BRITISH STANDARD

BS EN
61915-1:2008

Low-voltage switchgear and controlgear — Device profiles for networked industrial devices —

Part 1: General rules for the development of device profiles

ICS 29.130.20

BSI
British Standards

National foreword

This British Standard is the UK implementation of EN 61915-1:2008. It is identical to IEC 61915-1:2007.

The UK participation in its preparation was entrusted by Technical Committee PEL/17, Switchgear, controlgear, and HV-LV co-ordination, to Subcommittee PEL/17/2, Low voltage switchgear and controlgear.

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

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

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2008

ISBN 978 0 580 56289 1

Amendments/corrigenda issued since publication

Date	Comments

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 61915-1

January 2008

ICS 29.130.20

English version

Low-voltage switchgear and controlgear - Device profiles for networked industrial devices - Part 1: General rules for the development of device profiles

(IEC 61915-1:2007)

Appareillage à basse tension - Profils d'appareil pour les appareils industriels mis en réseau - Partie 1: Règles générales pour le développement de profils d'appareil (CEI 61915-1:2007)

Niederspannungsschaltgeräte - Geräteprofile für vernetzte industrielle Schaltgeräte - Teil 1: Allgemeine Festlegungen für die Entwicklung von Geräteprofilen (IEC 61915-1:2007)

This European Standard was approved by CENELEC on 2007-12-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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

CENELEC

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

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Ref. No. EN 61915-1:2008 E

Foreword

The text of document 17B/1575/FDIS, future edition 1 of IEC 61915-1, prepared by SC 17B, Low-voltage switchgear and controlgear, of IEC TC 17, Switchgear and controlgear, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 61915-1 on 2007-12-01.

The following dates were fixed:

– latest date by which the EN has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2008-09-01
– latest date by which the national standards conflicting with the EN have to be withdrawn	(dow)	2010-12-01

Annex ZA has been added by CENELEC.

_____

Endorsement notice

The text of the International Standard IEC 61915-1:2007 was approved by CENELEC as a European Standard without any modification.

_____

CONTENTS

INTRODUCTION

The purpose of this International Standard is to provide a framework within which IEC product committees can define profiles for devices within their scope.

NOTE This framework follows the principles given in IEC/TR 62390, the “Common automation device – Profile guideline”, and refers to ISO 15745, “Industrial automation systems and integration – Open systems application integration framework”.

Profiles define a common set of functionality (data and behaviour) for a class of devices in a given industrial domain, thus allowing system designers, system integrators and maintenance staff to handle profile-based devices without special tool configuration. Profiles also provide consistent structuring and semantics of device functionality.

This part of IEC 61915 (Part 1) defines general rules for the development of device profiles for networked industrial devices, including recommendations of general interest and application, for example a documentation template and a profile exchange language. This will allow uniformity of profile structure throughout the different device types.

IEC product committees may define “root device profiles” for their devices, in which they will specify the amount of information which their products should make available through any network, using the general rules defined in this part of IEC 61915. This will facilitate uniformity of profiles throughout the corresponding family of devices. These root device profiles will be published in subsequent parts of the IEC 61915 series.

This International Standard also gives manufacturers or other organizations a common framework to represent their network capable devices.

Manufacturers or other organizations may use the root device profiles specified by the IEC product committees for various device types as a basis for developing device profiles corresponding to their products, using the general rules defined in this part of IEC 61915 to add the required manufacturer-specific extensions. Alternatively, they may develop their own device profiles using only the general rules. These manufacturer’s device profiles will typically be published within the product documentation.

This International Standard facilitates the writing of network independent application software.

LOW-VOLTAGE SWITCHGEAR AND CONTROLGEAR – DEVICE PROFILES FOR NETWORKED INDUSTRIAL DEVICES –

Part 1: General rules for the development of device profiles

1 Scope

The IEC 61915 series is intended to improve interoperability of devices, network tools and application software.

This part of IEC 61915 defines a framework for common representation of networked industrial devices and provides a template for documenting such a representation, independent of the network used. This framework follows the principles given in IEC/TR 62390, the “Common automation device – Profile guideline”, and refers to ISO 15745, “Industrial automation systems and integration – Open systems application integration framework”.

NOTE 1 The device profile format specified in this part of IEC 61915 is compatible with devices connected to both bit- and byte-oriented networks.

This part of IEC 61915 applies to root device profiles, generic device profiles, and specific device profiles. The root device profiles will be published in subsequent parts of the IEC 61915 series.

NOTE 2 This International Standard is specifically intended for products covered by the IEC 60947 series.

NOTE 3 Organisations such as consortia are encouraged to use the rules defined in this part of IEC 61915 to develop generic device profiles for use within their own organisations.

Users (product manufacturers and other organizations) should use the root device profiles together with the rules defined in this part of IEC 61915. This part of IEC 61915 allows users to make extensions to the root device profiles and/or generic device profiles. Where no suitable root device profile exists, the user may develop generic or specific device profiles using the rules defined in this part of IEC 61915.

This part of IEC 61915 recommends the use of a profile exchange language for representation of the device profile information in order to facilitate the profile’s use by network tools and application software.

NOTE 4 The types of devices may vary from simple devices, such as pilot lights, push-buttons and limit switches, to more complex devices with many bytes of information, such as motor controllers, semiconductor motor starters, etc.

2 Normative references

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

IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*

IEC 61131-3:2003, *Programmable controllers – Part 3: Programming languages*

IEC/TR 62390:2005, *Common automation device – Profile guideline*

ISO 1000:1992*, SI units and recommendations for the use of their multiples and of certain other units*
Amendment 1 (1998)

ISO 15745 (all parts)*, Industrial automation systems and integration – Open systems application integration framework*

ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2*

3 Definitions, abbreviations and symbols

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
device profile
representation of a device that describes the device's data and behaviour as viewed through a network, independent from any network technology

NOTE Description of the communication options to be used to transfer data using a given network technology is outside the scope of the device profile.

[IEC/TR 62390, Definition 3.1.9, modified]

3.1.2
functional element
entity of software or software combined with hardware, capable of accomplishing a specified function of a device

NOTE 1 A functional element has interface(s), associations to other functional elements and functions.

NOTE 2 A functional element can be made out of function block(s), object(s) or parameter list(s).

[IEC/TR 62390, Definition 3.1.12]

3.1.3
generic device profile
manufacturer's device profile for a family of similar devices (e.g. similar device types with differing feature levels)

3.1.4
manufacturer's device profile
device profile, defined by a manufacturer or any other organization, containing the mandatory elements and the selected optional elements of a root device profile, if such a root device profile is applicable, and which may also include manufacturer-specific extension(s)

NOTE 1 A manufacturer's device profile is either a generic device profile, or a specific device profile.

NOTE 2 Organizations include users' organizations, consortia, institutions, or standards bodies.

3.1.5
manufacturer-specific extension
information contained within a manufacturer's device profile which is specified by a particular manufacturer or other organization and is in addition to the mandatory and optional parts of the root profile

3.1.6
parameter
data element that represents device information that can be read from or written to a device, e.g. through the network or a local HMI

NOTE A parameter is typically characterized by a parameter name, data type and access direction.

