left).....	35
Figure 17	– Cell body in a layout with optimized column width (Label on top).....	36
Figure 18	– EDD source code for a layout with VARIABLES spanning columns	39
Figure 19	– Layout with VARIABLES spanning multiple columns	39
Figure 20	– EDD source code for layout for protruding elements example.....	41

Figure 21 – Layout for protruding elements	41
Figure 22 – EDD source code for layout for partially filled rows example.....	42
Figure 23 – Layout for partially filled rows	42
Figure 24 – EDD source code for layout for partially filled rows example.....	43
Figure 25 – Layout for partially filled rows	43
Figure 26 – EDD source code for layout for oversized elements example.....	44
Figure 28 – Oversized element in a layout with optimized column width	44
Figure 29 – EDD source code example for a layout for columns in stacked group	45
Figure 30 – Layout for columns in stacked group	45
Figure 31 – EDD source code for layout for columns with GRAPHS in stacked group example	46
Figure 32 – Layout for columns with GRAPHS in stacked group	46
Figure 33 – Example of an EDD for an overview menu	47
Figure 34 – Example of an EDD application for an overview window	47
Figure 35 – EDD source code for a layout with menu items spanning a single column	47
Figure 36 – Example of a layout with menu items spanning a single column	48
Figure 37 – Example of an EDD using COLUMNBREAK	48
Figure 38 – Example of an EDD application for an overview window	49
Figure 39 – EDD example for an overview window	49
Figure 40 – Example of an EDD application for an overview window	50
Figure 41 – EDD source code for a layout with small in-line images	50
Figure 42 – Example of a layout with small in-line images	51
Figure 43 – EDD source code for a multi-column layout with GROUP	52
Figure 44 – Example of a multi-column layout with GROUP	53
Figure 45 – Example of an EDD for in-line graphs and charts	53
Figure 46 – Example of an EDD application for an in-line graph	54
Figure 47 – Example of an EDD for full-width graphs and charts	54
Figure 48 – Example of an EDD application for a full-width graph in a layout with equal column width.....	55
Figure 49 – Example of an EDD application for a full-width graph in a layout with optimized column width	56
Figure 50 – Example of an EDD for nested containers	57
Figure 51 – Example of an EDD application for nested containers.....	57
Figure 52 – Example of an EDD for EDIT_DISPLAYS	58
Figure 53 – Example of an EDD application for EDIT_DISPLAYS	58
Figure 54 – Example of an EDD for images.....	58
Figure 55 – Example of an EDD application for images	59
Figure 56 – Example of an EDD for large inline-images	59
Figure 57 – Example of layout with a large inline-image	60
Figure 58 – EDD example for VALIDITY in online session.....	61
Figure 59 – Example of an EDD application for a gauge with limit regions.....	66
Figure 60 – Example of an EDD for a gauge with limit regions	68
Figure 61 – Example of a file declaration	70
Figure 62 – Example of comparing valve signatures.....	71

Figure 63 – Example of more complex file declaration.....	72
Figure 64 – Example of reviewing the stored radar signals.....	73
Figure 65 – Example of an EDD that inserts, replaces, or compares radar signals	78
Figure 66 – Example of a BLOCK_A	79
Figure 67 – Example of a wizard	84
Figure 68 – The different relations of a module	87
Figure 69 – Components and possible configuration of the modular devices	87
Figure 70 – Separate EDD file example with direct EDD referencing	88
Figure 71 – EDD example for module1	89
Figure 72 – EDD example for module2	89
Figure 73 – EDD example for modular device	90
Figure 74 – EDD example for module1	91
Figure 75 – EDD example for module2.....	91
Figure 76 – EDD example for module2.....	93
Figure 77 – Upload/download order of a modular device	93
Figure 78 – Example of a SCAN METHOD	95
Figure 79 – Example of a DETECT METHOD.....	96
Figure 80 – Example of a CHECK_CONFIGURATION METHOD	97
Figure 81 – Data caching for an online session	99
Figure 82 – Data caching for an offline 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 or window	101
Figure 87 – Data caching for a METHOD invoking a dialog using an edit cache	102
Figure 88 – Data caching for a METHOD invoking a dialog	102
Figure 89 – Data flow for download to the device	106
Figure 90 – Data flow for upload from the device	106
Figure 91 – Example device with 2 unique BLOCK_A definitions.....	113
Figure 92 – Example EDD for a device with 2 unique BLOCK_A definitions	114
Figure 93 – BLOCK_A example with PARAMETER_LISTS.....	115
Figure 94 – Example EDD for a BLOCK_A with PARAMETER_LISTS	116
Figure 95 – Example ISA100_Wireless device objects representation	117
Figure 96 – Example EDD for a ISA100_Wireless device with 2 unique BLOCK_A definitions	118
Figure 97 – BLOCK_A example with PARAMETER_LISTS.....	118
Figure 98 – Example EDD for a BLOCK_A with PARAMETER_LISTS	119
Figure D.1 – Upload caching model	125
Figure D.2 – Download caching model	126
Table 1 – List of defined root menu identifiers for handhelds.....	9
Table 2 – List of defined root menu identifiers for PC-based devices.....	9
Table 3 – Fall back alternatives for online root menus.....	10

Table 4 – Fall back alternatives for offline root menus.....	10
Table 5 – Label rule summary for simple variable references	19
Table 6 – Label rule summary for simple variable references	19
Table 7 – Prefix rule summary for complex variable references	20
Table 8 – Prefix rule summary for complex variable references	20
Table 9 – Body rule summary for complex variable references	20
Table 10 – Body rule summary for complex variable references	20
Table 11 – Suffix rule summary for complex variable references	20
Table 12 – Suffix rule summary for complex variable references	21
Table 13 – Help rule summary for simple variable references	21
Table 14 – Help rule summary for simple variable references	21
Table 15 – Help Prefix rule summary for complex variable references	22
Table 16 – Help Prefix rule summary for complex variable references	22
Table 17 – Help Suffix rule summary for complex variable references	22
Table 18 – Help Suffix rule summary for complex variable references	22
Table 19 – Permitted contained items and default STYLES	23
Table 20 – Uninitialized state of VARIABLES on user interface	27
Table 21 – Example of "write-only" variable in an online dialog	30
Table 22 – Description of layout content	33
Table 23 – Minimum and maximum width for input fields spanning one column	36
Table 24 – WIDTH and HEIGHT span and applicability	38
Table 25 – Example 1 VALIDITY in an online session	62
Table 26 – Example 2 VALIDITY in an online session	62
Table 27 – Example 3 VALIDITY in an online session	63
Table 28 – Example 4 VALIDITY in an online session	64
Table 29 – Examples of floating-point results	85
Table 30 – Usages of COMPONENT_PATH.....	86
Table 31 – Diagnostic classifications	94
Table 32 – Terminology for session management	97
Table 33 – Terminology used in data management	97
Table 34 – Builtins for method cache controlling	102
Table 35 – List of defined upload menu identifiers	107
Table 36 – List of defined download menu identifiers	109
Table B.1 – ARRAY predefined identifiers	121
Table B.2 – COLLECTION predefined identifiers.....	121
Table B.3 – COMMAND predefined identifiers.....	121
Table B.4 – IMAGE predefined identifiers.....	121
Table B.5 – MENU predefined identifiers.....	121
Table B.6 – METHOD predefined identifiers.....	122
Table B.7 – VARIABLE predefined identifiers.....	123

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 and control.

