

DIN EN IEC 61804-5

ICS 35.060; 35.240.50

Einsprüche bis 2021-04-05
Vorgesehen als Ersatz für
DIN EN 61804-5:2016-04**Entwurf**

**Geräte und Geräteintegration in Unternehmenssysteme –
Funktionsbausteine für die Prozessautomation und elektronische
Gerätebeschreibungssprache –
Teil 5: Bibliothek vorgefertigter Unterprogramme
(IEC 61804-5:2020);
Englische Fassung EN IEC 61804-5:2020**

Devices and intergration in enterprise systems –
Function blocks (FB) for process control and electronic device description
language (EDDL) –
Part 5: EDDL Built-in library
(IEC 61804-5:2020);
English version EN IEC 61804-5:2020

Les dispositifs et leur intégration dans les systèmes de l'entreprise –
Blocs fonctionnels (FB) pour les procédés industriels et le langage de description
électronique de produit (EDDL) –
Partie 5: Bibliothèque de Built-in EDDL
(IEC 61804-5:2020);
Version anglaise EN IEC 61804-5:2020

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2021-02-05 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 282 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE



Nationales Vorwort

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.

Die enthaltene IEC-Publikation wurde vom IEC/SC 65E „Devices and integration in enterprise systems“ erarbeitet und unterliegt dem Copyright der IEC .

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

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 den vorliegenden Normentwurf erfüllt.

Da sich die Benutzer des vorliegenden Norm-Entwurfs der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN IEC 61804-5:2020 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und Abschnitt 3 (außer Unterabschnitte 3.2), 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.

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 61804-5:2016-04 wurden folgende Änderungen vorgenommen:

- a) Kommunikationsprofile ISA100 und GPE wurden hinzugefügt;
- b) die folgende Builtins wurden abgelehnt:
- c) ABORT_ON_NO_DEVICE
- d) IGNORE_NO_DEVICE
- e) RETRY_ON_NO_DEVICE
- f) XMTR_ABORT_ON_NO_DEVICE
- g) XMTR_IGNORE_NO_DEVICE
- h) XMTR_RETRY_ON_NO_DEVICE
- i) get_status_code_string.

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 der IEC 61804

- enthält eine Liste von Builtins, die innerhalb von EDDL-Methoden verwendet werden können;
- definiert die Funktionalität und Parameter für jeden Builtin;
- beschreibt, wie EDDL-Variablen innerhalb von EDDL-Methoden referenziert werden können;

Die EDDL-Funktionen sind für jede der Kommunikationstechnologien durch das Profil begrenzt. Die Beschreibungen in diesem Teil der IEC 61804 beziehen sich auf diese Merkmale im Allgemeinen und nicht alle Kommunikationstechnologien unterstützen alle beschriebenen Merkmale. Die Profildefinitionen in der Übersicht dieses Teils und in der IEC 61804-3 sind heranzuziehen, um die von jeder Kommunikationstechnologie unterstützten Merkmale zu verstehen.

1 Anwendungsbereich

Dieser Teil der IEC 61804 beschreibt die EDDL Builtin Bibliothek und stellt die Profile der verschiedenen Feldbusse zur Verfügung.

3 Begriffe, Symbole, Abkürzungen und Vereinbarungen

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe, Symbole, Abkürzungen und Vereinbarungen.

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

Methoden-Variable

(en:method-variable)

Variable mit Umfang innerhalb einer Methode

– Leerseite –

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN IEC 61804-5

August 2020

ICS 25.040.40; 35.240.50

Supersedes EN 61804-5:2015 and all of its amendments
and corrigenda (if any)

English Version

Devices and intergration in enterprise systems - Function blocks
(FB) for process control and electronic device description
language (EDDL) - Part 5: EDDL Built-in library
(IEC 61804-5:2020)

Les dispositifs et leur intégration dans les systèmes de
l'entreprise - Blocs fonctionnels (FB) pour les procédés
industriels et le langage de description électronique de
produit (EDDL) - Partie 5: Bibliothèque de Built-in EDDL
(IEC 61804-5:2020)

Funktionsbausteine für die Prozessautomation und
elektronische Gerätebeschreibungssprache -
Teil 5: Bibliothek vorgefertigter Unterprogramme
(IEC 61804-5:2020)

This European Standard was approved by CENELEC on 2020-07-29. 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, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, 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: Rue de la Science 23, B-1040 Brussels

European foreword

The text of document 65E/634/CDV, future edition 2 of IEC 61804-5, 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 IEC 61804-5:2020.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by (dop) 2021-04-29
publication of an identical national standard or by endorsement
- latest date by which the national standards conflicting with the (dow) 2023-07-29
document have to be withdrawn

This document supersedes EN 61804-5:2015 and all of its amendments and corrigenda (if any).

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

Endorsement notice

The text of the International Standard IEC 61804-5:2020 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 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.

NOTE 1 Where 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 61804-3	2020	Devices and integration in enterprise systems - Function blocks (FB) for process control and electronic device description language (EDDL) - Part 3: EDDL syntax and semantics	EN IEC 61804-3	2020
ISO/IEC 9899	-	Information technology - Programming languages - C	-	-
IEEE 754	-	IEEE Standard for Floating-Point Arithmetic	-	-

CONTENTS

European foreword	2
Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	3
INTRODUCTION	22
1 Scope.....	23
2 Normative references	23
3 Terms, definitions, acronyms and abbreviated terms.....	23
3.1 Terms and definitions	23
3.2 Acronyms and abbreviated terms	23
4 Conventions for builtin descriptions.....	24
5 Builtin categories	25
5.1 Overview	25
5.2 User interface builtins	35
5.3 Communication builtins.....	36
5.4 Action builtins.....	38
6 EDDL builtin library.....	41
6.1 General	41
6.2 User interface builtins	42
6.2.1 Builtin ACKNOWLEDGE	42
6.2.2 Builtin acknowledge	42
6.2.3 Builtin DELAY	43
6.2.4 Builtin delay	43
6.2.5 Builtin delayfor	44
6.2.6 Builtin delayfor2	45
6.2.7 Builtin DISPLAY.....	47
6.2.8 Builtin display.....	48
6.2.9 Builtin display_bitenum.....	48
6.2.10 Builtin display_builtin_error.....	49
6.2.11 Builtin display_comm_error	49
6.2.12 Builtin display_comm_status	50
6.2.13 Builtin display_device_status.....	51
6.2.14 Builtin display_dynamics	51
6.2.15 Builtin display_dynamics2	52
6.2.16 Builtin display_message.....	53
6.2.17 Builtin display_message2	54
6.2.18 Builtin display_response_code	56
6.2.19 Builtin display_response_status	57
6.2.20 Builtin display_xmtr_status	57
6.2.21 Builtin edit_device_value	58
6.2.22 Builtin edit_device_value2	59
6.2.23 Builtin edit_local_value	61
6.2.24 Builtin edit_local_value2	62
6.2.25 Builtin get_acknowledgement.....	63
6.2.26 Builtin get_acknowledgement2.....	64

6.2.27	Builtin GET_DEV_VAR_VALUE	65
6.2.28	Builtin get_dev_var_value	66
6.2.29	Builtin GET_LOCAL_VAR_VALUE	67
6.2.30	Builtin get_local_var_value	67
6.2.31	Builtin Menu	68
6.2.32	Builtin MenuDisplay (version A)	69
6.2.33	Builtin MenuDisplay (version B)	70
6.2.34	Builtin PUT_MESSAGE	71
6.2.35	Builtin put_message	72
6.2.36	Builtin SELECT_FROM_LIST	74
6.2.37	Builtin select_from_list	75
6.2.38	Builtin select_from_menu	75
6.2.39	Builtin select_from_menu2	77
6.2.40	Prompt string formats	78
6.3	Communication builtins	79
6.3.1	Builtin abort_on_all_comm_errors	79
6.3.2	Builtin ABORT_ON_ALL_COMM_STATUS	80
6.3.3	Builtin ABORT_ON_ALL_DEVICE_STATUS	80
6.3.4	Builtin ABORT_ON_ALL_RESPONSE_CODES	81
6.3.5	Builtin abort_on_all_response_codes	81
6.3.6	Builtin ABORT_ON_COMM_ERROR	82
6.3.7	Builtin abort_on_comm_error	82
6.3.8	Builtin ABORT_ON_COMM_STATUS	83
6.3.9	Builtin ABORT_ON_DEVICE_STATUS	83
6.3.10	Builtin ABORT_ON_NO_DEVICE (deprecated)	84
6.3.11	Builtin ABORT_ON_RESPONSE_CODE	85
6.3.12	Builtin abort_on_response_code	86
6.3.13	Builtin ext_send_command	86
6.3.14	Builtin ext_send_command_trans	87
6.3.15	Builtin fail_on_all_comm_errors	88
6.3.16	Builtin fail_on_all_response_codes	89
6.3.17	Builtin fail_on_comm_error	89
6.3.18	Builtin fail_on_response_code	90
6.3.19	Builtin get_more_status	90
6.3.20	Builtin IGNORE_ALL_COMM_STATUS	91
6.3.21	Builtin IGNORE_ALL_DEVICE_STATUS	92
6.3.22	Builtin IGNORE_ALL_RESPONSE_CODES	92
6.3.23	Builtin IGNORE_COMM_ERROR	93
6.3.24	Builtin IGNORE_COMM_STATUS	93
6.3.25	Builtin IGNORE_DEVICE_STATUS	94
6.3.26	Builtin IGNORE_NO_DEVICE (deprecated)	94
6.3.27	Builtin IGNORE_RESPONSE_CODE	95
6.3.28	Builtin read_value	95
6.3.29	Builtin read_value2	96
6.3.30	Builtin ReadCommand	97
6.3.31	Builtin retry_on_all_comm_errors	97
6.3.32	Builtin RETRY_ON_ALL_COMM_STATUS	98

6.3.33	Builtin RETRY_ON_ALL_DEVICE_STATUS	98
6.3.34	Builtin RETRY_ON_ALL_RESPONSE_CODES	99
6.3.35	Builtin retry_on_all_response_codes	100
6.3.36	Builtin RETRY_ON_COMM_ERROR	100
6.3.37	Builtin retry_on_comm_error	101
6.3.38	Builtin RETRY_ON_COMM_STATUS	101
6.3.39	Builtin RETRY_ON_DEVICE_STATUS	102
6.3.40	Builtin RETRY_ON_NO_DEVICE (deprecated)	102
6.3.41	Builtin RETRY_ON_RESPONSE_CODE	103
6.3.42	Builtin retry_on_response_code	104
6.3.43	Builtin send	104
6.3.44	Builtin send_all_values	105
6.3.45	Builtin send_command	106
6.3.46	Builtin send_command_trans	106
6.3.47	Builtin send_trans	107
6.3.48	Builtin send_value	108
6.3.49	Builtin send_value2	108
6.3.50	Builtin SET_NUMBER_OF_RETRIES	109
6.3.51	Builtin WriteCommand	110
6.3.52	Builtin XMTR_ABORT_ON_ALL_COMM_STATUS	110
6.3.53	Builtin XMTR_ABORT_ON_ALL_DATA	111
6.3.54	Builtin XMTR_ABORT_ON_ALL_DEVICE_STATUS	111
6.3.55	Builtin XMTR_ABORT_ON_ALL_RESPONSE_CODES	112
6.3.56	Builtin XMTR_ABORT_ON_COMM_ERROR	112
6.3.57	Builtin XMTR_ABORT_ON_COMM_STATUS	113
6.3.58	Builtin XMTR_ABORT_ON_DATA	113
6.3.59	Builtin XMTR_ABORT_ON_DEVICE_STATUS	114
6.3.60	Builtin XMTR_ABORT_ON_NO_DEVICE (deprecated)	114
6.3.61	Builtin XMTR_ABORT_ON_RESPONSE_CODE	115
6.3.62	Builtin XMTR_IGNORE_ALL_COMM_STATUS	115
6.3.63	Builtin XMTR_IGNORE_ALL_DATA	116
6.3.64	Builtin XMTR_IGNORE_ALL_DEVICE_STATUS	116
6.3.65	Builtin XMTR_IGNORE_ALL_RESPONSE_CODES	117
6.3.66	Builtin XMTR_IGNORE_COMM_ERROR	117
6.3.67	Builtin XMTR_IGNORE_COMM_STATUS	118
6.3.68	Builtin XMTR_IGNORE_DATA	118
6.3.69	Builtin XMTR_IGNORE_DEVICE_STATUS	119
6.3.70	Builtin XMTR_IGNORE_NO_DEVICE (deprecated)	119
6.3.71	Builtin XMTR_IGNORE_RESPONSE_CODE	120
6.3.72	Builtin XMTR_RETRY_ON_ALL_COMM_STATUS	121
6.3.73	Builtin XMTR_RETRY_ON_ALL_DATA	121
6.3.74	Builtin XMTR_RETRY_ON_ALL_DEVICE_STATUS	122
6.3.75	Builtin XMTR_RETRY_ON_ALL_RESPONSE_CODES	122
6.3.76	Builtin XMTR_RETRY_ON_COMM_ERROR	123
6.3.77	Builtin XMTR_RETRY_ON_COMM_STATUS	123
6.3.78	Builtin XMTR_RETRY_ON_DATA	124
6.3.79	Builtin XMTR_RETRY_ON_DEVICE_STATUS	124