[IEC/TR 62390, Definition 3.1.22]

3.1.7
parameter assembly
collection of one or more parameters that can be read from or written to a device

NOTE Assemblies are typically used to increase efficiency of data exchanges.

3.1.8
parameter group
logical collection of parameters, typically associated with the same operational purpose or functional element in a device

NOTE 1 Parameter groups may be nested, i.e. it is possible to define a parameter group composed of other parameter groups.

NOTE 2 Contrary to parameter assemblies, parameter groups are not defined to increase efficiency of data exchanges. Instead, they are mainly defined for the purpose of organizing long lists of parameters into meaningful sets (e.g. for HMIs).

3.1.9
root device profile
device profile, defined by an IEC product committee, comprising mandatory and optional elements

NOTE Mandatory and optional elements include parameters, parameter groups, ..., as well as individual characteristics of these.

3.1.10
service
means for a user or an application to request execution of specific actions (e.g. fault reset, calibrate, identify, diagnostics)

NOTE 1 The service may be provided by the device or one of its functional elements.

NOTE 2 Actual execution may require that related preliminary conditions are satisfied.

NOTE 3 Services are futher detailed in 5.9.

3.1.11
specific device profile
manufacturer's device profile for a single device (e.g. a specific catalogue model)

NOTE A specific device profile is also commonly referred to as a device description.

3.2 Abbreviations and symbols

A	Applied
D	Device-specific
FE	Functional element
ID	Identifier
m	Mandatory (if defined in a generic device profile)
M	Mandatory (if defined in a root device profile)
O	Optional
R	Read
RW	Read/write
UML	Unified modelling language
W	Write
XML	Extensible markup language
na	Not applicable
r	Reserved

4 Device profiles

4.1 General

A device profile consists of the data (parameters, parameter assemblies and parameter groups) and behaviour (functional elements, state model and services) provided by the device. This device profile is used to represent the device independently of the network, e.g. when designing an industrial automation application.

A device profile shall define the format and content of any control and management information (see Annex E) that is received and/or sent by the device. Annex A defines the template for the device profile. The entire template is used as a basis for both root device profiles and manufacturer's device profiles. Unless otherwise instructed in this part of IEC 61915, unused fields shall remain empty.

NOTE 1 If some main template sections are completely empty (e.g. the manufacturer's header for a root device profile), these sections may be omitted in the profile.

Each profile shall stand alone without reference to other profiles, i.e. profiles shall not contain other profiles embedded within them (see Annex C for profile creation guidelines). Simpler device profiles should be subsets of the parameter lists, parameter assemblies, parameter groups, state models and services of more complex device profiles, rather than redefining this information.

Values of the parameters defined by the specific device profile will be transmitted on the network. The application uses the profile information to interpret the parameter values exchanged with the device.

NOTE 2 A device profile exists either on paper or in an electronic format.

NOTE 3 A device may store parts or all of the profile information; in this case, this information may also be read through the network from the device. Format of these exchanges is not covered by this standard.

Parameter assemblies and parameter groups shall only include parameters that are defined in the device profile.

Parameter names and device state names shall use the terminology utilised in the corresponding product standards.

NOTE 4 Annex D gives a recommended syntax for the documentation and transfer of device profiles when using XML.

4.2 Root device profile

A root device profile is created by the relevant IEC product committee for each device type (see Note 3 of Clause 1 for use by other organizations).

When defining root device profiles, IEC product committees shall apply the following rules, unless there is a substantial technical justification.

a) The same parameters shall be used for the root device profiles of all the devices within a product family.

b) The meaning assigned to the value of each parameter shall be the same throughout the family, e.g. for a start/stop bit (Boolean) parameter, the value 1 should always mean start.

c) Similarly, for assemblies the bit and byte order shall be consistent with assemblies in other root device profiles belonging to the same product family, e.g. in a motor starter control assembly, the start bit should be in the same position for each type of motor starter.

A root device profile shall specify which parts of the profile (e.g. parameters) are mandatory, i.e. required for all devices claiming compliance with this specific root device profile, and which parts are optional, i.e. need not be used by all devices using this specific root device profile.

The root device profile shall not include information which is network-specific.

Two practical examples of root device profiles are given in Clause B.2. Figure B.1 provides an example for a photoelectric switch and Figure B.2 provides an example for a motor starter.

EXAMPLE 1 The photoelectric switch root device profile is an example of a presence sensor device that can be configured over the network to detect the presence of an object either by the presence of light or the absence of light, and to transmit a value of 1 over the network for the Presence parameter, indicating the object's presence. The device can also be put in either configuration or automatic mode and normal or test states by sending the device appropriate parameters values over the network. The mandatory requirement of the Device and Operate mode parameters gives the device description “Photoelectric switch with mode control”. A manufacturer’s device profile could use this root profile to create a device with a profile that only includes the parameters Presence, Device mode and Operate mode. The device manufacturer’s description could be the same as the root profile. Or the manufacturer could make a device that adds the Alarm and Test parameters and describe the device as “Mode control photoelectric switch with alarm and test”.

EXAMPLE 2 The motor starter root device profile is an example of a motor controller device root profile that would allow a manufacturer’s profile based on it to represent either an electro-mechanical, solid state or softstart starter. A particular motor controller device may provide additional information over the network, such as motor current value. Its manufacturer could use the motor starter root device profile as a basis, and extend it by adding specific features such as a “Motor Full Load Current” parameter.

4.3 Manufacturer's device profile

4.3.1 General

Two main types of manufacturer’s device profiles may be defined:

– a generic device profile for a family of similar devices (e.g. similar device types with differing feature levels),

– a specific device profile for a single device (e.g. a specific catalogue model).

4.3.2 Manufacturer's device profile created using a root device profile

The manufacturer's device profile shall include all of the mandatory parts of the root device profile without alteration. Each element specified as optional by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each element specified as optional by the root device profile shall not be included in a specific device profile unless the corresponding feature is actually implemented in the device.

In addition, the root device profile may be extended using the rules given in Clause 6, by

- defining additional parameters;
- defining additional complex data types (if additional parameters need these complex data types);
- defining additional parameter assemblies;
- defining additional parameter groups;
- defining additional functional elements;
- defining sub-states of states specified in the root device profile state model;
- defining states concurrent to those specified in the root device profile state model;
- defining additional services.

Names of elements of the root device profile, whether mandatory or optional, shall not be reused for any of these additional elements.

NOTE Addition of a new parameter, for example, may be needed if none of the root device profile parameters is suitable for the manufacturer's device, or if any of the characteristics of a root device profile parameter need to be changed (e.g. data type, value range).

EXAMPLE Figure B.3 provides an example of a generic device profile using a root device profile. This generic device profile extends the photoelectric switch root profile provided in Figure B.1. The profile makes the root profile parameters Alarm and Test mandatory and adds the mandatory manufacturer-specific parameters Output mode, On delay, Off delay, One shot delay and Sensitivity for the device.

4.3.3 Manufacturer's device profile created without using a root device profile

If the procedure in 4.3.2 does not apply because a suitable root device profile does not exist, then a manufacturer's device profile may be created using the rules given in Clause 7.