This second edition cancels and replaces the first edition published in 2015. This edition constitutes a technical revision.

This edition was developed by merging material from multiple variants of existing EDDL specifications including those from FieldComm Group (Foundation™ Fieldbus¹, HART®²),

¹ FOUNDATION™ Fieldbus is the trademark of FieldComm Group. 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 FieldComm Group. 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.

51 PROFIBUS™³ Nutzerorganisation e.V. (PNO), and ISA100_Wireless™⁴ Compliance Institute
52 (ISA100 WCI). When a profile deviation exists, it is now indicated in the context where the
53 related deviation is found. As a result, the formatting and numbering of this edition may be
54 different from any of the individual specifications from which this edition was derived.

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

- 57 • Communication profiles ISA100 and GPE were added.
- 58 • Description of rules for optimized-column-width layout have been added.
- 59 • Description of the concatenation of labels and help was added.
- 60 • Color banding for meter type charts was added.

61

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

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

63

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

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

67 A list of all parts in the IEC 61804 series, published under the general title *Function blocks*
68 *(FB) for process control and Electronic Device Description Language (EDDL)*, can be found on
69 the IEC website.

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

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

- 75 • reconfirmed,
- 76 • withdrawn,
- 77 • replaced by a revised edition, or
- 78 • amended.

79

80 The National Committees are requested to note that for this publication the stability date
81 is 2023.

82 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE
83 DELETED AT THE PUBLICATION STAGE.

84

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.

⁴ ISA100_Wireless™ is the trademark of ISA100 Wireless Compliance Institute. 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.

85 **IMPORTANT – The ‘colour inside’ logo on the page of this publication indicates that it**
86 **contains colour which are considered to be useful for the correct understanding. Users**
87 **should therefore print this document using a colour printer.**

88

INTRODUCTION

89 This part of IEC 61804

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

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

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

101 EDDL features are limited by profile for each of the communication technologies. The
102 descriptions in this part of IEC 61804 refer to these features in a general sense and not all
103 communication technologies will support all of the features described. The profile definitions
104 in IEC 61804-3 shall be referred to in order to understand the features supported by each
105 communication technology.

106

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 are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

IEC 61804-5, *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

3.1 General terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

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.

- 149 **3.1.3**
150 **contained item**
151 user interface elements that can be contained in containers
- 152 Note 1 to entry: Contained items can include variables, methods, graphs, charts, images, static text.
- 153 **3.1.4**
154 **device developer**
155 individual or team that develops a device and an EDD that describes the device
- 156 **3.1.5**
157 **handheld**
158 device with limited display resolution that restricts EDD applications user interface
- 159 **3.2 Terms and definitions related to modular devices**
- 160 **3.2.1**
161 **channel**
162 connection to a process that is being measured or controlled
- 163 **3.2.2**
164 **component**
165 software or hardware item contained within the modular device concept
- 166 Note 1 to entry: A component cannot function separately from a modular device hosting it. A component may
167 support one or more types of modular devices.
- 168 **3.2.3**
169 **interface**
170 basic declarations of basic constructs
- 171 Note 1 to entry: An interface defines all public parts that components may use.
- 172 **3.2.4**
173 **modal window**
174 a child window that requires users to interact with it before they can return to operating the
175 parent application, thus preventing the workflow on the application main window
- 176 **3.2.5**
177 **modular device**
178 device that can contain a variety of software and or hardware components
- 179 **3.3 Abbreviated terms and acronyms**
- | | |
|--------|--|
| CP | Communication Profile |
| CPF | Communication Profile Family |
| CS | Communication Server |
| EDD | Electronic Device Description |
| EDDL | Electronic Device Description Language |
| FDI | Field Device Integration |
| FF | Foundation Fieldbus |
| GPE | Generic Protocol Extension |
| HART | Highway Addressable Remote Transducer |
| ISA100 | ISA100_Wireless™ |
| PB | PROFIBUS |
| PC | Personal computer |

PI PROFIBUS and PROFINET International
 PI PROFILE PA PI specific profile for Process Automation field devices
 PN PROFINET

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 CPF9)
- FOUNDATION fieldbus (according to IEC 61784-1 CPF1)
- PROFIBUS (according to IEC 61784-1 CPF3)
- PROFINET (according to IEC 61784-2 CPF3)
- Communication Server
 - IEC 62769-4 FDI Technical Specification: Part 4: FDI Packages
 - IEC 62769-7 FDI Technical Specification: Part 7: Communication Devices
- ISA100_Wireless (according to IEC 62734)
- GPE (Generic Protocol Extension)
 - IEC 62769-4 FDI Technical Specification: Part 4: FDI Packages

IEC 62769-7 FDI Technical Specification: Part 7: Communication Devices 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 61804-2:2004. 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). In order to fit in a

223 limited display space, a handheld application may render all menus using STYLE TABLE.
 224 Handheld applications shall render references to images in MENUs of STYLE TABLE always
 225 as hyperlink.

226 **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*	TABLE	Legacy root menu prefix for BLOCK_A MENU_ITEMS for FOUNDATION fieldbus and ISA100_Wireless devices e.g. Menu_Top_TB
offline_root_menu	TABLE	Offline configuration. Handhelds may optionally support this menu.
hh_process_variables_root_menu	TABLE	Process variable views for FOUNDATION fieldbus and HART devices.
hh_device_root_menu	TABLE	Device feature views for set up for FOUNDATION fieldbus and HART devices.
hh_diagnostic_root_menu	TABLE	Diagnostic views for FOUNDATION fieldbus and HART devices.

227
 228 **4.3 Menu conventions for PC-based applications**

229 **4.3.1 Overview**

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

236 **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

237
 238 For HART EDDs the EDD application shall display offline_root_menu with a default STYLE
 239 WINDOW if STYLE is not defined.

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

242 The EDDs should contain the online root menus from Table 2. The online root menus are
 243 optional for PROFIBUS, PROFINET and FOUNDATION fieldbus. The offline root menu is
 244 optional for FOUNDATION fieldbus and ISA100_Wireless. HART EDDs shall have root_menu,
 245 in addition to all root menus in Table 2. GPE shall have at least one Online menu. GPE shall
 246 have the offline_root_menu. Communication Server shall have at least one Online menu

247 **4.3.2 Online Root Menus**

248 **4.3.2.1 General**

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

Table 3 – Fall back alternatives for online root menus

Communication profile	Description of fall back alternatives
FF, ISA100	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, in STYLE TABLE.
PB, PN	Menu_Main_Specialist if exists or Menu_Main_Maintenance. These menus include online and offline sub-menus. These would be rendered as STYLE MENU.