6.3.80	Builtin XMTR_RETRY_ON_NO_DEVICE (deprecated)	125
6.3.81	Builtin XMTR_RETRY_ON_RESPONSE_CODE	125
6.4	Action builtins	126
6.4.1	Builtin fgetval	126
6.4.2	Builtin fsetval	126
6.4.3	Builtin get_date	127
6.4.4	Builtin get_double	127
6.4.5	Builtin get_float	128
6.4.6	Builtin get_signed	129
6.4.7	Builtin get_string	129
6.4.8	Builtin get_unsigned	130
6.4.9	Builtin igetval	131
6.4.10	Builtin isetval	131
6.4.11	Builtin isOffline	132
6.4.12	Builtin lgetval	132
6.4.13	Builtin lsetval	133
6.4.14	Builtin put_date	133
6.4.15	Builtin put_double	134
6.4.16	Builtin put_float	134
6.4.17	Builtin put_signed	135
6.4.18	Builtin put_string	136
6.4.19	Builtin put_unsigned	136
6.4.20	Builtin sgetval	137
6.4.21	Builtin ssetval	138
6.5	Variable access builtins	138
6.5.1	Builtin assign	138
6.5.2	Builtin assign_double	139
6.5.3	Builtin assign_float	139
6.5.4	Builtin assign_int	140
6.5.5	Builtin assign_var	140
6.5.6	Builtin assign2	141
6.5.7	Builtin dassign	142
6.5.8	Builtin fassign	143
6.5.9	Builtin float_value	143
6.5.10	Builtin fvar_value	144
6.5.11	Builtin get_date_value	144
6.5.12	Builtin get_date_value2	145
6.5.13	Builtin get_double_value	146
6.5.14	Builtin get_double_value2	147
6.5.15	Builtin get_float_value	147
6.5.16	Builtin get_float_value2	148
6.5.17	Builtin get_signed_value	149
6.5.18	Builtin get_signed_value2	150
6.5.19	Builtin get_string_value	151
6.5.20	Builtin get_string_value2	152
6.5.21	Builtin get_unsigned_value	153
6.5.22	Builtin get_unsigned_value2	154

6.5.23	Builtin iassign.....	155
6.5.24	Builtin int_value.....	155
6.5.25	Builtin ivar_value.....	156
6.5.26	Builtin lassign.....	156
6.5.27	Builtin long_value.....	156
6.5.28	Builtin lvar_value.....	157
6.5.29	Builtin put_date_value	157
6.5.30	Builtin put_date_value2	158
6.5.31	Builtin put_double_value	159
6.5.32	Builtin put_double_value2	160
6.5.33	Builtin put_float_value	161
6.5.34	Builtin put_float_value2	162
6.5.35	Builtin put_signed_value.....	163
6.5.36	Builtin put_signed_value2.....	163
6.5.37	Builtin put_string_value	164
6.5.38	Builtin put_string_value2	165
6.5.39	Builtin put_unsigned_value.....	166
6.5.40	Builtin put_unsigned_value2.....	167
6.5.41	Builtin ret_double_value	168
6.5.42	Builtin ret_double_value2	169
6.5.43	Builtin ret_float_value	169
6.5.44	Builtin ret_float_value2	170
6.5.45	Builtin ret_signed_value	170
6.5.46	Builtin ret_signed_value2	171
6.5.47	Builtin ret_unsigned_value	171
6.5.48	Builtin ret_unsigned_value2	172
6.5.49	Builtin vassign.....	173
6.6	Normal termination builtins	173
6.6.1	Builtin discard_on_exit.....	173
6.6.2	Builtin save_on_exit.....	174
6.6.3	Builtin save_values.....	175
6.6.4	Builtin send_on_exit.....	175
6.7	Abnormal termination builtins	176
6.7.1	Builtin abort.....	176
6.7.2	Builtin add_abort_method (version A)	177
6.7.3	Builtin add_abort_method (version B)	177
6.7.4	Builtin method_abort.....	178
6.7.5	Builtin pop_abort_method.....	179
6.7.6	Builtin process_abort	179
6.7.7	Builtin push_abort_method	179
6.7.8	Builtin remove_abort_method (version A)	180
6.7.9	Builtin remove_abort_method (version B)	180
6.7.10	Builtin remove_all_abort_methods	181
6.8	String builtins	181
6.8.1	Builtin atof	181
6.8.2	Builtin atoi	182
6.8.3	Builtin BUILD_MESSAGE	182

6.8.4	Builtin dictionary_string	183
6.8.5	Builtin ftoa	183
6.8.6	Builtin get_comm_error_string	184
6.8.7	Builtin get_dictionary_string	185
6.8.8	Builtin get_enum_string	185
6.8.9	Builtin get_response_code_string	186
6.8.10	Builtin get_rspcode_string	187
6.8.11	Builtin get_rspcode_string_by_id	187
6.8.12	Builtin get_status_code_string (deprecated)	188
6.8.13	Builtin get_status_string	188
6.8.14	Builtin get_stddict_string	189
6.8.15	Builtin get_variable_string	190
6.8.16	Builtin itoa (version A)	190
6.8.17	Builtin itoa (version B)	191
6.8.18	Builtin strleft	191
6.8.19	Builtin strcmp	192
6.8.20	Builtin strlen	192
6.8.21	Builtin strlwr	193
6.8.22	Builtin strmid	193
6.8.23	Builtin strright	194
6.8.24	Builtin strstr	194
6.8.25	Builtin strtrim	195
6.8.26	Builtin strupr	196
6.8.27	Builtin trunc	196
6.9	Date time builtins	197
6.9.1	Builtin AddTime	197
6.9.2	Builtin DATE_AND_TIME_VALUE_to_string	197
6.9.3	Builtin Date_to_DayOfMonth	198
6.9.4	Builtin DATE_to_days	198
6.9.5	Builtin Date_to_Month	199
6.9.6	Builtin DATE_to_string	199
6.9.7	Builtin Date_To_Time	200
6.9.8	Builtin Date_to_Year	200
6.9.9	Builtin days_to_DATE	201
6.9.10	Builtin DELAY_TIME	201
6.9.11	Builtin DiffTime	201
6.9.12	Builtin From_DATE_AND_TIME_VALUE	202
6.9.13	Builtin From_TIME_VALUE	202
6.9.14	Builtin GetCurrentDate	203
6.9.15	Builtin GetCurrentDateAndTime	203
6.9.16	Builtin GetCurrentTime	204
6.9.17	Builtin GET_TICK_COUNT	204
6.9.18	Builtin Make_Time	205
6.9.19	Builtin seconds_to_TIME_VALUE	205
6.9.20	Builtin seconds_to_TIME_VALUE8	206
6.9.21	Builtin Time_To_Date	206
6.9.22	Builtin TIME_VALUE_to_Hour	206

6.9.23	Builtin TIME_VALUE_to_Minute.....	207
6.9.24	Builtin TIME_VALUE_to_Second	207
6.9.25	Builtin TIME_VALUE_to_seconds	208
6.9.26	Builtin TIME_VALUE_to_string.....	209
6.9.27	Builtin timet_to_string	209
6.9.28	Builtin timet_to_TIME_VALUE.....	210
6.9.29	Builtin timet_To_TIME_VALUE	210
6.9.30	Builtin timet_to_TIME_VALUE8.....	211
6.9.31	Builtin To_Date	211
6.9.32	Builtin To_Date_and_Time	212
6.9.33	Builtin To_Time.....	212
6.9.34	Builtin To_TIME_VALUE	213
6.9.35	Builtin To_TIME_VALUE8	213
6.10	Diagnostic builtins	214
6.10.1	Builtin get_comm_error	214
6.10.2	Builtin get_dds_error	214
6.10.3	Builtin get_response_code	215
6.11	Math builtins.....	216
6.11.1	Builtin abs	216
6.11.2	Builtin acos	217
6.11.3	Builtin asin	217
6.11.4	Builtin atan	218
6.11.5	Builtin ByteToDouble	218
6.11.6	Builtin ByteToFloat.....	219
6.11.7	Builtin ByteToLong.....	219
6.11.8	Builtin ByteToShort.....	219
6.11.9	Builtin cbrt.....	220
6.11.10	Builtin ceil.....	220
6.11.11	Builtin cos	221
6.11.12	Builtin cosh	221
6.11.13	Builtin DoubleToByte	222
6.11.14	Builtin drand.....	222
6.11.15	Builtin dseed	223
6.11.16	Builtin exp	223
6.11.17	Builtin FloatToByte.....	224
6.11.18	Builtin floor	224
6.11.19	Builtin fmod	225
6.11.20	Builtin fpclassify	225
6.11.21	Builtin is_NaN	226
6.11.22	Builtin log	226
6.11.23	Builtin log10	227
6.11.24	Builtin log2	227
6.11.25	Builtin LongToByte.....	227
6.11.26	Builtin nan	228
6.11.27	Builtin NaN_value	229
6.11.28	Builtin nanf	229
6.11.29	Builtin pow	230

6.11.30	Builtin round	230
6.11.31	Builtin ShortToByte.....	231
6.11.32	Builtin sin	231
6.11.33	Builtin sinh	232
6.11.34	Builtin sqrt.....	232
6.11.35	Builtin tan.....	232
6.11.36	Builtin tanh.....	233
6.12	List builtins	233
6.12.1	Builtin get_date_lelem	233
6.12.2	Builtin get_date_lelem2	234
6.12.3	Builtin get_double_lelem	235
6.12.4	Builtin get_double_lelem2	236
6.12.5	Builtin get_float_lelem	237
6.12.6	Builtin get_float_lelem2	238
6.12.7	Builtin get_signed_lelem	239
6.12.8	Builtin get_signed_lelem2	239
6.12.9	Builtin get_string_lelem	240
6.12.10	Builtin get_string_lelem2	241
6.12.11	Builtin get_unsigned_lelem	242
6.12.12	Builtin get_unsigned_lelem2	243
6.12.13	Builtin ListDeleteElementAt (version A)	244
6.12.14	Builtin ListDeleteElementAt (version B)	245
6.12.15	Builtin ListDeleteElementAt2	245
6.12.16	Builtin ListInsert (version A).....	246
6.12.17	Builtin ListInsert (version B).....	247
6.12.18	Builtin ListInsert2	248
6.13	Translation builtins.....	248
6.13.1	Builtin DICT_ID.....	248
6.13.2	Builtin get_block_instance_by_object_index.....	249
6.13.3	Builtin get_block_instance_by_tag.....	250
6.13.4	Builtin get_block_instance_count.....	251
6.13.5	Builtin get_resolve_status	251
6.13.6	Builtin ITEM_ID	252
6.13.7	Builtin MEMBER_ID	252
6.13.8	Builtin ObjectReference.....	253
6.13.9	Builtin resolve_array_ref.....	254
6.13.10	Builtin resolve_array_ref2.....	255
6.13.11	Builtin resolve_block_ref	255
6.13.12	Builtin resolve_block_ref2	256
6.13.13	Builtin resolve_list_ref	257
6.13.14	Builtin resolve_local_ref	257
6.13.15	Builtin resolve_local_ref2	258
6.13.16	Builtin resolve_param_list_ref	259
6.13.17	Builtin resolve_param_ref.....	259
6.13.18	Builtin resolve_param_ref2.....	260
6.13.19	Builtin resolve_record_ref.....	261
6.13.20	Builtin resolve_record_ref2.....	261