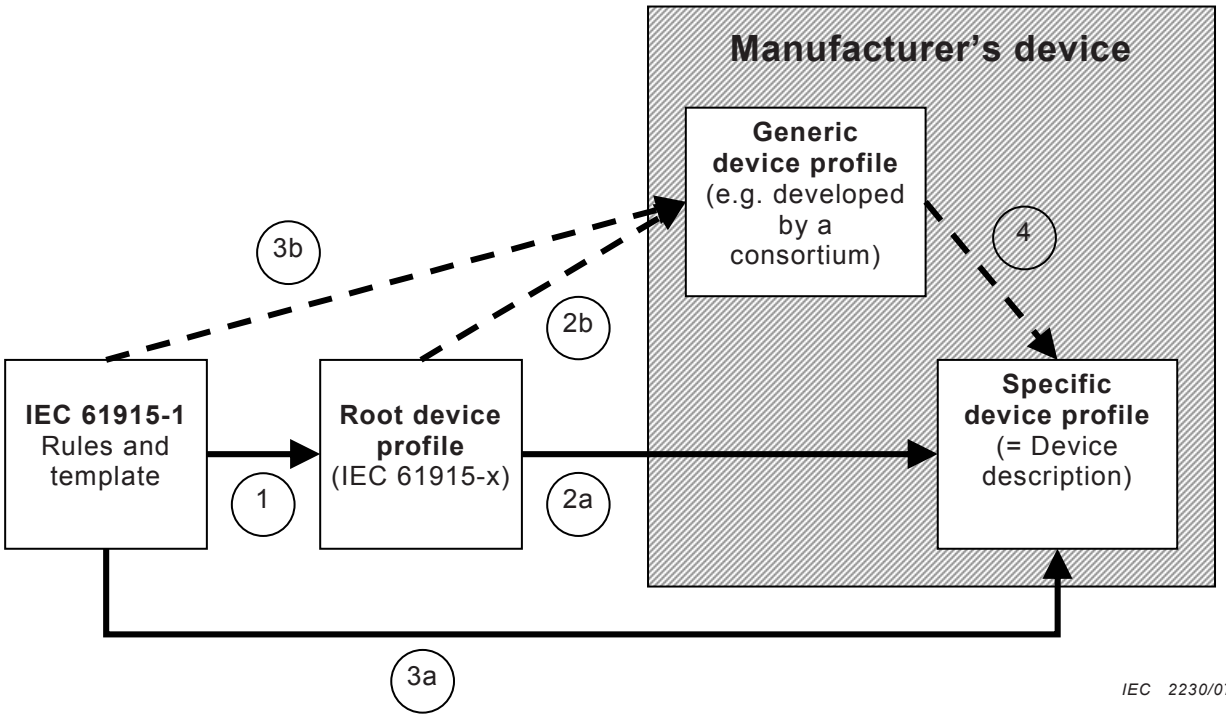
NOTE 1 A suitable root device profile may not exist for a product, for example if the product has been designed before a corresponding root device profile was published, or if the product includes new technology or new features.

NOTE 2 A manufacturer may also create a specific device profile based on a generic device profile (e.g. a generic device profile defined by a users' organization).

EXAMPLE Figure B.4 provides an example of a specific device profile that does not use a root profile. The example is for a photoelectric switch with learning and target sensitivity. Since the learning feature of the device is not accessible via the network, no parameters are provided in the profile to support the feature. The device is put in the Learning state when a device-adjusting button is pushed and transitions back to the Automatic state when the device-adjusting button is released.

4.4 Profile relationships

Figure 1 shows the relationship between this part of IEC 61915, the root device profiles in the subsequent parts of the IEC 61915 series, and the manufacturer's device profiles (generic device profiles or specific device profiles).



Key

1 Product committee creates a root device profile using the template (see Clause 5)

2 Manufacturer or organization creates a manufacturer's device profile using a root device profile (2a specific device profile, 2b generic device profile) (see Clause 6)

3 When no suitable root device profile exists, a manufacturer or organization creates a manufacturer's device profile using the template and rules given in Clause 7 (3a specific device profile, 3b generic device profile)

4 Manufacturer creates a specific device profile using a generic device profile

Figure 1 – Relationship between IEC 61915-1 and device profiles

5 Creating a root device profile using the device profile template

5.1 General

The IEC product committee shall complete the relevant parts of the following sections of the device profile template (see Annex A):

– root device profile header – see 5.2;

– parameters (root device profile) – see 5.3;

– complex data types (root device profile) (if parameters need these complex data types) – see 5.4;

– parameter assemblies (root device profile) – see 5.5;

– parameter groups (root device profile) – see 5.6;

– functional elements (root device profile) – see 5.7;

– state model (root device profile) – see 5.8;

– services (root device profile) – see 5.9.

When an IEC product committee has completed the relevant parts of the device profile template with the device information, it has created a root device profile.

5.2 Root device profile header

5.2.1 General

The root device profile header shall contain the root device profile ID, root device profile version, root device profile release date and device description.

5.2.2 Root device profile ID

A root device profile ID shall be assigned for each root device profile. The format for a root device profile ID shall be a text string using the format P(SB SN)PN, where

P is always the upper case letter P;

SB is a text string that identifies the standards body followed by a space;

SN is a text string that identifies the standard body's document that contains the root device profile;

PN is a five-digit integer (00001…99999) allocated by the IEC product committee which is unique to the specific combination of SB and SN.

NOTE The text strings for SB and SN may include multiple dashes or other punctuation.

EXAMPLE P(IEC 60947-5-2)10042.

5.2.3 Root device profile version

A root device profile version shall be assigned for each root device profile. Version numbers shall be used to record changes or modifications to a root device profile. Version number changes shall occur when any one or more of the following change:

– parameters;
– complex data types;
– parameter assemblies;
– parameter groups;
– functional elements;
– the state model;
– services.

The initial release of a root device profile shall be version 001. All profiles with a version number of 000 shall be considered unreleased.

The format for a root device profile version shall be a text string using the format VAAA, where

V is always the upper case letter V;

AAA is the version number.

5.2.4 Root device profile release date

Each root device profile shall contain a version release date. The release date of the root device profile shall be a text string using the format YYYY-MM-DD (a 10-character string including the dashes), where

YYYY is the year;

MM is the month of the year (01-12);

DD is the day of the month (01-31).

5.2.5 Device description

The device description is a text string which describes the type of device and which is specified by the IEC product committee.

5.3 Parameters (root device profile)

5.3.1 General

A root device profile shall contain one or more parameters. The information required by 5.3.2 to 5.3.8 shall be given for each parameter.

Examples of parameter categories are given in Annex E.

5.3.2 Parameter name (mandatory)

The “Parameter name” field shall contain a text string (maximum 32 characters) specified by the IEC product committee.

5.3.3 Data type (mandatory)

The “Data type” field shall contain a type name selected by the IEC product committee either from the valid simple data types listed in Table 1 or from the complex data types defined in the profile using the rules in 5.4.

NOTE If the use of derived data types is needed, it is strongly recommended to use the definitions and derived data types as specified in IEC 61131-3.