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 variables Root Menu

The process_variables_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
FF, ISA100	BLOCK_A PARAMETERS
HART	MENU upload_variables
PB, PN	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, ISA100_Wireless, PROFIBUS, PROFINET, Communication Server profile and GPE profile 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;

```

```

353     ITEMS
354     {
355         mode_page                /* menu: style=page */
356     }
357 }
358
359 MENU mode_page
360 {
361     LABEL "Process Variables";
362     STYLE PAGE;
363     ITEMS
364     {
365         transducer_group,        /* menu: style=group */
366         function_group           /* menu: style=group */
367     }
368 }
369
370 MENU transducer_group
371 {
372     LABEL "Transducer";
373     STYLE GROUP;
374     ITEMS
375     {
376         trans_target_mode,       /* variable */
377         trans_actual_mode       /* variable */
378     }
379 }
380
381 MENU function_group
382 {
383     LABEL "Function";
384     STYLE GROUP;
385     ITEMS
386     {
387         func_target_mode,        /* variable */
388         func_actual_mode        /* variable */
389     }
390 }
391
392 METHOD teach_in
393 {
394     LABEL "Teach-in";
395     DEFINITION
396     {
397         /* elided */
398     }
399 }
400
401 MENU process_variables_root_menu
402 {
403     LABEL "Process Variables";
404     STYLE MENU;
405     /* not required to define STYLE in this case, */
406     /* because of default STYLE of the root menu */
407
408     ITEMS
409     {
410         overview_window,        /* menu: style=window */
411         primary_vars_window     /* menu: style=window */
412     }
413 }
414
415 MENU overview_window
416 {
417     LABEL "Overview";
418     STYLE WINDOW;
419     ITEMS
420     {
421         process_vars_page       /* menu: style=page */
422     }
423 }
424
425 MENU process_vars_page
426 {
427     LABEL "Process Variables";
428     STYLE PAGE;
429     ITEMS
430     {
431         pressure_group,         /* menu: style=group */

```

```

431         temperature_group          /* menu: style=group */
432     }
433 }
434
435 MENU pressure_group
436 {
437     LABEL "Pressure";
438     STYLE GROUP;
439     ITEMS
440     {
441         pv_digital_value,           /* variable */
442         pv_upper_range_value,       /* variable */
443         pv_lower_range_value       /* variable */
444     }
445 }
446
447 MENU temperature_group
448 {
449     LABEL "Temperature";
450     STYLE GROUP;
451     ITEMS
452     {
453         sv_digital_value,           /* variable */
454         sv_upper_range_value,       /* variable */
455         sv_lower_range_value       /* variable */
456     }
457 }
458
459 MENU primary_vars_window
460 {
461     LABEL "Primary Variables";
462     STYLE WINDOW;
463     ITEMS
464     {
465         pressure_chart_page,       /* menu: style=page */
466         temperature_chart_page    /* menu: style=page */
467     }
468 }
469
470 MENU pressure_chart_page
471 {
472     LABEL "Pressure";
473     STYLE PAGE;
474     ITEMS
475     {
476         pressure_chart             /* chart */
477     }
478 }
479
480 CHART pressure_chart
481 {
482     /* elided */
483 }
484
485 MENU temperature_chart_page
486 {
487     LABEL "Temperature";
488     STYLE PAGE;
489     ITEMS
490     {
491         temperature_chart          /* chart */
492     }
493 }
494
495 CHART temperature_chart
496 {
497     /* elided */
498 }
499
500 MENU device_root_menu
501 {
502     LABEL "Device";
503     STYLE MENU;
504     /* not required to define STYLE in this case, */
505     /* because of default STYLE of the root menu */
506
507     ITEMS
508     {
509         process_related_window,    /* menu: style=window */

```

```

509         device_specific_window, /* menu: style=window */
510         master_reset             /* method */
511     }
512 }
513
514 MENU process_related_window
515 {
516     LABEL "Process Related";
517     STYLE WINDOW;
518     ITEMS
519     {
520         identification_page, /* menu: style=page */
521         output_info_page    /* menu: style=page */
522     }
523 }
524
525 MENU identification_page
526 {
527     LABEL "Identification";
528     STYLE PAGE;
529     ITEMS
530     {
531         tag, /* variable */
532         manufacturer, /* variable */
533         device_type, /* variable */
534         device_revision, /* variable */
535         descriptor, /* variable */
536         message /* variable */
537     }
538 }
539
540
541 MENU output_info_page
542 {
543     LABEL "Output Information";
544     STYLE PAGE;
545     ITEMS
546     {
547         range_values_group, /* menu: style=group */
548         sensor_limits_group /* menu: style=group */
549     }
550 }
551
552 MENU range_values_group
553 {
554     LABEL "Range Values";
555     STYLE GROUP;
556     ITEMS
557     {
558         pv_units, /* variable */
559         pv_urv, /* variable */
560         pv_lrv /* variable */
561     }
562 }
563
564 MENU sensor_limits_group
565 {
566     LABEL "Sensor Limits";
567     STYLE GROUP;
568     ITEMS
569     {
570         sensor_units, /* variable */
571         upper_sensor_limit, /* variable */
572         lower_sensor_limit /* variable */
573     }
574 }
575
576 MENU device_specific_window
577 {
578     LABEL "Device Specific";
579     STYLE WINDOW;
580     ITEMS
581     {
582         identification_page, /* menu: style=page */
583         calibration_page /* menu: style=page */
584     }
585 }
586

```