6.13.21	Builtin VARID	262
6.14	Block transfer builtins	262
6.14.1	Builtin abortTransferPort	262
6.14.2	Builtin closeTransferPort	263
6.14.3	Builtin fGetByte	263
6.14.4	Builtin get_transfer_status	264
6.14.5	Builtin openTransferPort	265
6.14.6	Builtin readItemFromDevice	265
6.14.7	Builtin writeItemToDevice	266
6.15	File builtins	266
6.15.1	Builtin browsedidentity	266
6.15.2	Builtin re_read_file	267
6.15.3	Builtin re_write_file	267
6.16	Identification builtins	268
6.16.1	Builtin GET_DD_REVISION	268
6.16.2	Builtin GET_DEVICE_REVISION	268
6.16.3	Builtin GET_DEVICE_TYPE	268
6.16.4	Builtin GET_MANUFACTURER	269
6.17	Development support builtins	269
6.17.1	Builtin _ERROR	269
6.17.2	Builtin _TRACE	270
6.17.3	Builtin _WARNING	270
6.17.4	Builtin LOG_MESSAGE	271
6.18	DDL references	271
6.18.1	Direct referencing of VARIABLES	271
6.18.2	Accessing standard dictionary strings	272
6.18.3	String support using DD_STRING	273
6.18.4	Referencing DD item attributes	273
6.19	Method functions	274
7	Builtins return codes	275
Figure 1 – A sample METHOD		275
Figure 2 – DD_STRING parameters passed by reference		275

Table 1	Format for the builtin overview table	24
Table 2	Contents of the builtin overview table	24
Table 3	Format for the builtins lexical element tables	24
Table 4	Contents of the lexical element table	25
Table 5	Builtin overview	25
Table 6	Usage of builtins	35
Table 7	User interface builtins	35
Table 8	Communication builtins	36
Table 9	Action builtins	39
Table 10	Builtin ACKNOWLEDGE	42
Table 11	Builtin acknowledge	43

Table 12 – Builtin DELAY	43
Table 13 – Builtin delay	44
Table 14 – Builtin delayfor	44
Table 15 – Builtin delayfor2	46
Table 16 – Builtin DISPLAY	47
Table 17 – Builtin display	48
Table 18 – Builtin display_bitenum	48
Table 19 – Builtin display_builtin_error	49
Table 20 – Builtin display_comm_error	50
Table 21 – Builtin display_comm_status	50
Table 22 – Builtin display_device_status	51
Table 23 – Builtin display_dynamics	52
Table 24 – Builtin display_dynamics2	53
Table 25 – Builtin display_message	54
Table 26 – Builtin display_message2	55
Table 27 – Builtin display_response_code	56
Table 28 – Builtin display_response_status	57
Table 29 – Builtin display_xmtr_status	58
Table 30 – Builtin edit_device_value	58
Table 31 – Builtin edit_device_value2	60
Table 32 – Builtin edit_local_value	61
Table 33 – Builtin edit_local_value2	63
Table 34 – Builtin get_acknowledgement	64
Table 35 – Builtin get_acknowledgement2	65
Table 36 – Builtin GET_DEV_VAR_VALUE	66
Table 37 – Builtin get_dev_var_value	67
Table 38 – Builtin GET_LOCAL_VAR_VALUE	67
Table 39 – Builtin get_local_var_value	68
Table 40 – Builtin Menu	69
Table 41 – Builtin MenuDisplay	70
Table 42 – Builtin MenuDisplay	71
Table 43 – Builtin PUT_MESSAGE	72
Table 44 – Builtin put_message	73
Table 45 – Builtin SELECT_FROM_LIST	74
Table 46 – Builtin select_from_list	75
Table 47 – Builtin select_from_menu	76
Table 48 – Builtin select_from_menu2	78
Table 49 – Format options	79
Table 50 – Builtin abort_on_all_comm_errors	79
Table 51 – Builtin ABORT_ON_ALL_COMM_STATUS	80
Table 52 – Builtin ABORT_ON_ALL_DEVICE_STATUS	81
Table 53 – Builtin ABORT_ON_ALL_RESPONSE_CODES	81

Table 54 – Builtin abort_on_all_response_codes.....	82
Table 55 – Builtin ABORT_ON_COMM_ERROR.....	82
Table 56 – Builtin abort_on_comm_error	83
Table 57 – Builtin ABORT_ON_COMM_STATUS	83
Table 58 – Builtin ABORT_ON_DEVICE_STATUS	84
Table 59 – Builtin ABORT_ON_NO_DEVICE	84
Table 60 – Available abort and retry masks	85
Table 61 – Builtin ABORT_ON_RESPONSE_CODE.....	86
Table 62 – Builtin abort_on_response_code	86
Table 63 – Builtin ext_send_command	87
Table 64 – Builtin ext_send_command_trans	88
Table 65 – Builtin fail_on_all_comm_errors	88
Table 66 – Builtin fail_on_all_response_codes	89
Table 67 – Builtin fail_on_comm_error.....	90
Table 68 – Builtin fail_on_response_code.....	90
Table 69 – Builtin get_more_status	91
Table 70 – Builtin IGNORE_ALL_COMM_STATUS.....	91
Table 71 – Builtin IGNORE_ALL_DEVICE_STATUS.....	92
Table 72 – Builtin IGNORE_ALL_RESPONSE_CODES	93
Table 73 – Builtin IGNORE_COMM_ERROR	93
Table 74 – Builtin IGNORE_COMM_STATUS	94
Table 75 – Builtin IGNORE_DEVICE_STATUS	94
Table 76 – Builtin IGNORE_NO_DEVICE.....	95
Table 77 – Builtin IGNORE_RESPONSE_CODE	95
Table 78 – Builtin read_value	96
Table 79 – Builtin read_value2	97
Table 80 – Builtin ReadCommand.....	97
Table 81 – Builtin retry_on_all_comm_errors.....	98
Table 82 – Builtin RETRY_ON_ALL_COMM_STATUS	98
Table 83 – Builtin RETRY_ON_ALL_DEVICE_STATUS	99
Table 84 – Builtin RETRY_ON_ALL_RESPONSE_CODES.....	99
Table 85 – Builtin retry_on_all_response_codes.....	100
Table 86 – Builtin RETRY_ON_COMM_ERROR.....	101
Table 87 – Builtin retry_on_comm_error	101
Table 88 – Builtin RETRY_ON_COMM_STATUS.....	102
Table 89 – Builtin RETRY_ON_DEVICE_STATUS.....	102
Table 90 – Builtin RETRY_ON_NO_DEVICE	103
Table 91 – Builtin RETRY_ON_RESPONSE_CODE.....	103
Table 92 – Builtin retry_on_response_code	104
Table 93 – Builtin send	105
Table 94 – Builtin send_all_values	105
Table 95 – Builtin send_command	106

Table 96 – Builtin send_command_trans	106
Table 97 – Builtin send_trans	107
Table 98 – Builtin send_value	108
Table 99 – Builtin send_value2	109
Table 100 – Builtin SET_NUMBER_OF_RETRIES	110
Table 101 – Builtin WriteCommand	110
Table 102 – Builtin XMTR_ABORT_ON_ALL_COMM_STATUS	111
Table 103 – Builtin XMTR_ABORT_ON_ALL_DATA	111
Table 104 – Builtin XMTR_ABORT_ON_ALL_DEVICE_STATUS	112
Table 105 – Builtin XMTR_ABORT_ON_ALL_RESPONSE_CODES	112
Table 106 – Builtin XMTR_ABORT_ON_COMM_ERROR	113
Table 107 – Builtin XMTR_ABORT_ON_COMM_STATUS	113
Table 108 – Builtin XMTR_ABORT_ON_DATA	114
Table 109 – Builtin XMTR_ABORT_ON_DEVICE_STATUS	114
Table 110 – Builtin XMTR_ABORT_ON_NO_DEVICE	115
Table 111 – Builtin XMTR_ABORT_ON_RESPONSE_CODE	115
Table 112 – Builtin XMTR_IGNORE_ALL_COMM_STATUS	116
Table 113 – Builtin XMTR_IGNORE_ALL_DATA	116
Table 114 – Builtin XMTR_IGNORE_ALL_DEVICE_STATUS	117
Table 115 – Builtin XMTR_IGNORE_ALL_RESPONSE_CODES	117
Table 116 – Builtin XMTR_IGNORE_COMM_ERROR	118
Table 117 – Builtin XMTR_IGNORE_COMM_STATUS	118
Table 118 – Builtin XMTR_IGNORE_DATA	119
Table 119 – Builtin XMTR_IGNORE_DEVICE_STATUS	119
Table 120 – Builtin XMTR_IGNORE_NO_DEVICE	120
Table 121 – Builtin XMTR_IGNORE_RESPONSE_CODE	120
Table 122 – Builtin XMTR_RETRY_ON_ALL_COMM_STATUS	121
Table 123 – Builtin XMTR_RETRY_ON_ALL_DATA	121
Table 124 – Builtin XMTR_RETRY_ON_ALL_DEVICE_STATUS	122
Table 125 – Builtin XMTR_RETRY_ON_ALL_RESPONSE_CODES	122
Table 126 – Builtin XMTR_RETRY_ON_COMM_ERROR	123
Table 127 – Builtin XMTR_RETRY_ON_COMM_STATUS	123
Table 128 – Builtin XMTR_RETRY_ON_DATA	124
Table 129 – Builtin XMTR_RETRY_ON_DEVICE_STATUS	124
Table 130 – Builtin XMTR_RETRY_ON_NO_DEVICE	125
Table 131 – Builtin XMTR_RETRY_ON_RESPONSE_CODE	125
Table 132 – Builtin fgetval	126
Table 133 – Builtin fsetval	126
Table 134 – Builtin get_date	127
Table 135 – Builtin get_double	128
Table 136 – Builtin get_float	128
Table 137 – Builtin get_signed	129

Table 138 – Builtin get_string	130
Table 139 – Builtin get_unsigned	130
Table 140 – Builtin igetval.....	131
Table 141 – Builtin isetval.....	131
Table 142 – Builtin isOffline	132
Table 143 – Builtin lgetval.....	132
Table 144 – Builtin lsetval.....	133
Table 145 – Builtin put_date	133
Table 146 – Builtin put_double	134
Table 147 – Builtin put_float	135
Table 148 – Builtin put_signed	135
Table 149 – Builtin put_string	136
Table 150 – Builtin put_unsigned	137
Table 151 – Builtin sgetval.....	137
Table 152 – Builtin ssetval.....	138
Table 153 – Builtin assign.....	138
Table 154 – Builtin assign_double.....	139
Table 155 – Builtin assign_float.....	140
Table 156 – Builtin assign_int.....	140
Table 157 – Builtin assign_var.....	141
Table 158 – Builtin assign2.....	142
Table 159 – Builtin dassign.....	142
Table 160 – Builtin fassign.....	143
Table 161 – Builtin float_value.....	143
Table 162 – Builtin fvar_value	144
Table 163 – Builtin get_date_value	144
Table 164 – Builtin get_date_value2	145
Table 165 – Builtin get_double_value	146
Table 166 – Builtin get_double_value2	147
Table 167 – Builtin get_float_value.....	148
Table 168 – Builtin get_float_value2	149
Table 169 – builtin get_signed_value	149
Table 170 – Builtin get_signed_value2.....	150
Table 171 – Builtin get_string_value	151
Table 172 – Builtin get_string_value2	152
Table 173 – Builtin get_unsigned_value.....	153
Table 174 – Builtin get_unsigned_value2.....	154
Table 175 – Builtin iassign.....	155
Table 176 – Builtin int_value.....	155
Table 177 – Builtin ivar_value.....	156
Table 178 – Builtin lassign.....	156
Table 179 – Builtin long_value.....	157