The data types STRING and UNICODE shall include their length, in bytes, in the type field.

EXAMPLE STRING10.

Table 1 – Valid simple data types

Type name	Description	Definition and range	Standard
BOOL	Bit or Boolean	Represented by a 0 or a 1	IEC 61131-3
BYTE	Byte	Bit string of 8 bits	IEC 61131-3
WORD	Word	Bit string of 16 bits	IEC 61131-3
DWORD	Double word	Bit string of 32 bits	IEC 61131-3
LWORD	Long word	Bit string of 64 bits	IEC 61131-3
SINT	Short integer	–128 to 127	IEC 61131-3
USINT	Unsigned short integer	0 to 255	IEC 61131-3
INT	Single integer	–32768 to 32767	IEC 61131-3
UINT	Unsigned integer	0 to 65535	IEC 61131-3
DINT	Double integer	-2^{31} to $2^{31}-1$	IEC 61131-3
UDINT	Unsigned double integer	0 to $2^{32}-1$	IEC 61131-3
LINT	Long integer	-2^{63} to $2^{63}-1$	IEC 61131-3
ULINT	Unsigned long integer	0 to $2^{64}-1$	IEC 61131-3
REAL	Single real	IEC 60559 basic single floating point. Allows an approximate range of $-1{,}2 \times 10^{-38}$ to $1{,}8 \times 10^{38}$	IEC 61131-3
LREAL	Double real	IEC 60559 basic double floating point. Allows an approximate range of $-1{,}2 \times 10^{-308}$ to $1{,}8 \times 10^{308}$	IEC 61131-3
STRING	Text string	1 byte per character	IEC 61131-3
UNICODE	Unicode	2 bytes per character	ISO/IEC 10646

5.3.4 Units (mandatory)

The “Units” field shall contain a text string that specifies the units of the parameter, using SI units as defined in ISO 1000 where applicable.

When no units are defined (e.g. for a counter) or required (e.g. for a data type of Boolean), the text string “na” shall be used.

5.3.5 Offset and multiplier (mandatory)

The “Offset” and the “Multiplier” fields shall together specify how the parameter value is interpreted, according to the following formula:

$$\text{Engineering value} = (\text{parameter value} + \text{offset}) \times \text{multiplier}$$

The offset and the multiplier shall be floating point numbers, without units. An offset and a multiplier shall always be specified.

For non-numeric data types, the “Offset” and “Multiplier” fields shall each contain the text string “na”.

For numeric data types, if no offset is required, then the value of “0” shall be used for the offset. If no scaling is required, the value of “1” shall be used for the multiplier.

EXAMPLE 1 A parameter of value 100 with engineering units in °C, an offset of 0 and a multiplier of 1 results in an engineering value of 100 °C.

EXAMPLE 2 A parameter of value 100 with engineering units in °C, an offset of 0 and a multiplier of 0,1 results in an engineering value of 10,0 °C.

EXAMPLE 3 A parameter of value 100 with engineering units in °C, an offset of 1 000 and a multiplier of 1 results in an engineering value of 1 100 °C.

EXAMPLE 4 A parameter of value 100 with engineering units in °C, an offset of 1 000 and a multiplier of 0,1 results in an engineering value of 110,0 °C.

5.3.6 Range (mandatory)

The “Range” field shall specify the limits of the range of numeric data values of the parameter before the modification by the offset and the multiplier. The range shall be specified as the minimum value followed by an ellipsis (…) followed by the maximum value with no spaces. The range shall be inclusive of the specified minimum and maximum values.

EXAMPLE 1 A parameter range of 40…200 with engineering units in °C, an offset of 1 000 and a multiplier of 1 results in an engineering value of 1 040 °C…1 200 °C. There are 161 parameter values within the range.

EXAMPLE 2 A parameter range of 40…200 with engineering units in °C, an offset of 1 000 and a multiplier of 0,1 results in an engineering value of 104,0 °C…120,0 °C. There are 161 parameter values within the range.

NOTE Both examples contain the same number of parameter values.

The parameter range may be more limited than that of the parameter type, e.g. the parameter type may be USINT (0…255), while the range may only be 40…200.

If meanings are assigned to particular values outside or inside the range, or to sets of values inside the range, they shall be defined in the description field of the parameter.

EXAMPLE 3 A parameter for motor current that can take the values from 100 to 200 (overload current), from 600 to 1 000 (inrush current) and the particular value 10 000 (ultimate short-circuit current) would result in a general range of 100…10 000, and specifics would be detailed in the description field.

When no range is required (e.g. for a data type of Boolean), the text string “na” shall be inserted in the “Range” field.

5.3.7 Access (mandatory)

The “Access” field shall specify the access to the parameter allowed through the network.

Access shall be specified as either

R for parameters readable from the connected device; or

RW for parameters both readable from and writable to the connected device.

Parameters shall not be specified as write access only.

5.3.8 Required (mandatory)

A root device profile shall specify for each parameter whether the device is required to implement it or not.

The root device profile's “Required” field shall contain either

M for mandatory parameters, i.e. those required to be implemented by the device; or

O for optional parameters.

5.3.9 Parameter description (optional)

The “Description” field shall, if used, contain a text description of the parameter and/or its use.

This field may also contain a description of any specific meaning of the parameter's values, formatted as the parameter value followed by an equal sign (=) followed by the parameter value meaning. There shall be no spaces immediately before or after the equal sign. The entire string shall be enclosed within quotation marks as shown in the following examples.

EXAMPLE 1 "0=no object sensed"
"1=object sensed"

EXAMPLE 2 "100…200=overload current"
"600…1 000=inrush current"
”10 000=ultimate short circuit current”

5.3.10 Recommended parameters for device identification

5.3.10.1 General

An IEC product committee may want to specify within a root device profile some parameters for device identification. To allow consistency between the root device profiles defined by various IEC product committees, it is recommended to use for this purpose parameters as defined in the following subclauses.

5.3.10.2 Root device profile ID

Identifies the root device profile on which this manufacturer’s device profile is based (see 5.2.2). Recommended data type is STRING24.

NOTE If a root device profile is not being used, then the rules given in Clause 7 should be followed.

5.3.10.3 Root device profile version

Identifies the version of the root device profile on which this manufacturer’s device profile is based (see 5.2.3). Recommended data type is STRING4.

5.3.10.4 Manufacturer ID

Identifies the manufacturer of the device (see 6.2.6). Recommended data type is STRING32.

5.3.10.5 Model number

Identifies the model identification number, specified by the manufacturer. Recommended data type is STRING32.

5.3.10.6 Software revision

Identifies the software or firmware version of microprocessor code that is contained within the device, specified by the manufacturer. Recommended data type is STRING8.

5.3.10.7 Hardware revision

Identifies the version of the device, excluding the software or firmware version of microprocessor code, specified by the manufacturer. Recommended data type is STRING8.

5.3.10.8 Serial number

Identifies the number or string, defined and assigned by the manufacturer that uniquely identifies each individual device or batch of devices produced. Recommended data type is STRING32.

5.3.10.9 Additional information

Contains any additional device information specified by the manufacturer. Recommended data type is STRING64.