```

587 MENU calibration_page
588 {
589     LABEL "Calibration";
590     STYLE PAGE;
591     ITEMS
592     {
593         sensor_limits_group,      /* menu: style=group */
594         sensor_trim_group         /* menu: style=group */
595     }
596 }
597
598 MENU sensor_trim_group
599 {
600     LABEL "Sensor Trim";
601     STYLE GROUP;
602     ITEMS
603     {
604         upper_sensor_trim_point, /* variable */
605         lower_sensor_trim_point, /* variable */
606         sensor_trim               /* method */
607     }
608 }
609
610 METHOD master_reset
611 {
612     LABEL "Master Reset";
613     DEFINITION
614     {
615         /* elided */
616     }
617 }

```

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.

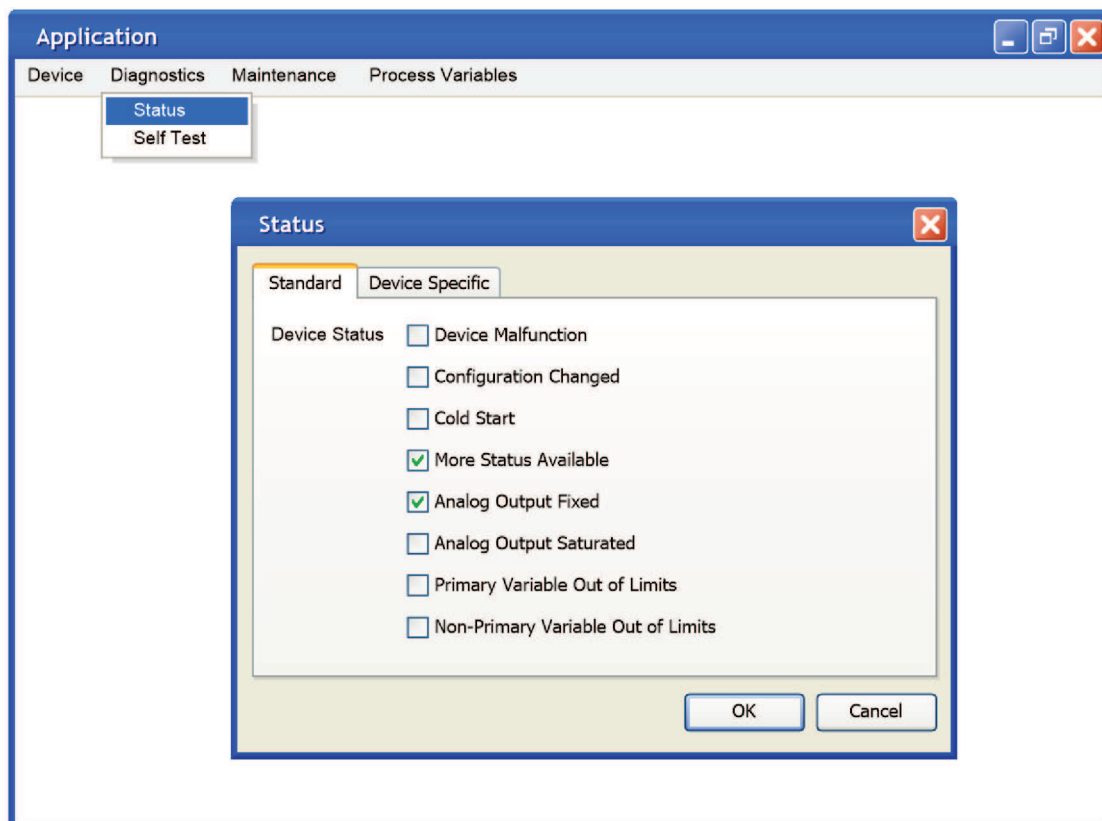


Figure 2 – Example of an EDD application for diagnostics

4.3.5.2 Process variables

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

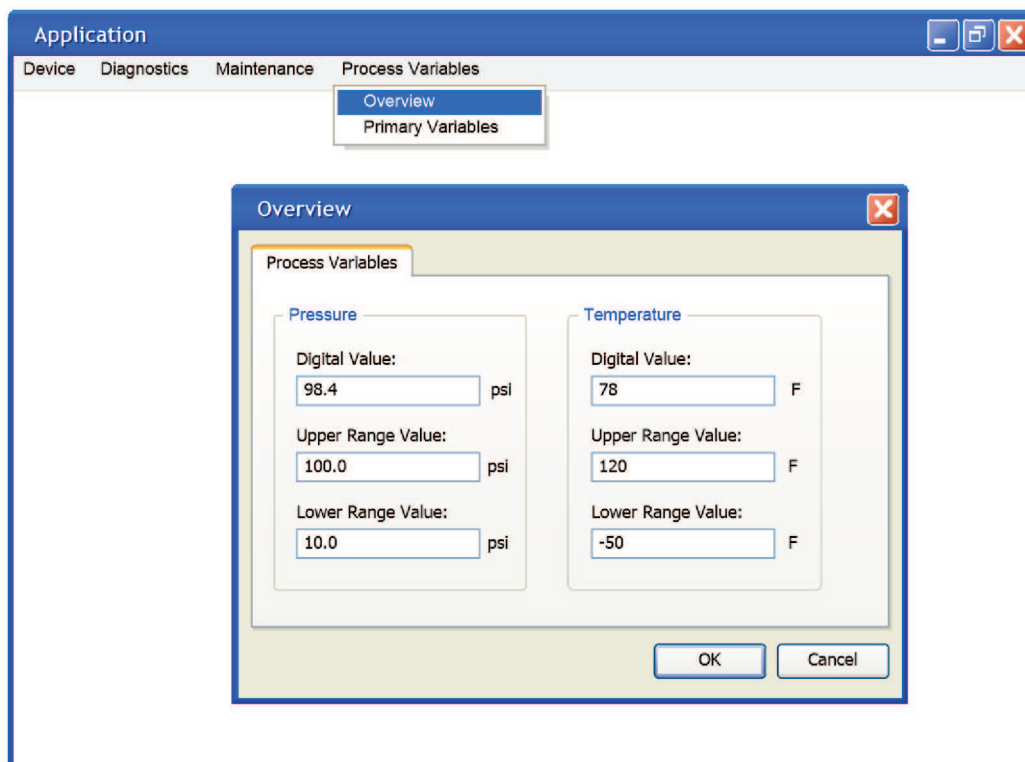


Figure 3 – Example of an EDD application for process variables

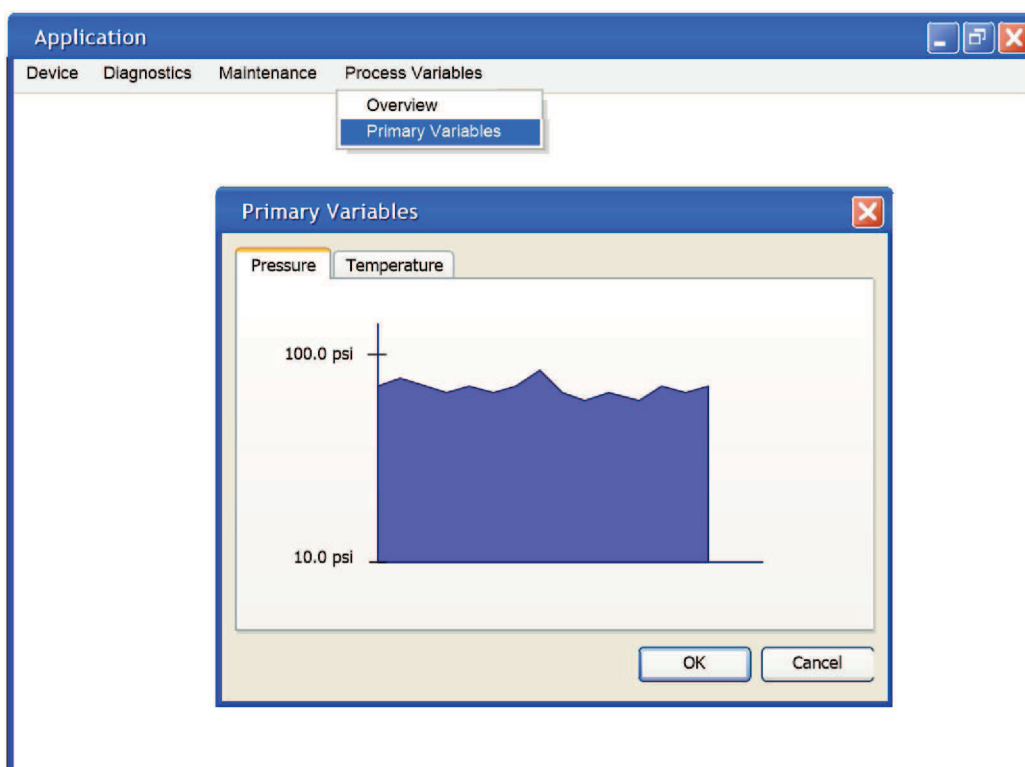


Figure 4 – Example of an EDD application for primary variables

4.3.5.3 Device features

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

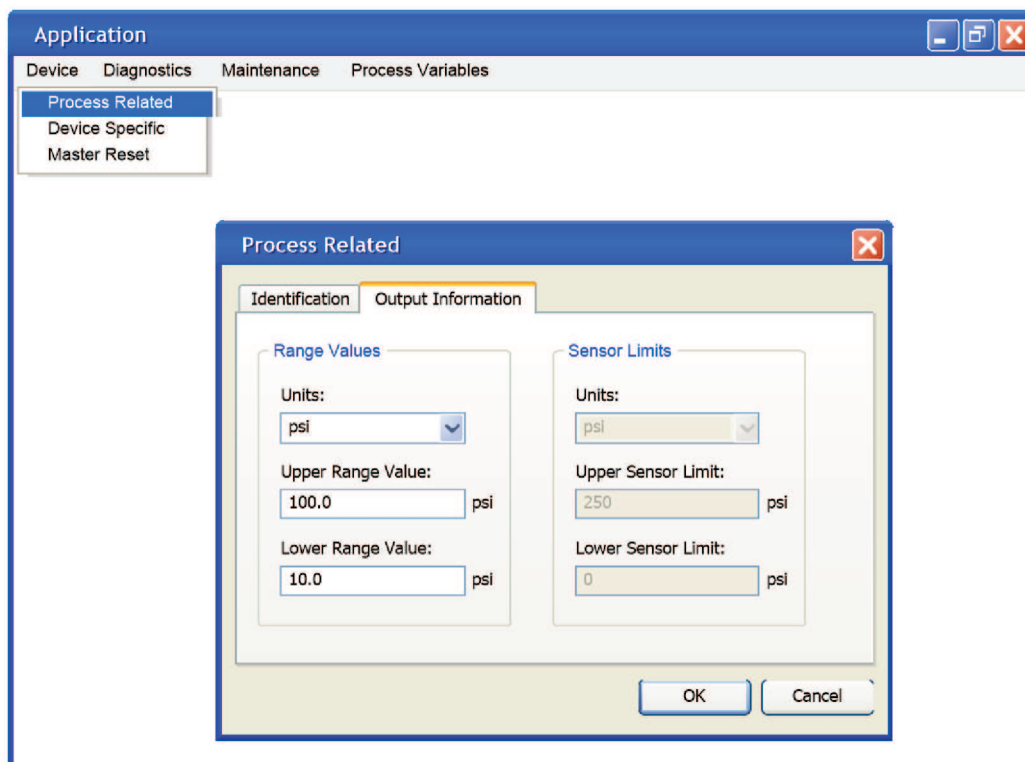


Figure 5 – Example of an EDD application for process-related device features

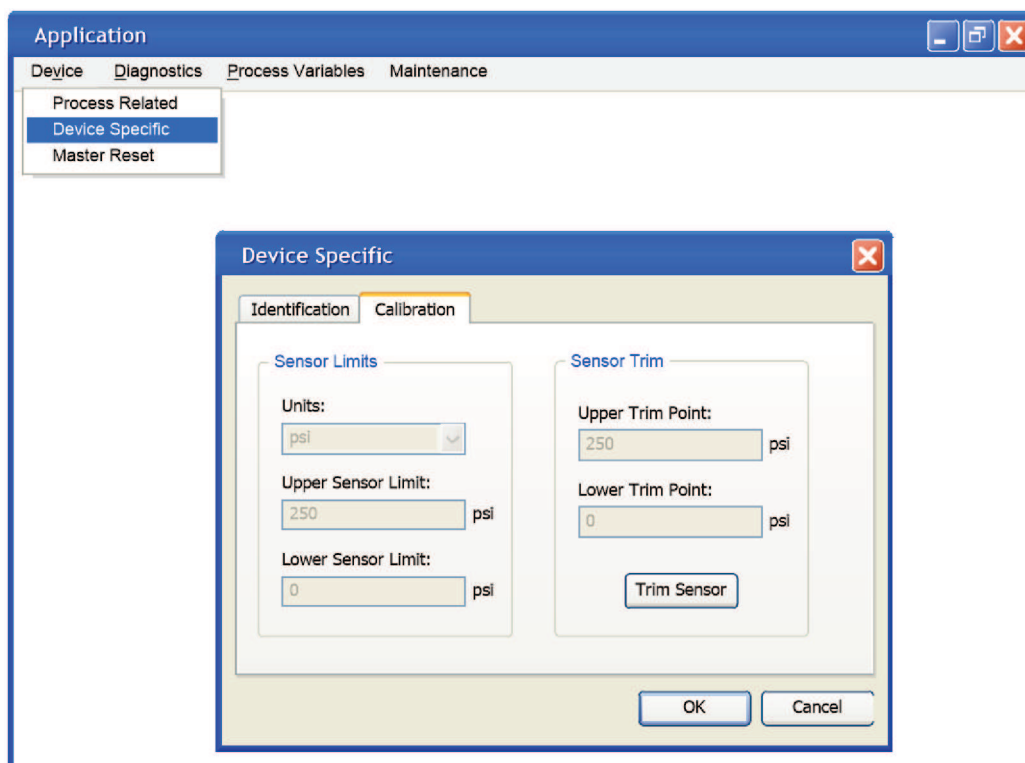