Table 180 – Builtin lvar_value	157
Table 181 – Builtin put_date_value	158
Table 182 – Builtin put_date_value2	159
Table 183 – Builtin put_double_value	160
Table 184 – Builtin put_double_value2	160
Table 185 – Builtin put_float_value	161
Table 186 – Builtin put_float_value2	162
Table 187 – Builtin put_signed_value	163
Table 188 – Builtin put_signed_value2	164
Table 189 – Builtin put_string_value	165
Table 190 – Builtin put_string_value2	166
Table 191 – Builtin put_unsigned_value	167
Table 192 – Builtin put_unsigned_value2	168
Table 193 – Builtin ret_double_value	168
Table 194 – Builtin ret_double_value2	169
Table 195 – Builtin ret_float_value	169
Table 196 – Builtin ret_float_value2	170
Table 197 – Builtin ret_signed_value	171
Table 198 – Builtin ret_signed_value2	171
Table 199 – Builtin ret_unsigned_value	172
Table 200 – Builtin ret_unsigned_value2	172
Table 201 – Builtin vassign	173
Table 202 – Builtin discard_on_exit	174
Table 203 – Builtin save_on_exit	175
Table 204 – Builtin save_values.....	175
Table 205 – Builtin send_on_exit	176
Table 206 – Builtin abort.....	177
Table 207 – Builtin add_abort_method	177
Table 208 – Builtin add_abort_method	178
Table 209 – Builtin method_abort	178
Table 210 – Builtin pop_abort_method	179
Table 211 – Builtin process_abort.....	179
Table 212 – Builtin push_abort_method.....	180
Table 213 – Builtin remove_abort_method	180
Table 214 – Builtin remove_abort_method	181
Table 215 – Builtin remove_all_abort_methods.....	181
Table 216 – Builtin atof.....	182
Table 217 – Builtin atoi	182
Table 218 – Builtin BUILD_MESSAGE	183
Table 219 – Builtin dictionary_string	183
Table 220 – Builtin ftoa.....	184
Table 221 – Builtin get_comm_error_string	184

Table 222 – Builtin get_dictionary_string.....	185
Table 223 – Builtin get_enum_string	186
Table 224 – Builtin get_response_code_string	186
Table 225 – Builtin get_rspcode_string	187
Table 226 – Builtin get_rspcode_string_by_id.....	188
Table 227 – Builtin get_status_code_string.....	188
Table 228 – Builtin get_status_string.....	189
Table 229 – Builtin get_stddict_string.....	190
Table 230 – Builtin get_variable_string	190
Table 231 – Builtin itoa (version A).....	191
Table 232 – Builtin itoa (version B).....	191
Table 233 – Builtin strleft	191
Table 234 – Builtin strcmp	192
Table 235 – Builtin strlen	192
Table 236 – Builtin strlwr	193
Table 237 – Builtin strmid	194
Table 238 – Builtin stright.....	194
Table 239 – Builtin strstr	195
Table 240 – Builtin strtrim.....	195
Table 241 – Builtinstrupr.....	196
Table 242 – Builtin trunc.....	196
Table 243 – Builtin AddTime.....	197
Table 244 – Builtin DATE_AND_TIME_VALUE_to_string	198
Table 245 – Builtin Date_to_DayOfMonth	198
Table 246 – Builtin DATE_to_days.....	199
Table 247 – Builtin Date_to_Month	199
Table 248 – Builtin DATE_to_string.....	200
Table 249 – Builtin Date_To_Time	200
Table 250 – Builtin Date_to_Year.....	200
Table 251 – Builtin days_to_DATE.....	201
Table 252 – Builtin DELAY_TIME.....	201
Table 253 – Builtin DiffTime.....	202
Table 254 – Builtin From_DATE_AND_TIME_VALUE.....	202
Table 255 – Builtin From_TIME_VALUE	203
Table 256 – Builtin GetCurrentDate.....	203
Table 257 – Builtin GetCurrentDateAndTime	203
Table 258 – Builtin GetCurrentTime	204
Table 259 – Builtin GET_TICK_COUNT	204
Table 260 – Builtin Make_Time	205
Table 261 – Builtin seconds_to_TIME_VALUE	205
Table 262 – Builtin seconds_to_TIME_VALUE	206
Table 263 – Builtin Time_To_Date	206

Table 264 – Builtin TIME_VALUE_to_Hour	207
Table 265 – Builtin TIME_VALUE_to_Minute	207
Table 266 – Builtin TIME_VALUE_to_Second	208
Table 267 – Builtin TIME_VALUE_to_seconds	208
Table 268 – Builtin TIME_VALUE_to_string	209
Table 269 – Builtin timet_to_string	210
Table 270 – Builtin timet_to_TIME_VALUE	210
Table 271 – Builtin timet_To_TIME_VALUE	211
Table 272 – Builtin timet_to_TIME_VALUE8	211
Table 273 – Builtin To_Date	211
Table 274 – Builtin To_Date_and_Time	212
Table 275 – Builtin To_Time	212
Table 276 – Builtin To_TIME_VALUE	213
Table 277 – Builtin To_TIME_VALUE8	213
Table 278 – Builtin get_comm_error	214
Table 279 – Builtin get_dds_error	215
Table 280 – Builtin get_response_code	216
Table 281 – Builtin abs	216
Table 282 – Builtin acos	217
Table 283 – Builtin asin	217
Table 284 – Builtin atan	218
Table 285 – Builtin ByteToDouble	218
Table 286 – Builtin ByteToFloat	219
Table 287 – Builtin ByteToLong	219
Table 288 – Builtin ByteToShort	220
Table 289 – Builtin cbrt	220
Table 290 – Builtin ceil	221
Table 291 – Builtin cos	221
Table 292 – Builtin cosh	221
Table 293 – Builtin DoubleToByte	222
Table 294 – Builtin drand	222
Table 295 – Builtin dseed	223
Table 296 – Builtin exp	223
Table 297 – Builtin FloatToByte	224
Table 298 – Builtin floor	224
Table 299 – Builtin fmod	225
Table 300 – Builtin fpclassify	225
Table 301 – Builtin is_NaN	226
Table 302 – Builtin log	226
Table 303 – Builtin log10	227
Table 304 – Builtin log2	227
Table 305 – Builtin LongToByte	228

Table 306 – Builtin nan	228
Table 307 – Builtin NaN_value	229
Table 308 – Builtin nanf	230
Table 309 – Builtin pow	230
Table 310 – Builtin round	230
Table 311 – Builtin ShortToByte	231
Table 312 – Builtin sin	231
Table 313 – Builtin sinh	232
Table 314 – Builtin sqrt	232
Table 315 – Builtin tan	233
Table 316 – Builtin tanh	233
Table 317 – Builtin get_date_lelem	234
Table 318 – Builtin get_date_lelem2	235
Table 319 – Builtin get_double_lelem	236
Table 320 – Builtin get_double_lelem2	237
Table 321 – Builtin get_float_lelem	237
Table 322 – Builtin get_float_lelem2	238
Table 323 – Builtin get_signed_lelem	239
Table 324 – Builtin get_signed_lelem2	240
Table 325 – Builtin get_string_lelem	241
Table 326 – Builtin get_string_lelem2	242
Table 327 – Builtin get_unsigned_lelem	243
Table 328 – Builtin get_unsigned_lelem2	244
Table 329 – Builtin ListDeleteElementAt	245
Table 330 – Builtin ListDeleteElementAt	245
Table 331 – Builtin ListDeleteElementAt2	246
Table 332 – Builtin ListInsert	247
Table 333 – Builtin ListInsert	247
Table 334 – Builtin ListInsert2	248
Table 335 – Builtin DICT_ID	249
Table 336 – Builtin get_block_instance_by_object_index	249
Table 337 – Builtin get_block_instance_by_tag	250
Table 338 – Builtin get_block_instance_count	251
Table 339 – Builtin get_resolve_status	252
Table 340 – Builtin ITEM_ID	252
Table 341 – Builtin MEMBER_ID	253
Table 342 – Builtin ObjectReference	253
Table 343 – Builtin resolve_array_ref	254
Table 344 – Builtin resolve_array_ref2	255
Table 345 – Builtin resolve_block_ref	256
Table 346 – Builtin resolve_block_ref2	257
Table 347 – Builtin resolve_list_ref	257

Table 348 – Builtin resolve_local_ref	258
Table 349 – Builtin resolve_local_ref2	258
Table 350 – Builtin resolve_param_list_ref	259
Table 351 – Builtin resolve_param_ref	260
Table 352 – Builtin resolve_param_ref2	260
Table 353 – Builtin resolve_record_ref	261
Table 354 – Builtin resolve_record_ref2	262
Table 355 – Builtin VARID	262
Table 356 – Builtin abortTransferPort	263
Table 357 – Builtin closeTransferPort	263
Table 358 – Builtin fGetByte	264
Table 359 – Builtin get_transfer_status	264
Table 360 – Builtin openTransferPort	265
Table 361 – Builtin readItemFromDevice	265
Table 362 – Builtin writeItemToDevice	266
Table 363 – Builtin browseIdentity	267
Table 364 – Builtin re_read_file	267
Table 365 – Builtin re_write_file	267
Table 366 – Builtin GET_DD_REVISION	268
Table 367 – Builtin GET_DEVICE_REVISION	268
Table 368 – Builtin GET_DEVICE_TYPE	269
Table 369 – Builtin GET_MANUFACTURER	269
Table 370 – Builtin _ERROR	270
Table 371 – Builtin _TRACE	270
Table 372 – Builtin _WARNING	271
Table 373 – Builtin LOG_MESSAGE	271
Table 374 – Referencable EDD item attributes	273
Table 375 – Contents of the return codes description table	276
Table 376 – Return code descriptions	276
Table 377 – Return code descriptions	276

INTRODUCTION

This part of IEC 61804

- contains a list of builtins that can be used inside of EDDL methods;
- defines the functionality and parameters for each builtin;
- describes how EDDL variables can be referenced from inside EDDL methods;

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

DEVICES AND INTEGRATION IN ENTERPRISE SYSTEMS – FUNCTION BLOCKS (FB) FOR PROCESS CONTROL AND ELECTRONIC DEVICE DESCRIPTION LANGUAGE (EDDL) –

Part 5: EDDL Builtin library

1 Scope

This part of IEC 61804 specifies the EDDL builtin library and provides the profiles of the various fieldbuses.

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 61804-3:2020, *Devices and integration in enterprise systems – Function blocks (FB) for process control and electronic device description language (EDDL) – Part 3: EDDL syntax and semantics*

ISO/IEC 9899, *Information technology – Programming Languages – C*

IEEE 754, *IEEE Standard for Binary Floating-Point Arithmetic*

3 Terms, definitions, acronyms and abbreviated terms

3.1 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:

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

3.1.1

method-variable

variable with scope within a method

3.2 Acronyms and abbreviated terms

CP	Communication Profile
CPF	Communication Profile Family
CS	Communication Server
DD	Device Description
EDD	Electronic Device Description
EDDL	Electronic Device Description Language
FF	Foundation Fieldbus

GPE	Generic Protocol Extension
HART	HART® Communication Foundation
I/O	Input/output
ID	Identifier
ISA100	ISA100_Wireless™
PB	PROFIBUS
PI	PROFIBUS & PROFINET ^{Fehler Textmarke nicht definiert.} International
PN	PROFINET
PNO	PROFIBUS Nutzerorganisation

4 Conventions for builtin descriptions

The following conventions apply for the builtin overview.