5.4 Complex data types (root device profile)

5.4.1 General

Some parameters can require the use of complex data types (arrays, structures or enumerations), in addition to the simple data types listed in Table 1.

The IEC product committee may define one or more complex data types in accordance with 5.4.2 to 5.4.4.

5.4.2 Array data type

5.4.2.1 General

An array is a collection of elements of the same data type; the data type of the elements can be simple or complex. Each element of an array is associated with an index (number) within a specified range, corresponding to the number of elements in the array. This index is used to access each element of an array individually.

EXAMPLE Individual elements within an array of four elements may be accessed using indexes, e.g. 1 to 4.

NOTE When implementing an array, enough data storage needs to be allocated for each element (based on the array data type), and for the number of elements which can be indexed by the specified index range.

The definition of an array data type uses a single row in the template.

This is illustrated by the “Current measure” array data type in Figure 2 example.

Data type name	Category	Number of elements or element names	Element data type	*Additional information*
Current measure	Array	3	UINT	Current L1-L3

Figure 2 – Array data type example *IEC 2231/07*

5.4.2.2 Data type name (mandatory)

The “Data type name” field shall contain a descriptive text name of the array data type (maximum 32 characters).

5.4.2.3 Category (mandatory)

The “Category” field shall specify the category of the complex data type. For an array data type, category shall be specified as “Array”.

5.4.2.4 Number of elements or element names (mandatory)

The “Number of elements or element names” field shall specify the number of elements in the array data type, i.e. the maximum number of indexes.

5.4.2.5 Element data type (mandatory)

The array data type shall identify the data type of its elements, see example given in Figure 2.

For an array data type, the “Element data type” field shall contain a type name selected by the IEC product committee either from the valid simple data types listed in Table 1 or from the complex data types defined in this subclause.

5.4.2.6 Additional information (optional)

The “Additional information” field shall, if used, contain a text description providing additional information on the use of the array data type.

5.4.3 Structured data type

5.4.3.1 General

A structure is a collection of named elements, which can be of different data types; the data types of the elements can be simple or complex. Each element within a structured data type is associated with a specified name. This name is used to access each element of a structure individually, in addition to the structure name.

EXAMPLE For instance, a parameter named “Motor_1_status”, of data type Status, will contain a Ramping element that may be accessed using “Motor_1_status.Ramping” (see Figure 3).

The definition of a structured data type uses (n+1) rows in the template, where n is the number of members in the structure. The first row provides general information on the structured data type, while the following rows specify the structure elements.

This is illustrated by the “Status” structured data type in Figure 3 example.

Data type name	Category	Number of elements or element names	Element data type	*Additional information*
Status	Struct	11	—	This data type is used to document motor status as follows:
—	—	Ready	BOOL	Motor ready
—	—	On	BOOL	Motor on
—	—	Fault	BOOL	Motor faulted
—	—	Warning	BOOL	Motor warning
—	—	—	BOOL	Manufacturer Specific 1
—	—	—	BOOL	Manufacturer Specific 2
—	—	—	BOOL	Manufacturer Specific 3
—	—	—	BOOL	Manufacturer Specific 4
—	—	Current	USINT	Motor current (range limited to 6 bits)
—	—	Local_Control	BOOL	In local control
—	—	Ramping	BOOL	Ramping (motor starting)

Figure 3 – Structured data type example

IEC 2232/07

5.4.3.2 Data type name (mandatory)

On the first row, the “Data type name” field shall contain a descriptive text name of the structured data type (maximum 32 characters).

On the following rows, the “Data type name” field shall contain “—“ (em dash).

5.4.3.3 Category (mandatory)

The “Category” field shall specify the category of the complex data type. For a structured data type, category shall be specified as “Struct” on the first row.

On the following rows, the “Category” field shall contain “—“ (em dash).

5.4.3.4 Number of elements or element names (mandatory)

The “Number of elements or element names” field shall provide information on the structure elements.

On the first row, the “Number of elements or element names” field shall contain the number of elements in the structured data type.

On the following rows, the “Number of elements or element names” field shall contain either

- a descriptive text name of each structure element (maximum 32 characters), or
- a “—“ (em dash) for fields left undefined in a root profile (i.e. to be defined in the manufacturer’s device profile).

As a result of this rule, a manufacturer’s device profile shall not contain any em dash in this field.

5.4.3.5 Element data type (mandatory)

The structured data type shall identify the individual data type of its elements, see examples given in Annex B.

On the first row, the “Element data type” field shall contain “—“ (em dash).

On the following rows, the “Element data type” field shall contain a type name selected by the IEC product committee either from the valid simple data types listed in Table 1 or from the complex data types defined in this subclause.

5.4.3.6 Additional information (optional)

On the first row, the “Additional information” field shall, if used, contain a text description providing additional information on the use of the structured data type.

On the following rows, the “Additional information” field shall, if used, contain a text description providing additional information on the use of the corresponding structure element.

5.4.4 Enumerated data type

5.4.4.1 General

An enumerated data type defines an ordered set of enumerated values, starting with the first identifier of the enumeration list, and ending with the last. A parameter associated with an enumerated data type can only take one of the values given in the enumeration list of the data type.

EXAMPLE For instance, a parameter named “Motor_1_Control”, of data type “Local control”, may only take the values “On” or “Off” (respectively 1 or 0).

NOTE Values of an enumerated data type are often associated with numerals, but they need not be. Numerals will be used for data encoding (e.g. for network transmission), while enumerated values will be used for display.

The definition of an enumerated data type uses (*n*+1) rows in the template, where *n* is the number of enumerated values. The first row provides general information on the enumerated data type, while the following rows specify the enumerated values.

This is illustrated by the “Local Control 1”, “Local Control 2” and “Ramp type” enumerated data types in Figure 4 example.

Data type name	Category	Number of elements or element names	Element data type	*Additional information*
Local control 1	Enum	2	—	
—	—	—	—	“Off=Motor Off”
—	—	—	—	“On=Motor On”
Local control 2	Enum	2	BOOL	Local control can only take the two values listed below
—	—	—	—	“0=Off”
—	—	—	—	“1=On”
Ramp type	Enum	3	USINT	
—	—	—	—	“0=Linear”
—	—	—	—	“1=S ramp”
—	—	—	—	“2=U ramp”

Figure 4 – Enumerated data type examples

IEC 2233/07

5.4.4.2 Data type name (mandatory)

On the first row, the “Data type name” field shall contain a descriptive text name of the enumerated data type (maximum 32 characters).

On the following rows, the “Data type name” field shall contain “—“ (em dash).

5.4.4.3 Category (mandatory)

The “Category” field shall specify the category of the complex data type. For an enumerated data type, category shall be specified as “Enum” on the first row.

On the following rows, the “Category” field shall contain “—“ (em dash).

5.4.4.4 Number of elements or element names (mandatory)

On the first row, the “Number of elements or element names” field shall contain the number of possible values in the enumerated data type.

On the following rows, the “Number of elements or element names” field shall contain “—“ (em dash) on all rows.

5.4.4.5 Element data type (mandatory/optional)

If the data type field is used, it shall identify the individual data type associated with the values of the enumerated data type (see examples in Figure 4).