Figure 6 – Example of an EDD application for device features

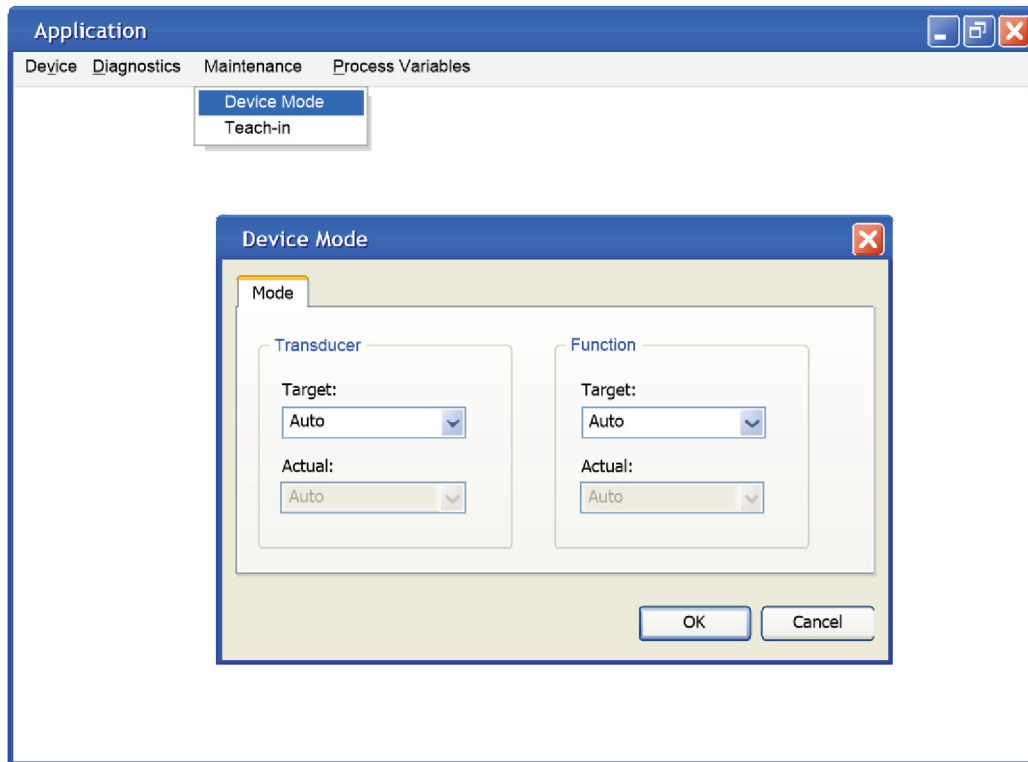


Figure 7 – Example of an EDD application for maintenance features

4.4 Label Concatenation for Indirect Variable References

4.4.1 General

For indirect variable-references only, the LABEL displayed shall be a concatenation of the strings found in the EDD item containers through which the VARIABLE was referenced. This occurs, for example, when a VARIABLE is referenced via a COLLECTION MEMBER or ARRAY ELEMENT. For example, PV.DEVICE_VARIABLE.DIGITAL_VALUE is an indirect reference to a VARIABLE via both arrays and collections. A reference is defined as:

```
reference := [ identifier | reference [ expression ] |
              reference . identifier ]
```

To specify Label concatenation, variable-references will be classified as a direct, simple, or complex reference.

A "direct reference" is:

```
direct-reference := identifier
```

String concatenation is not applicable to direct references.

A "simple reference" is:

```
simple-reference := [ identifier [ expression ] | identifier . identifier ]
```

And a "complex reference" is:

```
complex-reference := [ simple-reference . identifier |
                      simple-reference [ expression ] ]
```

676 **Table 5 – Label rule summary for simple variable references**

678 **Table 6 – Label rule summary for simple variable references**

693 If the final term is encountered with no Prefix-string being generated then the rules for simple
694 variable-references are used to generate the LABEL using the final term in the complex
695 variable-reference.

Table 7 – Prefix rule summary for complex variable references

Reference Type	Description exists	
	Yes	No
FILE REFERENCE_ARRAY COLLECTION RECORD BLOCK_A	Prefix is Description.	Skip this term. Continue to next term in the complex variable-reference to locate Prefix string.

Table 8 – Prefix rule summary for complex variable references

Reference Type	Label exists	
	Yes	No
LIST VALUE_ARRAY	Prefix shall be Label "[n]" where n is the value of the index.	Prefix shall be "[n]" where n is the value of the index.

Body-String

The Body-string is a sequential concatenation of strings generated from the intermediate terms in the complex variable-reference. See Table 9 and Table 10 for a summary of the rules for constructing the Body-string.

Generation of the Body-string ceases when the final term in the complex variable-reference is reached. The Body-string shall be the null (i.e., "") string if the final term is reached before a Body-string is created. The final term in the complex variable-reference shall be used to generate the Suffix.

Table 9 – Body rule summary for complex variable references

Reference Type	
FILE COLLECTION RECORD	Nothing is concatenated to the Body-string.
LIST VALUE_ARRAY	[n] is concatenated to the Body-string where n is the value of the index found in the reference.

Table 10 – Body rule summary for complex variable references

Reference Type	Prefix is from a FILE, COLLECTION or REFERENCE_ARRAY	
	Yes	No
REFERENCE_ARRAY	Nothing is concatenated to the Body-string.	[n] is concatenated to the Body-string where n is the index.

Suffix-String

The Suffix-string is generated from the final term in the complex variable-reference. See Table 11 and Table 12 for a summary of the rules for constructing the Suffix-string.

Table 11 – Suffix rule summary for complex variable references

Reference Type	
----------------	--

Reference Type	
LIST VALUE_ARRAY	[n] is Suffix where n is the value of the index.

Table 12 – Suffix rule summary for complex variable references

Reference Type	Description exists	
	Yes	No
FILE REFERENCE_ARRAY COLLECTION RECORD	Suffix is Description.	No Suffix (i.e., an empty string).

4.5 Help Concatenation

4.5.1 General

The Help-string for a simple or complex variable-reference is generated in a manner similar to that used for the Label-string and depends on whether it is a simple or complex variable-reference.

In the following clauses, any help that holds an empty string ("") shall be treated the same as if the help does not exist.

4.5.2 Simple variable references

For simple variable-references, the help string is generated using the rules in Table 13 and Table 14.

Table 13 – Help rule summary for simple variable references

Reference Type	Container's Help + Member/Element Help exists	Only Container's Help exists	Only Member/Element Help exists	Neither Help exists
FILE REFERENCE_ARRAY COLLECTION RECORD	Help shall be space concatenation of both Help-strings.	Help shall be Container's Help.	Help shall be Member/Element Help.	Help shall be same as with direct variable-reference.

Table 14 – Help rule summary for simple variable references

Reference Type	Container's Help exists	
	Yes	No
LIST VALUE_ARRAY	Help shall be Container's Help.	Help shall be an empty string.

4.5.3 Complex variable references

For a complex variable-reference, in general, there is possibly a Help-string Prefix and Suffix. Unlike Label concatenation, there is no Help-string Body. Basically, the Help string is the first-found Help space-concatenated to the Help from the final term in the complex variable-reference. If both the Help Prefix and Suffix are empty strings then the Help-string shall be an empty string.

The Help Prefix is generated based on the first term in the complex variable-reference. In the case of a FILE, REFERENCE_ARRAY, or COLLECTION reference, the first non-empty Member/Element Help is used. See the Prefix rules summarized in Table 15 and Table 16.

If the final term is encountered with no Prefix being generated then the rules for simple variable-references are used to generate the Help based on the final term in the complex variable-reference.

Table 15 – Help Prefix rule summary for complex variable references

Reference Type	
LIST VALUE_ARRAY	Prefix is Help string from the list or value-array (possibly empty).

Table 16 – Help Prefix rule summary for complex variable references

Reference Type	Member/Element Help exists	
	Yes	No
FILE REFERENCE_ARRAY COLLECTION RECORD	Help Prefix is Member/Element Help string.	Skip this reference. Continue to next reference to locate HELP Prefix string.

The Help suffix is generated from the final term in the complex variable-reference (see Table 17 and Table 18).

Table 17 – Help Suffix rule summary for complex variable references

Reference Type	
LIST VALUE_ARRAY	Suffix is Help string from referenced VARIABLE (possibly empty).

Table 18 – Help Suffix rule summary for complex variable references

Reference Type	Member/Element Help exists	
	Yes	No
FILE REFERENCE_ARRAY COLLECTION RECORD	Help Suffix is Member/Element Help string.	No Suffix (i.e., an empty string).

4.6 Containers and contained items

4.6.1 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.

4.6.2 Permitted and default STYLES

Table 19 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.

General rules:

- a) 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 19 – Permitted contained items and default STYLES

Container (Parent)	Permitted contained items (Child items)																Default STYLE of contained items (used if STYLE is not defined or in case of conflicts)	
	MENU	DIALOG, WINDOW	PAGE	GROUP	TABLE	EDIT_DISPLAY	VARIABLE	COLLECTION, RECORD	ARRAY, LIST	CHART, GRAPH, GRID	IMAGE	static text	METHOD	PLUGIN	COLUMNBREAK, ROWBREAK	SEPARATOR		
MENU	✓	1	✗	✗	10	✓	8	D	8	2	2	✓	✓	✓	6	6	Communication Profiles: CS, FF, GPE, ISA100, PB, PN MENU if no data item is in the menu, otherwise DIALOG. Communication Profiles: HART MENU	
DIALOG, WINDOW	✓	1	✓	✓	7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	5	PAGE	
PAGE	✓	1	✗	✓	7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	5	GROUP	
GROUP	✓	1	✗	✓	7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	5	GROUP if the parent of the parent GROUP is not of STYLE GROUP. DIALOG rendered as button or hyperlink if the parent of the parent GROUP is of STYLE GROUP.	
TABLE	✗	9	✗	✗	✓	✓	3	D	✓	2	3	✓	✓	✓	6	6	TABLE	

Key

✓ Shall be rendered as defined for the contained item; e.g., for a "MENU" according to its STYLE attribute.

✗ Not allowed. Default STYLE shall be used to solve this conflict (see column "Default STYLE of contained items")

D Default STYLE shall be used (see column "Default STYLE of contained items")

1 Shall be rendered as button or hyperlink by using the LABEL. The item shall be presented when the button or hyperlink is selected.

2 Shall be rendered as a button or hyperlink by using the LABEL. The item shall be presented on a dialog (i.e., a modal window), when the button or hyperlink is selected.

3 Shall be rendered as button or hyperlink by using the LABEL, or inline. If not rendered inline, then the item shall be presented on a dialog (i.e., a modal window), when the button or hyperlink is selected.

- 4 Shall be rendered as a button or hyperlink by using the LABEL. The item shall be presented on a modeless window, when the button or hyperlink is selected.
- 5 Shall be interpreted as COLUMNBREAK.
- 6 Shall be rendered as a line or ignored.
- 7 **Communication Profiles: CS, FF, GPE, ISA100, PB, PN**
 Shall be rendered according to note “✓”.
- Communication Profiles: HART**
 Shall be rendered according to note “X”.
- 8 **Communication Profiles: CS, FF, GPE, ISA100, PB, PN**
 Shall be rendered according to note “2”.
- Communication Profiles: HART**
 Shall be ignored and not presented.
- 9 **Communication Profiles: CS, FF, GPE, ISA100, PB, PN**
 Shall be rendered according to note “1”.
- Communication Profiles: HART**
 Shall be rendered according to note “X”.
- 10 **Communication Profiles: CS, FF, GPE, ISA100, PB, PN**
 Shall be rendered according to note “4”.
- Communication Profiles: HART**
 Shall be rendered according to note “D” if subordinate to DIALOG or WINDOW, otherwise according to note “4”.

777

778

779 4.6.3 Containers

780 4.6.3.1 Menu

781 Communication Profiles: CS, FF, GPE, ISA100, PB, PN

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

784 Communication Profiles: HART

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

786 4.6.3.2 Window

787 A menu of STYLE WINDOW takes the form of a modeless window. The LABEL of the menu
 788 shall be used as the caption for the window.

789 4.6.3.3 Dialog

790 A MENU of STYLE DIALOG shall be a modal window. If a dialog contains a subordinated
 791 window, the EDD application shall manage the window as a modal subdialog. The LABEL of
 792 the menu shall be used as the caption for the window.

793 4.6.3.4 Table

794 Communication Profiles: CS, FF, GPE, ISA100, PB, PN

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

798 Communication Profiles: HART

799 A menu of STYLE TABLE shall be rendered as a cascaded menu tree. Each item shall
800 be represented as one or multiple rows, e.g. ARRAYs. The submenus build a
801 hierarchy, which shall be shown. The label of the MENU-item (including
802 BIT_ENUMERATED VARIABLES) shall be displayed. Selecting MENU item shall cause
803 the contents to be displayed.

804 4.6.3.5 Page

805 A menu of STYLE PAGE within a WINDOW or DIALOG shall be rendered as a tab and spans
806 the entire canvas width. When multiple pages (tabs) are rendered in the same window or
807 dialog, each tab shall be individually selectable to enable the contents of a particular tab. The
808 LABEL of the menu shall be used as the caption for the tab. There shall be an implicit
809 ROWBREAK before and after the PAGE.

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

812 COLUMNBREAKs and SEPARATORs between pages shall be ignored.

813 4.6.3.6 Group

814 A menu of STYLE GROUP takes the form of a group box. The LABEL of the menu shall be
815 used as the caption for the group. In order to maintain a clear arrangement on the user
816 interface, the following restricting rules apply.

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

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

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