- Table 1 defines the format for the builtin overview table.
- Table 2 defines the contents of the columns of Table 1.
- EDDL basic constructs and attributes referenced in this document are written in upper case letters.

Table 1 – Format for the builtin overview table

Reference	Builtin	Communication profiles

Table 2 – Contents of the builtin overview table

Column	Text	Meaning
Reference		Reference to the respective subclause that specifies the builtin.
Builtin		Name of the builtin.
Communication profiles	✓	The builtin is supported by the respective Communication profile.
	x	The builtin is not supported by the respective Communication profile.
	d	The usage of the builtin is deprecated.

The following conventions apply for the builtin definitions.

- Table 3 defines the format for the builtins lexical element tables.
- Table 4 defines the contents of the columns of Table 3.
- EDDL basic constructs and attributes referenced in this document are written in upper case letters.

Table 3 – Format for the builtins lexical element tables

Parameter name	Type	Direction	Usage	Description

Table 4 – Contents of the lexical element table

Column	Text	Meaning
Parameter name	<name>	The name of the formal parameter.
	<return>	The return value of the builtin.
Type	data type	Abstract data type (basic or derived) of the formal parameter.
Direction	I	An input parameter.
	O	An output parameter is implicitly passed by reference so that the called builtin function can modify the value. The return value of a builtin is passed by value, if any.
	I/O	An input output parameter is implicitly passed by reference so that the called builtin function can modify the value.
	—	There is no returned value (VOID).
Usage	m	Formal parameter is mandatory.
	o	Formal parameter is optional.
	c	Formal parameter is conditional.
	—	There is no returned value (VOID).
Description	<text>	Descriptive text for the parameter.
	—	There is no returned value (VOID).

5 Builtin categories

5.1 Overview

Table 5 provides an alphabetical overview of the EDDL builtin library and specify the profile per communication technology.

Table 5 – Builtin overview

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.17.1	_ERROR	✓	✓	✓	✓	✓	✓	✓
6.17.2	_TRACE	✓	✓	✓	✓	✓	✓	✓
6.17.3	_WARNING	✓	✓	✓	✓	✓	✓	✓
6.7.1	abort	✓	✗	✓	✓	✗	✓	✓
6.3.1	abort_on_all_comm_errors	✗	✓	✗	✗	✓	✗	✗
6.3.2	ABORT_ON_ALL_COMM_STATUS	✗	✗	✗	✓	✗	✗	✗
6.3.3	ABORT_ON_ALL_DEVICE_STATUS	✗	✗	✗	✓	✗	✗	✗
6.3.4	ABORT_ON_ALL_RESPONSE_CODES	✗	✗	✗	✓	✗	✗	✗
6.3.5	abort_on_all_response_codes	✗	✓	✗	✗	✓	✗	✗
6.3.6	ABORT_ON_COMM_ERROR	✗	✗	✗	✓	✗	✗	✗
6.3.7	abort_on_comm_error	✗	✓	✗	✗	✓	✗	✗
6.3.8	ABORT_ON_COMM_STATUS	✗	✗	✗	✓	✗	✗	✗
6.3.9	ABORT_ON_DEVICE_STATUS	✗	✗	✗	✓	✗	✗	✗

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.3.10	ABORT_ON_NO_DEVICE (deprecated)	x	x	x	d	x	x	x
6.3.11	ABORT_ON_RESPONSE_CODE	x	x	x	✓	x	x	x
6.3.12	abort_on_response_code	x	✓	x	x	✓	x	x
6.14.1	abortTransferPort	x	x	x	✓	x	x	x
6.11.1	abs	✓	✓	✓	✓	✓	✓	✓
6.2.1	ACKNOWLEDGE	✓	x	✓	✓	x	✓	✓
6.2.2	acknowledge	✓	x	✓	✓	x	✓	✓
6.11.2	acos	✓	✓	✓	✓	✓	✓	✓
6.7.2	add_abort_method (version A)	✓	x	✓	✓	x	✓	✓
6.7.3	add_abort_method (version B)	x	✓	x	x	✓	x	x
6.9.1	AddTime	✓	✓	✓	✓	✓	✓	✓
6.11.3	asin	✓	✓	✓	✓	✓	✓	✓
6.5.1	assign	x	✓	x	x	✓	x	x
6.5.2	assign_double	✓	x	✓	✓	x	✓	✓
6.5.3	assign_float	✓	x	✓	✓	x	✓	✓
6.5.4	assign_int	✓	x	✓	✓	x	✓	✓
6.5.5	assign_var	✓	x	✓	✓	x	✓	✓
6.5.6	assign2	x	✓	x	x	✓	x	x
6.11.4	atan	✓	✓	✓	✓	✓	✓	✓
6.8.1	atof	✓	x	✓	✓	x	✓	✓
6.8.2	atoi	✓	x	✓	✓	x	✓	✓
6.15.1	browseIdentity	x	x	x	✓	x	x	x
6.8.3	BUILD_MESSAGE	✓	✓	✓	✓	✓	✓	✓
6.11.5	ByteToDouble	✓	x	✓	x	x	✓	✓
6.11.6	ByteToFloat	✓	x	✓	x	x	✓	✓
6.11.7	ByteToLong	✓	x	✓	x	x	✓	✓
6.11.8	ByteToShort	✓	x	✓	x	x	✓	✓
6.11.9	cbrt	✓	✓	✓	✓	✓	✓	✓
6.11.10	ceil	✓	✓	✓	✓	✓	✓	✓
6.14.2	closeTransferPort	x	x	x	✓	x	x	x
6.11.11	cos	✓	✓	✓	✓	✓	✓	✓
6.11.12	cosh	✓	✓	✓	✓	✓	✓	✓
6.5.7	dassign	✓	x	✓	✓	x	✓	✓
6.9.2	DATE_AND_TIME_VALUE_to_string	✓	x	✓	✓	x	✓	✓
6.9.3	Date_to_DayOfMonth	✓	x	✓	✓	x	✓	✓
6.9.4	DATE_to_days	✓	x	✓	✓	x	✓	✓
6.9.5	Date_to_Month	✓	x	✓	✓	x	✓	✓
6.9.6	DATE_to_string	✓	x	✓	✓	x	✓	✓
6.9.7	Date_To_Time	✓	x	✓	✓	x	✓	✓

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.9.8	Date_to_Year	✓	✗	✓	✓	✗	✓	✓
6.9.9	days_to_DATE	✓	✗	✓	✓	✗	✓	✓
6.2.3	DELAY	✓	✗	✓	✓	✗	✓	✓
6.2.4	delay	✓	✗	✓	✓	✗	✓	✓
6.9.10	DELAY_TIME	✓	✗	✓	✓	✗	✓	✓
6.2.5	delayfor	✗	✓	✗	✗	✓	✗	✗
6.2.6	delayfor2	✗	✓	✗	✗	✓	✗	✗
6.13.1	DICT_ID	✗	✓	✗	✗	✓	✗	✗
6.8.4	dictionary_string	✓	✗	✓	✓	✗	✓	✓
6.9.11	DiffTime	✓	✓	✓	✓	✓	✓	✓
6.6.1	discard_on_exit	✓	✓	✓	✓	✓	✓	✓
6.2.7	DISPLAY	✓	✗	✓	✓	✗	✓	✓
6.2.8	display	✓	✗	✓	✓	✗	✓	✓
6.2.9	display_bitenum	✓	✗	✓	✗	✗	✓	✓
6.2.10	display_builtin_error	✗	✓	✗	✗	✓	✗	✗
6.2.11	display_comm_error	✗	✓	✗	✗	✓	✗	✗
6.2.12	display_comm_status	✗	✗	✗	✓	✗	✗	✗
6.2.13	display_device_status	✗	✗	✗	✓	✗	✗	✗
6.2.14	display_dynamics	✗	✓	✗	✗	✓	✗	✗
6.2.15	display_dynamics2	✗	✓	✗	✗	✓	✗	✗
6.2.16	display_message	✗	✓	✗	✗	✓	✗	✗
6.2.17	display_message2	✗	✓	✗	✗	✓	✗	✗
6.2.18	display_response_code	✗	✓	✗	✗	✓	✗	✗
6.2.19	display_response_status	✗	✗	✗	✓	✗	✗	✗
6.2.20	display_xmtr_status	✗	✗	✗	✓	✗	✗	✗
6.11.13	DoubleToByte	✓	✗	✓	✗	✗	✓	✓
6.11.14	drand	✓	✓	✓	✓	✓	✓	✓
6.11.15	dseed	✓	✓	✓	✓	✓	✓	✓
6.2.21	edit_device_value	✗	✓	✗	✗	✓	✗	✗
6.2.22	edit_device_value2	✗	✓	✗	✗	✓	✗	✗
6.2.23	edit_local_value	✗	✓	✗	✗	✓	✗	✗
6.2.24	edit_local_value2	✗	✓	✗	✗	✓	✗	✗
6.11.16	exp	✓	✓	✓	✓	✓	✓	✓
6.3.13	ext_send_command	✗	✗	✗	✓	✗	✗	✗
6.3.14	ext_send_command_trans	✗	✗	✗	✓	✗	✗	✗
6.3.15	fail_on_all_comm_errors	✗	✓	✗	✗	✓	✗	✗
6.3.16	fail_on_all_response_codes	✗	✓	✗	✗	✓	✗	✗
6.3.17	fail_on_comm_error	✗	✓	✗	✗	✓	✗	✗
6.3.18	fail_on_response_code	✗	✓	✗	✗	✓	✗	✗

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.5.8	fassign	✓	✗	✓	✓	✗	✓	✓
6.14.2	fGetByte	✗	✗	✗	✓	✗	✗	✗
6.4.1	fgetval	✓	✗	✓	✓	✗	✓	✓
6.5.9	float_value	✓	✗	✓	✓	✗	✓	✓
6.11.17	FloatToByte	✓	✗	✓	✗	✗	✓	✓
6.11.18	floor	✓	✓	✓	✓	✓	✓	✓
6.11.19	fmod	✓	✓	✓	✓	✓	✓	✓
6.11.20	fpclassify	✓	✓	✓	✓	✓	✓	✓
6.9.12	From_DATE_AND_TIME_VALUE	✓	✗	✓	✓	✗	✓	✓
6.9.13	From_TIME_VALUE	✓	✗	✓	✓	✗	✓	✓
6.4.2	fsetval	✓	✗	✓	✓	✗	✓	✓
6.8.5	ftoa	✓	✗	✓	✗	✗	✓	✓
6.5.10	fvar_value	✓	✗	✓	✓	✗	✓	✓
6.2.25	get_acknowledgement	✗	✓	✗	✗	✓	✗	✗
6.2.26	get_acknowledgement2	✗	✓	✗	✗	✓	✗	✗
6.13.2	get_block_instance_by_object_index	✗	✓	✗	✗	✓	✗	✗
6.13.3	get_block_instance_by_tag	✗	✓	✗	✗	✓	✗	✗
6.13.4	get_block_instance_count	✗	✓	✗	✗	✓	✗	✗
6.10.1	get_comm_error	✗	✓	✗	✗	✓	✗	✗
6.8.6	get_comm_error_string	✗	✓	✗	✗	✓	✗	✗
6.4.3	get_date	✗	✓	✗	✗	✓	✗	✗
6.12.1	get_date_lelem	✗	✓	✗	✗	✓	✗	✗
6.12.2	get_date_lelem2	✗	✓	✗	✗	✓	✗	✗
6.5.11	get_date_value	✗	✓	✗	✗	✓	✗	✗
6.5.12	get_date_value2	✗	✓	✗	✗	✓	✗	✗
6.16.1	GET_DD_REVISION	✓	✗	✓	✗	✗	✓	✓
6.10.2	get_dds_error	✗	✓	✗	✗	✓	✗	✗
6.2.27	GET_DEV_VAR_VALUE	✓	✗	✓	✓	✗	✓	✓
6.2.28	get_dev_var_value	✓	✗	✓	✓	✗	✓	✓
6.16.2	GET_DEVICE_REVISION	✓	✗	✓	✗	✗	✓	✓
6.16.3	GET_DEVICE_TYPE	✓	✗	✓	✗	✗	✓	✓
6.8.7	get_dictionary_string	✓	✗	✓	✓	✗	✓	✓
6.4.4	get_double	✗	✓	✗	✗	✓	✗	✗
6.12.3	get_double_lelem	✗	✓	✗	✗	✓	✗	✗
6.12.4	get_double_lelem2	✗	✓	✗	✗	✓	✗	✗
6.5.13	get_double_value	✗	✓	✗	✗	✓	✗	✗
6.5.14	get_double_value2	✗	✓	✗	✗	✓	✗	✗
6.8.8	get_enum_string	✓	✗	✓	✓	✗	✓	✓
6.4.5	get_float	✗	✓	✗	✗	✓	✗	✗