NOTE The data type specified in this field is the data type that will be used to actually encode the enumerated values during data transmission, typically a BOOL or USINT.

On the first row, the “Element data type” field shall contain either

– a “—“ (em dash), or
– a data type name selected by the IEC product committee from the valid simple data types listed in Table 1.

On the following rows, the “Element data type” field shall contain “—“ (em dash).

5.4.4.6 Additional information (mandatory)

On the first row, the “Additional information” field shall, if used, contain a text description providing additional information on the use of the enumerated data type.

On each of the following rows, the “Additional information” field shall contain a description of the specific meaning of the enumerated values, formatted as a value followed by an equal sign (=) followed by the value meaning. There shall be no spaces immediately before or after the equal sign. The entire string shall be enclosed within quotation marks as shown in the following example.

EXAMPLE "0=Off"
"1=On"

5.5 Parameter assemblies (root device profile)

5.5.1 General

The IEC product committee may define one or more parameter assemblies in accordance with 5.5.2 to 5.5.5.

All parameter assemblies within a root device profile shall be optional (see 5.5.4).

A root device profile may contain multiple parameter assemblies. Individual parameters may be represented in multiple parameter assemblies within a root device profile.

Parameter assemblies shall define the data structures for exchange of one or more parameters and shall be independent of the operating system and network technology. Parameter assemblies can be used to read information from a device, write information to a device, or both.

Parameter assemblies shall specify the location of the parameters within the assemblies.

NOTE 1 Assemblies are typically used to increase efficiency of data exchanges.

NOTE 2 Parameter assemblies do not necessarily represent the ordering of the data within the network message.

Within each parameter assembly, any fields which are not part of the parameter assembly’s data, and are specified only for byte alignment purpose, shall be labelled as “na” and shall be considered undefined. These fields may be discarded when the assembly is transmitted over a bit-oriented network.

5.5.2 Parameter assembly name (mandatory)

The “Parameter assembly name” field shall contain a descriptive text name of the parameter assembly (maximum 32 characters).

5.5.3 Access (mandatory)

The “Access” field shall specify the access to the parameter assembly allowed through the network.

Access shall be specified as either

R for parameter assemblies readable from the connected device, or

W for parameter assemblies writable to the connected device, or

RW for assemblies which have both read access and write access.

Read access parameter assemblies may contain read and/or read/write access parameters.

Write and read/write access parameter assemblies shall only contain read/write access parameters.

5.5.4 Required (mandatory)

All parameter assemblies within a root device profile are optional. Therefore, the “Required” field shall contain the letter “O”.

5.5.5 Parameter assembly data (mandatory)

The parameter assembly shall identify the particular parameters using the parameter names given in the template field – see examples given in Annex B. Parameters that span a series of bytes, such as a text string, shall list the range of bytes in the “Byte” field of the parameter assembly.

Where multiple byte parameters are used within a parameter assembly, they shall start on byte boundaries, bit zero.

NOTE Byte ordering within multiple byte parameters is technology-specific and is therefore not specified.

The device profile format specified in this part of IEC 61915 is intended to be compatible with devices connected to both bit- and byte-oriented networks.

In the case of a bit-oriented network, the three description formats shown in Figure 5, Figure 6 and Figure 7 are equivalent.

Bit	
0	Parameter A
1	Parameter B
2	Parameter C
3	Parameter D

Figure 5 – Example description format (1)

IEC 2234/07

Bit	3	2	1	0
	Parameter D	Parameter C	Parameter B	Parameter A

Figure 6 – Example description format (2)

IEC 2235/07

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	na	na	na	na	Param. D	Param. C	Param. B	Param. A

Figure 7 – Example description format (3)

IEC 2236/07

In order to be able to use the same format, whatever the network type, the format of Figure 7 has been selected as the general template since it is suitable for both bit- and byte-oriented networks.

5.6 Parameter groups (root device profile)

5.6.1 General

It can be necessary for parameters in a device to be organized into groups, e.g. for consistency, or because some complex devices use a large number of parameters.

Parameter groups shall define logical sets of parameters. They shall be independent of the operating system and network technology.

NOTE In contrast to parameter assemblies, parameter groups are not defined to increase efficiency of data exchanges. Instead, they are mainly defined for the purpose of organizing long lists of parameters into meaningful sets (e.g. for HMIs).

These groups may be defined using either the operational categories shown in Annex E as a basis, or in relation with functional elements in a device (see 5.7.1).

The IEC product committee may define one or more parameter groups in accordance with 5.6.2 to 5.6.8.

All parameter groups within a root device profile shall be optional (see 5.6.5).

A root device profile may contain multiple parameter groups. Individual parameters may be represented in multiple parameter groups within a root device profile.

EXAMPLE In a motor starter profile, the parameter “Motor thermal state” may be included in both groups “Operational measurements” corresponding to the operation category, and “Motor thermal protection” corresponding to the functional element.

Parameter groups may be nested, i.e. it is possible to define a parameter group composed of other parameter groups.

5.6.2 Group name (mandatory)

The “Group name” field shall contain a descriptive text name of the parameter group (maximum 32 characters).

5.6.3 Group type (mandatory)

The “Group type” field shall specify the type of the parameter group, i.e. the type of its members.

Group type shall be specified as either

P for parameter groups composed of parameters, or

G for parameter groups composed of other parameter groups.

NOTE The “G” type is used to define nesting of groups.

5.6.4 Number of members (mandatory)

The “Number of members” field shall specify the number of members (parameters or other groups) in the parameter group.

5.6.5 Required (mandatory)

All parameter groups within a root device profile are optional. Therefore, the “Required” field shall contain the letter “O”.

5.6.6 Description (optional)

The “Description” field shall, if used, contain a text description of the parameter group and/or its use.

5.6.7 Additional information (optional)

The “Additional information” field shall, if used, contain a text description providing additional information on the use of the parameter group.

NOTE This may include specific conditions for handling parameters within the group (e.g. have dependency relationships, or can only be invoked via secure access, or from certain state chart diagram states), or a reference to an external file containing the additional information required to use the parameters (e.g. executable file, description file).

5.6.8 Member names (mandatory)

The parameter group shall identify the particular members (parameters or other groups) using the parameter/group names given in the template field – see examples given in Annex B.

5.7 Functional elements (root device profile)

5.7.1 General

Some complex devices can be structured into physical modules, logical modules and/or functional elements (FE), as shown in Figure 8.

All functional elements are associated with parameters and typically corresponding behaviour. Functional elements can be specified using parameter lists, function blocks or objects.

NOTE Functional elements (parameter lists, function blocks or objects) are further detailed in IEC/TR 62390.

Modules and functional elements can be hierarchically structured. In some cases, the hierarchy of device, module and functional element can be merged; for example, if a device has only one module with a single functional element, it may only provide a parameter list.