```

833 VARIABLE ail_strev
834 {
835     LABEL "Analog Input 1: Static Revision";
836     TYPE INTEGER(4);
837 }
838
839 VARIABLE ail_bloMo
840 {
841     LABEL "Analog Input 1: Block Mode";
842     TYPE INTEGER(1);
843 }
844
845 MENU ail_group_double /* Leads to double display of */
846 {
847     LABEL "Analog Input 1";
848     STYLE GROUP;
849     ITEMS
850     {
851         ail_strev,
852         ail_bloMo
853     }
854 }
855
856 COLLECTION ail_groupcol
857 {
858     MEMBERS
859     {
860         strev,ail_strev,"Static Revision";
861         bloMo,ail_bloMo,"Block Mode";
862     }
863 }
864
865 MENU ail_group_single /* Leads to single display of */
866 {
867     LABEL "Analog Input 1";
868     STYLE GROUP;
869     ITEMS
870     {
871         ail_groupcol.strev,
872         ail_groupcol.bloMo
873     }
874 }

```

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

4.6.4 Contained items

4.6.4.1 Overview

How a container renders contained items is described in 4.6.3.1 . 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. However, there shall be some methodology, such as a "tool tip", "pop- up window", "expanding window", etc, to display the entire string to the user.

4.6.4.2 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.

4.6.4.3 Variables

4.6.4.3.1 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.

- 896 • HANDLING = WRITE: the value should not be read from the device due to being
897 referenced as a contained item, e.g in a MENU.
- 898 • HANDLING = READ & WRITE: The value should be editable.
- 899 • CLASS = DYNAMIC: the value should be updated continuously with current read values
900 from the device. If the value has been edited, continuous updates shall no longer occur so
901 that the edited value is not overwritten with the value read from the device.
- 902 For string type VARIABLES the value should not be truncated or wrapped in multiple lines.
903 Word wrapping could be confused with a true "new line"characters ('\n') in the string value.
- 904 When no VARIABLE value is available to be displayed, e.g. due to a change in VALIDITY or
905 HANDLING, the EDD application shall indicate that the variable is in uninitialized state.
- 906 Table 20 shows how the EDD application should indicate the uninitialized state of
907 VARIABLES.

908 **Table 20 – Uninitialized state of VARIABLES on user interface**

VARIABLE TYPE	Uninitialized state indication
Numeric values ENUMERATED String values Date, time and duration values	Blank with indication of "uncertain" state close to the value field
BIT_ENUMERATED	Indication of no bit set along with indication of "uncertain" state close to the value field
BOOLEAN	Indication of FALSE along with indication of "uncertain" state close to the value field

- 909
- 910 **4.6.4.3.2 Variable of TYPE BIT_ENUMERATED**
- 911 Multiple bits can be packaged in a single-bit enumerated variable. Each bit could have a
912 distinct meaning. It is possible to display each bit separately and, in addition, to display the
913 whole BIT_ENUMERATED variable. BIT_ENUMERATED variables may be displayed as
914 checkboxes.
- 915 In case of showing a bit of a BIT_ENUMERATED variable, the variable reference should be
916 extended with a mask in square brackets and the LABEL of the variable is not shown.
- 917 When the whole BIT_ENUMERATED VARIABLE is displayed, the display of the bits in the
918 VARIABLE shall appear grouped together.
- 919 When a bit enumeration is of CLASS DIAGNOSTIC then the bits displayed should be treated
920 as a status indicator. A red indication should be presented when the bit is set and green when
921 the bit is reset.
- 922 Communication Profiles: HART
- 923 When the whole BIT_ENUMERATED VARIABLE is displayed, the display of the bits in
924 the VARIABLE shall be displayed inside a group-box. The LABEL of the
925 BIT_ENUMERATED VARIABLE shall be used as the label of the group box. EDD
926 applications with limited display constraints that prevent the display of group boxes are
927 not required to meet this requirement.
- 928 Figure 9 shows an EDD example for displaying single bits of a BIT_ENUMERATED variable.

```

929 VARIABLE var1
930 {
931     LABEL "Var 1";
932     TYPE BIT_ENUMERATED
933     {
934         { 0x1, "Bit 0"},
935         { 0x2, "Bit 1"},
936         { 0x4, "Bit 2"}
937     }
938 }
939
940 MENU diagnostic_info
941 {
942     LABEL "Diagnostic";
943     ITEMS
944     {
945         var1[0x1],    // Bit 0
946         var1[0x2],    // Bit 1
947         var1[0x4]     // Bit 2
948     }
949 }

```

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.

```

953 VARIABLE var1
954 {
955     LABEL "Var 1";
956     TYPE BIT_ENUMERATED
957     {
958         { 0x1, "Bit 0"},
959         { 0x2, "Bit 1"},
960         { 0x4, "Bit 2"}
961     }
962 }
963
964 MENU var1_group
965 {
966     LABEL var1.LABEL;
967     STYLE GROUP;
968     ITEMS
969     {
970         var1[0x1],    // Bit 0
971         var1[0x2],    // Bit 1
972         var1[0x4]     // Bit 2
973     }
974 }
975
976 MENU diagnostic_info
977 {
978     LABEL "Diagnostic";
979     STYLE PAGE;
980     ITEMS
981     {
982         var1,          // all bit should be shown
983         COLUMNBREAK,
984         var1[0x1],     // Bit 0
985         var1[0x2],     // Bit 1
986         var1[0x4],     // Bit 2
987         COLUMNBREAK,
988         var1_group
989     }
990 }

```

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.

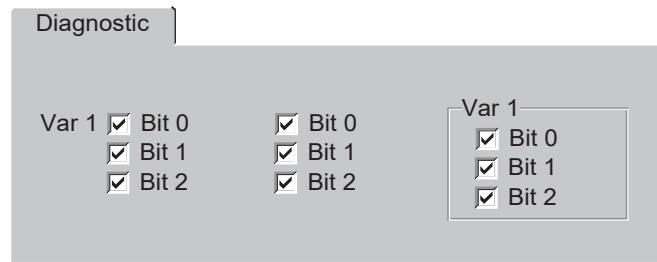


Figure 11 – Example of an EDD application for a variable of type BIT_ENUMERATED

4.6.4.3.3 Variable of TYPE INDEX

When a VARIABLE of type INDEX is presented to the user, the EDD application shall determine the value 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 referenced ELEMENTS exist then this shall be displayed else;
3. the LABEL of the ARRAY or LIST 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.

4.6.4.3.4 Variable with HANDLING WRITE

According to the HANDLING attribute a VARIABLE with HANDLING WRITE is a "write-only" variable. This means that the variable is editable and may be written to the device, but it should not be read from the device.

For online session, EDD application shall display the VARIABLE value as uninitialized (see Table 20), until the VARIABLE is modified during the same edit session.

For offline session, EDD application shall display the VARIABLE value from the current cache.

Figure 12 shows an EDD example with a "write-only" variable.