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.12.5	get_float_lelem	x	✓	x	x	✓	x	x
6.12.6	get_float_lelem2	x	✓	x	x	✓	x	x
6.5.15	get_float_value	x	✓	x	x	✓	x	x
6.5.16	get_float_value2	x	✓	x	x	✓	x	x
6.2.29	GET_LOCAL_VAR_VALUE	✓	x	✓	✓	x	✓	✓
6.2.30	get_local_var_value	✓	x	✓	✓	x	✓	✓
6.16.4	GET_MANUFACTURER	✓	x	✓	✓	x	✓	✓
6.3.19	get_more_status	x	x	x	✓	x	x	x
6.13.5	get_resolve_status	x	✓	x	x	✓	x	x
6.10.3	get_response_code	x	✓	x	x	✓	x	x
6.8.9	get_response_code_string	x	✓	x	x	✓	x	x
6.8.10	get_rspcode_string	x	x	x	✓	x	x	x
6.8.11	get_rspcode_string_by_id	✓	x	✓	x	x	✓	✓
6.4.6	get_signed	x	✓	x	x	✓	x	x
6.12.7	get_signed_lelem	x	✓	x	x	✓	x	x
6.12.8	get_signed_lelem2	x	✓	x	x	✓	x	x
6.5.17	get_signed_value	x	✓	x	x	✓	x	x
6.5.18	get_signed_value2	x	✓	x	x	✓	x	x
6.8.12	get_status_code_string(deprecated)	x	x	x	d	x	x	x
6.8.13	get_status_string	x	✓	x	x	✓	x	x
6.8.14	get_stddict_string	x	✓	x	x	✓	x	x
6.4.7	get_string	x	✓	x	x	✓	x	x
6.12.9	get_string_lelem	x	✓	x	x	✓	x	x
6.12.10	get_string_lelem2	x	✓	x	x	✓	x	x
6.5.19	get_string_value	x	✓	x	x	✓	x	x
6.5.20	get_string_value2	x	✓	x	x	✓	x	x
6.9.17	GET_TICK_COUNT	✓	x	✓	x	x	✓	✓
6.14.4	get_transfer_status	x	x	x	✓	x	x	x
6.4.8	get_unsigned	x	✓	x	x	✓	x	x
6.12.11	get_unsigned_lelem	x	✓	x	x	✓	x	x
6.12.12	get_unsigned_lelem2	x	✓	x	x	✓	x	x
6.5.21	get_unsigned_value	x	✓	x	x	✓	x	x
6.5.22	get_unsigned_value2	x	✓	x	x	✓	x	x
6.8.15	get_variable_string	✓	x	✓	x	x	✓	✓
6.9.14	GetCurrentDate	✓	✓	✓	x	✓	✓	✓
6.9.15	GetCurrentDateAndTime	✓	✓	✓	x	✓	✓	✓
6.9.16	GetCurrentTime	✓	✓	✓	✓	✓	✓	✓
6.5.23	iassign	✓	x	✓	✓	x	✓	✓
6.4.9	igetval	✓	x	✓	✓	x	✓	✓

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.3.20	IGNORE_ALL_COMM_STATUS	x	x	x	✓	x	x	x
6.3.21	IGNORE_ALL_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.22	IGNORE_ALL_RESPONSE_CODES	x	x	x	✓	x	x	x
6.3.23	IGNORE_COMM_ERROR	x	x	x	✓	x	x	x
6.3.24	IGNORE_COMM_STATUS	x	x	x	✓	x	x	x
6.3.25	IGNORE_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.26	IGNORE_NO_DEVICE (deprecated)	x	x	x	d	x	x	x
6.3.27	IGNORE_RESPONSE_CODE	x	x	x	✓	x	x	x
6.5.24	int_value	✓	x	✓	✓	x	✓	✓
6.11.21	is_NaN	✓	✓	✓	x	✓	✓	✓
6.4.10	isetval	✓	x	✓	✓	x	✓	✓
6.4.11	isOffline	✓	✓	✓	✓	✓	✓	✓
6.13.6	ITEM_ID	x	✓	x	x	✓	x	x
6.8.16	itoa (version A)	✓	x	✓	x	x	✓	✓
6.8.17	itoa (version B)	x	x	x	✓	x	x	x
6.5.25	ivar_value	✓	x	✓	✓	x	✓	✓
6.5.26	lassign	✓	x	✓	x	x	✓	✓
6.4.12	lgetval	✓	x	✓	x	x	✓	✓
6.12.13	ListDeleteElementAt (version A)	✓	x	✓	✓	x	✓	✓
6.12.14	ListDeleteElementAt (version B)	x	✓	x	x	✓	x	x
6.12.15	ListDeleteElementAt2	x	✓	x	x	✓	x	x
6.12.16	ListInsert (version A)	✓	x	✓	✓	x	✓	✓
6.12.17	ListInsert (version B)	x	✓	x	x	✓	x	x
6.12.18	ListInsert2	x	✓	x	x	✓	x	x
6.11.22	log	✓	✓	✓	✓	✓	✓	✓
6.17.4	LOG_MESSAGE	✓	x	✓	✓	x	✓	✓
6.11.23	log10	✓	✓	✓	✓	✓	✓	✓
6.11.24	log2	✓	✓	✓	✓	✓	✓	✓
6.5.27	long_value	✓	x	✓	x	x	✓	✓
6.11.25	LongToByte	✓	x	✓	x	x	✓	✓
6.4.13	lsetval	✓	x	✓	x	x	✓	✓
6.5.28	lvar_value	✓	x	✓	x	x	✓	✓
6.9.18	Make_Time	✓	✓	✓	✓	✓	✓	✓
6.13.7	MEMBER_ID	x	✓	x	x	✓	x	x
6.2.31	Menu	✓	x	✓	✓	x	✓	✓
6.2.32	MenuDisplay (version A)	✓	x	✓	✓	x	✓	✓
6.2.33	MenuDisplay (version B)	x	✓	x	x	✓	x	x
6.7.4	method_abort	x	✓	x	x	✓	x	x
6.11.26	nan	✓	✓	✓	✓	✓	✓	✓

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.11.27	NaN_value	✓	✓	✓	✗	✓	✓	✓
6.11.28	nanf	✓	✓	✓	✓	✓	✓	✓
6.13.8	ObjectReference	✓	✗	✓	✗	✗	✓	✓
6.14.5	openTransferPort	✗	✗	✗	✓	✗	✗	✗
6.7.5	pop_abort_method	✓	✗	✓	✓	✗	✓	✓
6.11.29	pow	✓	✓	✓	✓	✓	✓	✓
6.7.6	process_abort	✓	✗	✓	✓	✗	✓	✓
6.7.7	push_abort_method	✓	✗	✓	✓	✗	✓	✓
6.4.14	put_date	✗	✓	✗	✗	✓	✗	✗
6.5.29	put_date_value	✗	✓	✗	✗	✓	✗	✗
6.5.30	put_date_value2	✗	✓	✗	✗	✓	✗	✗
6.4.15	put_double	✗	✓	✗	✗	✓	✗	✗
6.5.31	put_double_value	✗	✓	✗	✗	✓	✗	✗
6.5.32	put_double_value2	✗	✓	✗	✗	✓	✗	✗
6.4.16	put_float	✗	✓	✗	✗	✓	✗	✗
6.5.33	put_float_value	✗	✓	✗	✗	✓	✗	✗
6.5.34	put_float_value2	✗	✓	✗	✗	✓	✗	✗
6.2.34	PUT_MESSAGE	✓	✗	✓	✓	✗	✓	✓
6.2.35	put_message	✓	✗	✓	✓	✗	✓	✓
6.4.17	put_signed	✗	✓	✗	✗	✓	✗	✗
6.5.35	put_signed_value	✗	✓	✗	✗	✓	✗	✗
6.5.36	put_signed_value2	✗	✓	✗	✗	✓	✗	✗
6.4.18	put_string	✗	✓	✗	✗	✓	✗	✗
6.5.37	put_string_value	✗	✓	✗	✗	✓	✗	✗
6.5.38	put_string_value2	✗	✓	✗	✗	✓	✗	✗
6.4.19	put_unsigned	✗	✓	✗	✗	✓	✗	✗
6.5.39	put_unsigned_value	✗	✓	✗	✗	✓	✗	✗
6.5.40	put_unsigned_value2	✗	✓	✗	✗	✓	✗	✗
6.15.2	re_read_file	✗	✗	✗	✓	✗	✗	✗
6.15.3	re_write_file	✗	✗	✗	✓	✗	✗	✗
6.3.28	read_value	✗	✓	✗	✗	✓	✗	✗
6.3.29	read_value2	✗	✓	✗	✗	✓	✗	✗
6.3.30	ReadCommand	✓	✗	✓	✗	✗	✓	✓
6.14.6	readItemFromDevice	✗	✗	✗	✓	✗	✗	✗
6.7.8	remove_abort_method (version A)	✓	✗	✓	✓	✗	✓	✓
6.7.9	remove_abort_method (version B)	✗	✓	✗	✗	✓	✗	✗
6.7.10	remove_all_abort_methods	✓	✓	✓	✓	✓	✓	✓
6.13.9	resolve_array_ref	✗	✓	✗	✗	✓	✗	✗
6.13.10	resolve_array_ref2	✗	✓	✗	✗	✓	✗	✗