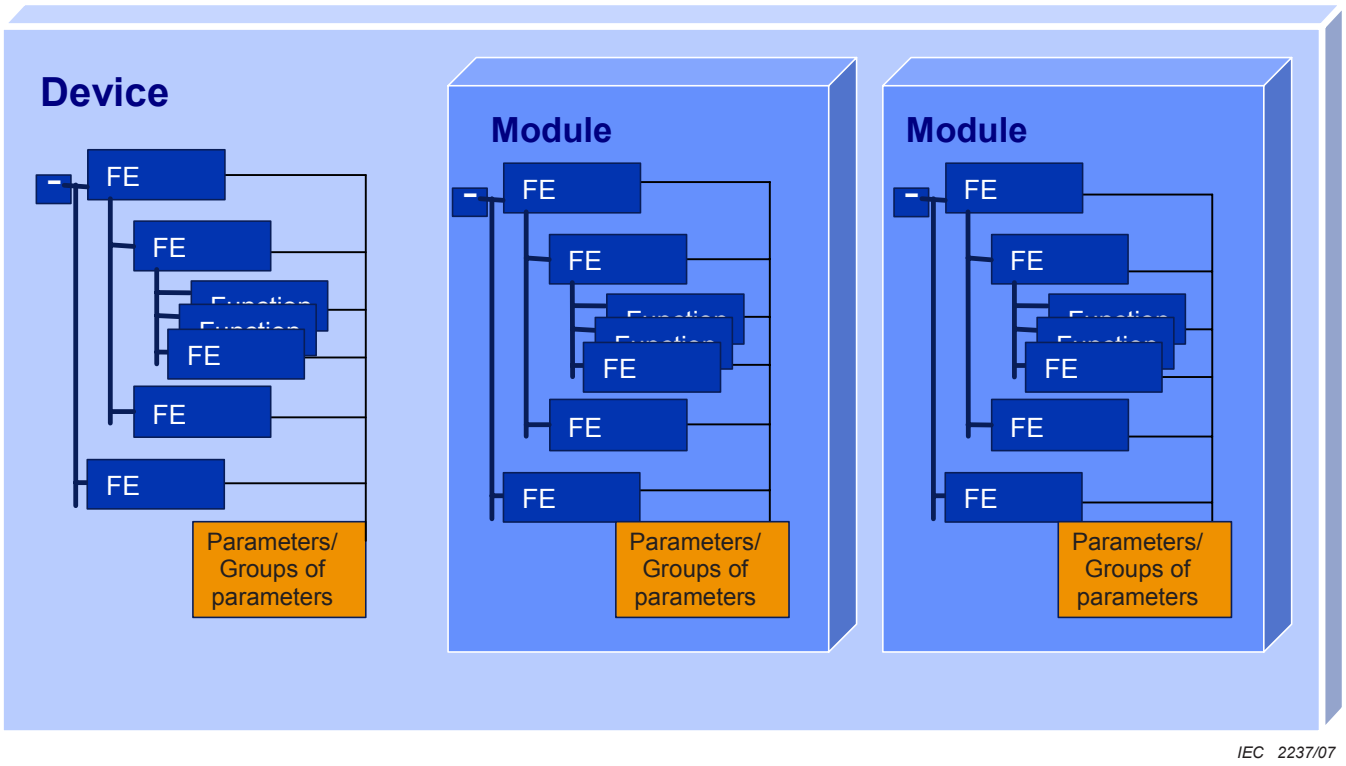
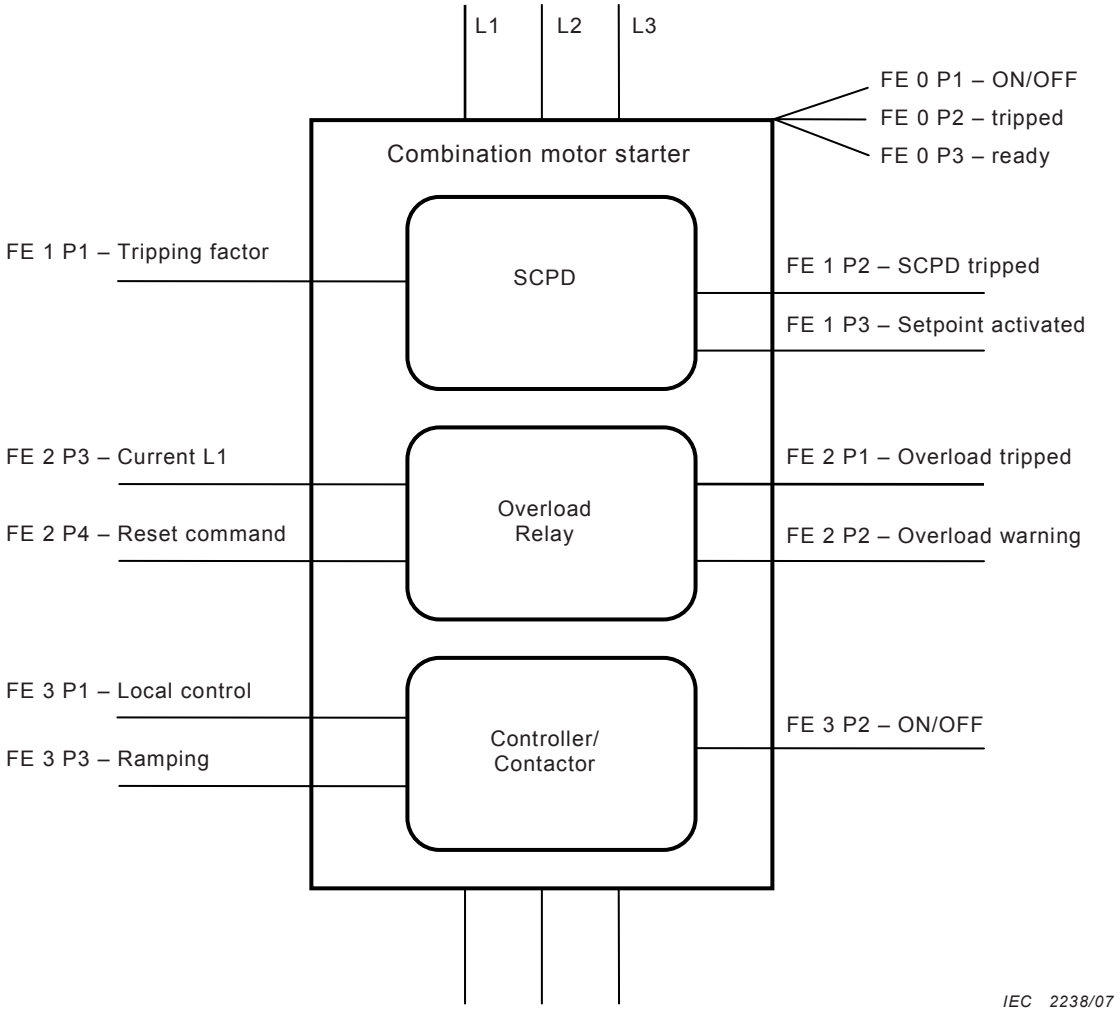


Figure 8 – Example device structure

EXAMPLE An example of a combination motor starter is shown in Figure 9.



Key

FE = functional element

P1 = parameter 1

SCPD = short circuit protective device

NOTE 1 Setpoint activated = SCPD in the reset position.