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.13.11	resolve_block_ref	x	✓	x	x	✓	x	x
6.13.12	resolve_block_ref2	x	✓	x	x	✓	x	x
6.13.13	resolve_list_ref	x	✓	x	x	✓	x	x
6.13.14	resolve_local_ref	x	✓	x	x	✓	x	x
6.13.15	resolve_local_ref2	x	✓	x	x	✓	x	x
6.13.16	resolve_param_list_ref	x	✓	x	x	✓	x	x
6.13.17	resolve_param_ref	x	✓	x	x	✓	x	x
6.13.18	resolve_param_ref2	x	✓	x	x	✓	x	x
6.13.19	resolve_record_ref	x	✓	x	x	✓	x	x
6.13.20	resolve_record_ref2	x	✓	x	x	✓	x	x
6.5.41	ret_double_value	x	✓	x	x	✓	x	x
6.5.42	ret_double_value2	x	✓	x	x	✓	x	x
6.5.43	ret_float_value	x	✓	x	x	✓	x	x
6.5.44	ret_float_value2	x	✓	x	x	✓	x	x
6.5.45	ret_signed_value	x	✓	x	x	✓	x	x
6.5.46	ret_signed_value2	x	✓	x	x	✓	x	x
6.5.47	ret_unsigned_value	x	✓	x	x	✓	x	x
6.5.48	ret_unsigned_value2	x	✓	x	x	✓	x	x
6.3.31	retry_on_all_comm_errors	x	✓	x	x	✓	x	x
6.3.32	RETRY_ON_ALL_COMM_STATUS	x	x	x	✓	x	x	x
6.3.33	RETRY_ON_ALL_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.34	RETRY_ON_ALL_RESPONSE_CODES	x	x	x	✓	x	x	x
6.3.35	retry_on_all_response_codes	x	✓	x	x	✓	x	x
6.3.36	RETRY_ON_COMM_ERROR	x	x	x	✓	x	x	x
6.3.37	retry_on_comm_error	x	✓	x	x	✓	x	x
6.3.38	RETRY_ON_COMM_STATUS	x	x	x	✓	x	x	x
6.3.39	RETRY_ON_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.40	RETRY_ON_NO_DEVICE (deprecated)	d	x	d	d	x	d	d
6.3.41	RETRY_ON_RESPONSE_CODE	x	x	x	✓	x	x	x
6.3.42	retry_on_response_code	x	✓	x	x	✓	x	x
6.11.30	round	✓	✓	✓	✓	✓	✓	✓
6.6.2	save_on_exit	✓	✓	✓	x	✓	✓	✓
6.6.3	save_values	✓	x	✓	✓	x	✓	✓
6.9.19	seconds_to_TIME_VALUE	✓	x	✓	✓	x	✓	✓
6.9.20	seconds_to_TIME_VALUE8	✓	✓	✓	x	✓	✓	✓
6.2.36	SELECT_FROM_LIST	✓	x	✓	✓	x	✓	✓
6.2.37	select_from_list	✓	x	✓	✓	x	✓	✓
6.2.38	select_from_menu	x	✓	x	x	✓	x	x
6.2.39	select_from_menu2	x	✓	x	x	✓	x	x

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.3.43	send	x	x	x	✓	x	x	x
6.3.44	send_all_values	x	✓	x	x	✓	x	x
6.3.45	send_command	x	x	x	✓	x	x	x
6.3.46	send_command_trans	x	x	x	✓	x	x	x
6.6.4	send_on_exit	✓	✓	✓	x	✓	✓	✓
6.3.47	send_trans	x	x	x	✓	x	x	x
6.3.48	send_value	x	✓	x	x	✓	x	x
6.3.49	send_value2	x	✓	x	x	✓	x	x
6.3.50	SET_NUMBER_OF_RETRIES	✓	x	✓	✓	x	✓	✓
6.4.20	sgetval	✓	x	✓	✓	x	✓	✓
6.11.31	ShortToByte	✓	x	✓	x	x	✓	✓
6.11.32	sin	✓	✓	✓	✓	✓	✓	✓
6.11.33	sinh	✓	✓	✓	✓	✓	✓	✓
6.11.34	sqrt	✓	✓	✓	✓	✓	✓	✓
6.4.21	ssetval	✓	x	✓	✓	x	✓	✓
6.8.19	strcmp	✓	✓	✓	✓	✓	✓	✓
6.8.18	strleft	✓	✓	✓	x	✓	✓	✓
6.8.20	strlen	✓	✓	✓	✓	✓	✓	✓
6.8.21	strlwr	✓	✓	✓	✓	✓	✓	✓
6.8.22	strmid	✓	✓	✓	✓	✓	✓	✓
6.8.23	strright	✓	✓	✓	x	✓	✓	✓
6.8.24	strstr	✓	✓	✓	✓	✓	✓	✓
6.8.25	strtrim	✓	✓	✓	✓	✓	✓	✓
6.8.26	strupr	✓	✓	✓	✓	✓	✓	✓
6.11.35	tan	✓	✓	✓	✓	✓	✓	✓
6.11.36	tanh	✓	✓	✓	✓	✓	✓	✓
6.9.21	Time_To_Date	✓	x	✓	✓	x	✓	✓
6.9.22	TIME_VALUE_to_Hour	✓	✓	✓	✓	✓	✓	✓
6.9.23	TIME_VALUE_to_Minute	✓	✓	✓	✓	✓	✓	✓
6.9.24	TIME_VALUE_to_Second	✓	✓	✓	✓	✓	✓	✓
6.9.25	TIME_VALUE_to_seconds	✓	✓	✓	✓	✓	✓	✓
6.9.26	TIME_VALUE_to_string	✓	✓	✓	✓	✓	✓	✓
6.9.27	timet_to_string	✓	✓	✓	✓	✓	✓	✓
6.9.28	timet_to_TIME_VALUE	✓	x	✓	x	x	✓	✓
6.9.29	timet_To_TIME_VALUE	x	x	x	✓	x	x	x
6.9.30	timet_to_TIME_VALUE8	✓	✓	✓	x	✓	✓	✓
6.9.31	To_Date	✓	x	✓	✓	x	✓	✓
6.9.32	To_Date_and_Time	✓	x	✓	x	x	✓	✓
6.9.33	To_Time	✓	x	✓	✓	x	✓	✓

Reference	Builtin	Communication profiles						
		CS	FF	GPE	HART	ISA100	PB	PN
6.9.34	To_TIME_VALUE	✓	x	✓	✓	x	✓	✓
6.9.35	To_TIME_VALUE8	✓	✓	✓	x	✓	✓	✓
6.8.27	trunc	✓	✓	✓	✓	✓	✓	✓
6.13.21	VARID	✓	x	✓	✓	x	✓	✓
6.5.49	vassign	✓	x	✓	✓	x	✓	✓
6.3.51	WriteCommand	✓	x	✓	x	x	✓	✓
6.14.7	writelnToDevice	x	x	x	✓	x	x	x
6.3.52	XMTR_ABORT_ON_ALL_COMM_STATUS	x	x	x	✓	x	x	x
6.3.53	XMTR_ABORT_ON_ALL_DATA	x	x	x	✓	x	x	x
6.3.54	XMTR_ABORT_ON_ALL_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.55	XMTR_ABORT_ON_ALL_RESPONSE_CODES	x	x	x	✓	x	x	x
6.3.56	XMTR_ABORT_ON_COMM_ERROR	x	x	x	✓	x	x	x
6.3.57	XMTR_ABORT_ON_COMM_STATUS	x	x	x	✓	x	x	x
6.3.58	XMTR_ABORT_ON_DATA	x	x	x	✓	x	x	x
6.3.59	XMTR_ABORT_ON_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.60	XMTR_ABORT_ON_NO_DEVICE (deprecated)	x	x	x	d	x	x	x
6.3.61	XMTR_ABORT_ON_RESPONSE_CODE	x	x	x	✓	x	x	x
6.3.62	XMTR_IGNORE_ALL_COMM_STATUS	x	x	x	✓	x	x	x
6.3.63	XMTR_IGNORE_ALL_DATA	x	x	x	✓	x	x	x
6.3.64	XMTR_IGNORE_ALL_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.65	XMTR_IGNORE_ALL_RESPONSE_CODES	x	x	x	✓	x	x	x
6.3.66	XMTR_IGNORE_COMM_ERROR	x	x	x	✓	x	x	x
6.3.67	XMTR_IGNORE_COMM_STATUS	x	x	x	✓	x	x	x
6.3.68	XMTR_IGNORE_DATA	x	x	x	✓	x	x	x
6.3.69	XMTR_IGNORE_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.70	XMTR_IGNORE_NO_DEVICE (deprecated)	x	x	x	d	x	x	x
6.3.71	XMTR_IGNORE_RESPONSE_CODE	x	x	x	✓	x	x	x
6.3.72	XMTR_RETRY_ON_ALL_COMM_STATUS	x	x	x	✓	x	x	x
6.3.73	XMTR_RETRY_ON_ALL_DATA	x	x	x	✓	x	x	x
6.3.74	XMTR_RETRY_ON_ALL_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.75	XMTR_RETRY_ON_ALL_RESPONSE_CODES	x	x	x	✓	x	x	x
6.3.76	XMTR_RETRY_ON_COMM_ERROR	x	x	x	✓	x	x	x
6.3.77	XMTR_RETRY_ON_COMM_STATUS	x	x	x	✓	x	x	x
6.3.78	XMTR_RETRY_ON_DATA	x	x	x	✓	x	x	x
6.3.79	XMTR_RETRY_ON_DEVICE_STATUS	x	x	x	✓	x	x	x
6.3.80	XMTR_RETRY_ON_NO_DEVICE (deprecated)	x	x	x	d	x	x	x
6.3.81	XMTR_RETRY_ON_RESPONSE_CODE	x	x	x	✓	x	x	x

There are the following special categories of builtins:

- User interface builtins;
- Communication builtins;
- Action builtins.

Table 6 shows restrictions to the usage of builtins, depending on their category.

Table 6 – Usage of builtins

Builtin usage	User interface builtins	Communication builtins	Action builtins
Methods referenced in user interface menus	Allowed	Allowed	Not allowed
PRE_EDIT_ACTIONS	Allowed	Not recommended	Allowed ^a
POST_EDIT_ACTIONS	Allowed	Not recommended	Allowed ^a
REFRESH_ACTIONS	Not recommended	GRAPH and CHART: allowed Others: not recommended	Allowed ^a
PRE_READ_ACTIONS	Not recommended	Allowed	Allowed ^a
POST_READ_ACTIONS	Not recommended	Allowed	Allowed ^a
PRE_WRITE_ACTIONS	Not recommended	Allowed	Allowed ^a
POST_WRITE_ACTIONS	Not recommended	Allowed	Allowed ^a
INIT_ACTIONS	Not recommended	Allowed	Not allowed
EXIT_ACTIONS	Not recommended	Allowed	Not allowed
^a Shall only be used for actions called from a VARIABLE.			

5.2 User interface builtins

User interface builtins display information and in addition some of them require interaction from the user. The list of user interface builtins are listed in Table 7.

Table 7 – User interface builtins

Builtin name
ACKNOWLEDGE
acknowledge
DELAY
delay
delayfor
delayfor2
DISPLAY
display
display_bitenum
display_builtin_error
display_comm_error
display_comm_status
display_device_status
display_dynamics
display_dynamics2

Builtin name
display_message
display_message2
display_response_code
display_response_status
display_xmtr_status
edit_device_value
edit_device_value2
edit_local_value
edit_local_value2
get_acknowledgement2
get_acknowledgement
GET_DEV_VAR_VALUE
get_dev_var_value
GET_LOCAL_VAR_VALUE
get_local_var_value
Menu
MenuDisplay (version A)
MenuDisplay (version B)
PUT_MESSAGE
put_message
SELECT_FROM_LIST
select_from_list
select_from_menu
select_from_menu2

5.3 Communication builtins

The communication builtins allow methods to communicate with the device or to control the communication with the device. The list of communication builtins is listed in Table 8.

Table 8 – Communication builtins

Builtin name
abort_on_all_comm_errors
ABORT_ON_ALL_COMM_STATUS
ABORT_ON_ALL_DEVICE_STATUS
ABORT_ON_ALL_RESPONSE_CODES
abort_on_all_response_codes
ABORT_ON_COMM_ERROR
abort_on_comm_error
ABORT_ON_COMM_STATUS
ABORT_ON_DEVICE_STATUS
ABORT_ON_NO_DEVICE
ABORT_ON_RESPONSE_CODE

Builtin name
abort_on_response_code
ext_send_command
ext_send_command_trans
fail_on_all_comm_errors
fail_on_all_response_codes
fail_on_comm_error
fail_on_response_code
get_more_status
IGNORE_ALL_COMM_STATUS
IGNORE_ALL_DEVICE_STATUS
IGNORE_ALL_RESPONSE_CODES
IGNORE_COMM_ERROR
IGNORE_COMM_STATUS
IGNORE_DEVICE_STATUS
IGNORE_NO_DEVICE
IGNORE_RESPONSE_CODE
read_value
read_value2
ReadCommand
retry_on_all_comm_errors
RETRY_ON_ALL_COMM_STATUS
RETRY_ON_ALL_DEVICE_STATUS
RETRY_ON_ALL_RESPONSE_CODES
retry_on_all_response_codes
RETRY_ON_COMM_ERROR
retry_on_comm_error
RETRY_ON_COMM_STATUS
RETRY_ON_DEVICE_STATUS
RETRY_ON_NO_DEVICE
RETRY_ON_RESPONSE_CODE
retry_on_response_code
send
send_all_values
send_command
send_command_trans
send_trans
send_value
send_value2
SET_NUMBER_OF_RETRIES
WriteCommand
XMTR_ABORT_ON_ALL_COMM_STATUS
XMTR_ABORT_ON_ALL_DATA
XMTR_ABORT_ON_ALL_DEVICE_STATUS

Builtin name
XMTR_ABORT_ON_ALL_RESPONSE_CODES
XMTR_ABORT_ON_COMM_ERROR
XMTR_ABORT_ON_COMM_STATUS
XMTR_ABORT_ON_DATA
XMTR_ABORT_ON_DEVICE_STATUS
XMTR_ABORT_ON_NO_DEVICE
XMTR_ABORT_ON_RESPONSE_CODE
XMTR_IGNORE_ALL_COMM_STATUS
XMTR_IGNORE_ALL_DATA
XMTR_IGNORE_ALL_DEVICE_STATUS
XMTR_IGNORE_ALL_RESPONSE_CODES
XMTR_IGNORE_COMM_ERROR
XMTR_IGNORE_COMM_STATUS
XMTR_IGNORE_DATA
XMTR_IGNORE_DEVICE_STATUS
XMTR_IGNORE_NO_DEVICE
XMTR_IGNORE_RESPONSE_CODE
XMTR_RETRY_ON_ALL_COMM_STATUS
XMTR_RETRY_ON_ALL_DATA
XMTR_RETRY_ON_ALL_DEVICE_STATUS
XMTR_RETRY_ON_ALL_RESPONSE_CODE
XMTR_RETRY_ON_ALL_RESPONSE_CODES
XMTR_RETRY_ON_COMM_ERROR
XMTR_RETRY_ON_COMM_STATUS
XMTR_RETRY_ON_DATA
XMTR_RETRY_ON_DEVICE_STATUS
XMTR_RETRY_ON_NO_DEVICE
XMTR_RETRY_ON_RESPONSE_CODE

5.4 Action builtins

The action builtins shall only be used for actions called from a VARIABLE. They allow for accessing the value of the variable from which the action was called. The list of action builtins is listed in Table 9.

Table 9 – Action builtins

Reference	Builtin name	Allowed data type for action variable	Allowed actions from which the builtin can be called
6.4.1	fgetval	FLOAT, DOUBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS POST_RQSTUPDATE_ACTIONS POST_USERCHANGE_ACTIONS
6.4.2	fsetval	FLOAT, DOUBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS POST_RQSTUPDATE_ACTIONS POST_USERCHANGE_ACTIONS
6.4.3	get_date	DATE_AND_TIME, TIME_VALUE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.4	get_double	DOUBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.5	get_float	FLOAT	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.6	get_signed	INTEGER	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.7	get_string	ASCII, BITSTRING, BIT_ENUMERATED, ENUMERATED, EUC, INDEX, PASSWORD, VISIBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.8	get_unsigned	BIT_ENUMERATED, BOOLEAN, ENUMERATED, INDEX, UNSIGNED_INTEGER	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS

Reference	Builtin name	Allowed data type for action variable	Allowed actions from which the builtin can be called
6.4.9	igetval	BIT_ENUMERATED, BOOLEAN, ENUMERATED, INDEX, INTEGER, UNSIGNED_INTEGER, TIME_VALUE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS POST_RQSTUPDATE_ACTIONS POST_USERCHANGE_ACTIONS
6.4.10	isetval	BIT_ENUMERATED, BOOLEAN, ENUMERATED, INDEX, INTEGER, UNSIGNED_INTEGER, TIME_VALUE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS POST_RQSTUPDATE_ACTIONS POST_USERCHANGE_ACTIONS
6.4.12	igetval	BIT_ENUMERATED, BOOLEAN, ENUMERATED, INDEX, INTEGER, UNSIGNED_INTEGER, TIME_VALUE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS
6.4.13	isetval	BIT_ENUMERATED, BOOLEAN, ENUMERATED, INDEX, INTEGER, UNSIGNED_INTEGER, TIME_VALUE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS
6.4.14	put_date	DATE_AND_TIME, TIME_VALUE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.15	put_double	DOUBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS

Reference	Builtin name	Allowed data type for action variable	Allowed actions from which the builtin can be called
6.4.16	put_float	FLOAT	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.17	put_signed	INTEGER	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.18	put_string	ASCII, BITSTRING, BIT_ENUMERATED, ENUMERATED, EUC, INDEX, PASSWORD, VISIBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.19	put_unsigned	BIT_ENUMERATED, BOOLEAN, ENUMERATED, INDEX, UNSIGNED_INTEGER	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS
6.4.20	sgetval	ASCII, BITSTRING, EUC, PACKED_ASCII, PASSWORD, VISIBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS POST_RQSTUPDATE_ACTIONS POST_USERCHANGE_ACTIONS
6.4.21	ssetval	ASCII, BITSTRING, EUC, PACKED_ASCII, PASSWORD, VISIBLE	PRE_EDIT_ACTIONS POST_EDIT_ACTIONS PRE_READ_ACTIONS POST_READ_ACTIONS PRE_WRITE_ACTIONS POST_WRITE_ACTIONS REFRESH_ACTIONS POST_RQSTUPDATE_ACTIONS POST_USERCHANGE_ACTIONS

6 EDDL builtin library

6.1 General

Clause 6 describes builtin subroutines, which can be used in an EDD. The builtins support user or device interactions. Typical user interactions are, for example, user inputs of VARIABLE values or acknowledgments. Typical device interactions are, for example, send commands to the device or interpreting response codes and status bytes. This Clause 6 contains the library of builtins.

No other builtin functions are available and a compliant EDD shall only use the library functions defined in this specification. Requirements that all conforming EDD applications shall meet are defined in this specification. A conformant EDD application shall support all library functions found in this specifications that are applicable to the respective communication profile(s). A EDD application shall support all compliant EDDs without requiring any modifications to the compliant EDD and without any additional information beyond that found in the compliant EDD.

6.2 User interface builtins

6.2.1 Builtin ACKNOWLEDGE

Communication profiles: CS, GPE, HART, PB, PN

Purpose: Display message and wait for key press

The builtin ACKNOWLEDGE will display the prompt and wait for acknowledgment from the user.

The user has the opportunity to abort the method while method execution is halted waiting for this function to return.

NOTE 1 Previously displayed messages are not guaranteed to remain on the screen.

NOTE 2 Some EDD applications will clear the display upon each call to this function.

Lexical structure

ACKNOWLEDGE(prompt)

The lexical elements of the builtin ACKNOWLEDGE are shown in Table 10.

Table 10 – Builtin ACKNOWLEDGE

Parameter name	Type	Direction	Usage	Description
prompt	DD_STRING	I	m	Message to be displayed (see 6.2.40 for syntax). String types that are not a DD_STRING may be also be used and will be automatically promoted to a DD_STRING (see 6.18.3).
<return>	int	O	m	One of the return codes specified in Table 377.

6.2.2 Builtin acknowledge

Communication profiles: CS, GPE, HART, PB, PN

Purpose: Display message and wait for key press

The builtin acknowledge will display the prompt and wait for acknowledgment from the user.

NOTE 1 New Device Descriptions can use the builtin ACKNOWLEDGE, which supports direct referencing of VARIABLES. The lower case version of this function is maintained for backwards compatibility only.

The user has the opportunity to abort the method while method execution is halted waiting for this function to return.

NOTE 2 Previously displayed messages are not guaranteed to remain on the screen.

NOTE 3 Some EDD applications will clear the display upon each call to this function.

Lexical structure

acknowledge(prompt, global_var_ids)

The lexical elements of the builtin acknowledge are shown in Table 11.

Table 11 – Builtin acknowledge

Parameter name	Type	Direction	Usage	Description
prompt	DD_STRING	I	m	Message to be displayed (see 6.2.35 for syntax). String types that are not a DD_STRING may be also be used and will be automatically promoted to a DD_STRING (see 6.18.3).
global_var_ids	Int[]	I	m	An array that contains unique identifiers of VARIABLE instances.
<return>	int	O	m	One of the return codes specified in Table 377.

6.2.3 Builtin DELAY

Communication profiles: CS, GPE, HART, PB, PN

Purpose: Display a message and delay

The builtin DELAY will display a prompt message and pause for the specified number of seconds. The delay time shall be a positive number.

The user has the opportunity to abort the method while method execution is halted waiting for this function to return.

NOTE 1 Previously displayed messages are not guaranteed to remain on the screen.

NOTE 2 Some EDD applications will clear the display upon each call to this function.

Lexical structure

DELAY(delay_time, prompt)

The lexical elements of the builtin DELAY are shown in Table 12.

Table 12 – Builtin DELAY

Parameter name	Type	Direction	Usage	Description
delay_time	unsigned integer	I	m	The delay in number of seconds.
prompt	DD_STRING	I	m	A message to be displayed (see 6.2.40 for syntax). String types that are not a DD_STRING may be also be used and will be automatically promoted to a DD_STRING (see 6.18.3).
<return>	VOID	—	—	—

6.2.4 Builtin delay

Communication profiles: CS, GPE, HART, PB, PN

Purpose: Display a message and delay

The builtin delay will display a prompt message, and pause for the specified number of seconds. The delay time shall be a positive number.

The user has the opportunity to abort the method while method execution is halted waiting for this function to return.

E DIN EN IEC 61804-5:2021-03
EN IEC 61804-5:2020

NOTE 1 Previously displayed messages are not guaranteed to remain on the screen.

NOTE 2 Some EDD applications will clear the display upon each call to this function.

NOTE 3 New Device Descriptions can use DELAY, which supports direct referencing of VARIABLES. The lower case version of this function is maintained for backwards compatibility only.

Lexical structure

`delay(delay_time, prompt, global_var_ids)`

The lexical elements of the builtin delay are shown in Table 13.

Table 13 – Builtin delay

Parameter name	Type	Direction	Usage	Description
delay_time	unsigned integer	I	m	The delay in number of seconds.
prompt	DD_STRING	I	m	A message to be displayed (see 6.2.35 for syntax). String types that are not a DD_STRING may be also be used and will be automatically promoted to a DD_STRING (see 6.18.3).
global_var_ids	Int[]	I	m	An array which contains unique identifiers of VARIABLE instances.
<return>	VOID	—	—	—

6.2.5 Builtin delayfor

Communication profiles: FF, ISA100

Purpose: Display a message and wait the specified time

The builtin delayfor prints a string on the application display and waits for the allotted amount of time to expire.

NOTE 1 Previously displayed messages are not guaranteed to remain on the screen.

NOTE 2 Some EDD applications will clear the display upon each call to this function.

Lexical structure

`delayfor(duration, prompt, ids, indices, id_count)`

The lexical elements of the builtin delayfor are shown in Table 14.

Table 14 – Builtin delayfor

Parameter name	Type	Direction	Usage	Description
duration	long	I	m	The delay in number of seconds. This shall be a positive number.
prompt	DD_STRING	I	m	A message to be displayed (see 6.2.6 for syntax). String types that are not a DD_STRING may be also be used and will be automatically promoted to a DD_STRING (see 6.18.3).
ids	unsigned long[]	I	m	The item ids of a VARIABLE instance used in the prompt message.
indices	unsigned long[]	I	m	The member ids or array indices of the VARIABLES used in the prompt message. Set to zero if there is no member id or array index.