NOTE 2 Tripping factor is needed to set the overcurrent protection.

NOTE 3 Current L1 is needed to set the overload current protection.

Figure 9 – Combination motor starter example

The IEC product committee may define one or more functional elements, together with an associated functional structure diagram.

All functional elements within a root device profile shall be optional.

5.7.2 Functional structure diagram (optional)

Functional structure diagrams shall be a graphical representation showing the relationships between the functional elements of the device (see 5.7.3). These may be either function block diagrams or object diagrams.

NOTE The need for this diagram depends on the complexity of the relationships between the functional elements.

5.7.3 Functional element list (optional)

5.7.3.1 General

The IEC product committee may define one or more functional elements (function blocks or objects) in accordance with 5.7.3.2 to 5.7.3.4.

5.7.3.2 Functional element name (mandatory)

The “Functional element name” field shall contain a text string (maximum 32 characters) specified by the IEC product committee.

5.7.3.3 Required (mandatory)

All functional elements within a root device profile are optional. Therefore, the “Required” field shall contain the letter “O”.

5.7.3.4 Parameter group (optional)

The “Parameter group” field shall, if used, contain the name of the parameter group associated with the functional element.

5.7.3.5 State model (optional)

The “State model” field shall, if used, contain the name of the state model (state chart diagram and state transition table) associated with the functional element.

5.7.3.6 Description (mandatory)

The “Description” field shall contain a text description of the functional element and/or its use.

NOTE This field is mandatory in this case, as this is the only way to specify the functional element.

5.8 State model (root device profile)

5.8.1 General

All profiles shall define at least one state model for the device. They may include a state model for the entire device, for functional elements, or both.

A state model comprises a state chart diagram and a state transition table. A state model aids in understanding the device behaviour or operation of the device as seen through the network. The state model clarifies how an external influence can affect the state of the device or when the internal state of the device affects its observable behaviour.

All device states that are visible through the network shall be defined.

Network-specific states are outside the scope of this part of IEC 61915.

NOTE It is the responsibility of the profile designer to decide upon the level of complexity of the state model. It is expected that state models included in root device profiles will be typically simpler than those used by manufacturers.

5.8.2 State model name

The “State model name” field shall, if used, contain a descriptive text name of the state model (state chart diagram and state transition table) (maximum 32 characters).

This field is mandatory if the root profile contains more than one state model. Otherwise it is optional.

5.8.3 State chart diagrams

State chart diagrams shall be a graphical representation of the device behaviour and shall be in accordance with ISO/IEC 19501. All states shall be identified by their state name (see 5.8.4.2) and all transitions shall be numbered.

An example of a state chart diagram is shown in Figure 10.

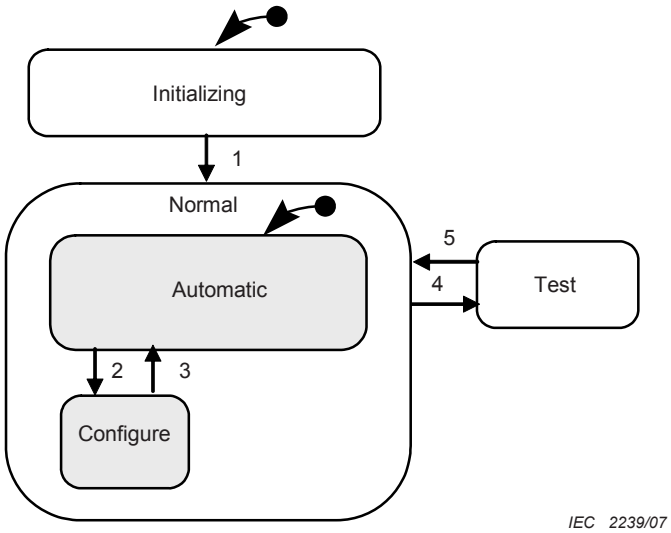
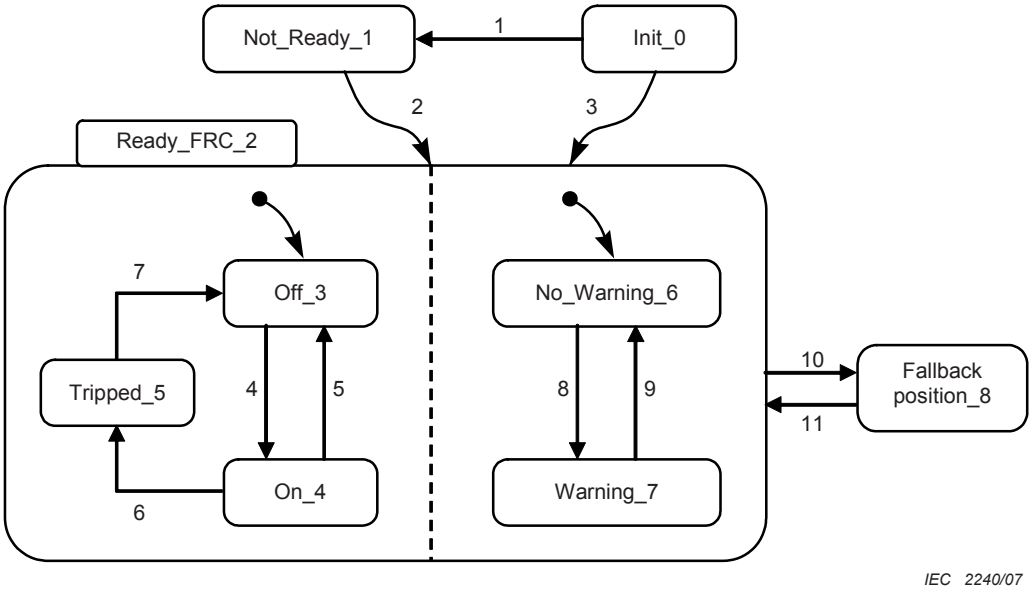


Figure 10 – Example of a state chart diagram for a photoelectric switch

Another example of a state chart diagram for a motor starter is shown in Figure 11.



NOTE

1 = resume_1	4 = on	7 = reset	10 = fallback position
2 = automatic	5 = off	8 = warning coming	11 = ready
3 = resume_2	6 = protection	9 = warning going	

Figure 11 – Example of a state chart diagram for a motor starter

5.8.4 State transition tables

5.8.4.1 General

State transition tables complement the state chart diagrams. The format of the state transition table shall be as shown in the device profile template (see Annex A).

State transition tables shall describe each state and the events that cause each transition to take place. These events may include commands sent via the network, internally generated events, or external events detected by the connected device.

Figure 12 and

Figure 13 show the state transition tables corresponding respectively to the state chart diagrams shown in Figure 10 and Figure 11.

NOTE It is recommended that all root device profiles include the ability to report the current state of the device.

STATE NAME			STATE DESCRIPTION
Initializing			Initial state of the device upon power-up. The device is not yet available for normal operation
Normal			The device is available for automatic operation
Automatic			"Presence" and "Alarm" parameter values are available to the network
Configure			When in this state, the device can be commanded by the "Operate mode" parameter to alter its operation accordingly (light or dark signal indicates presence) The device will not perform its normal sensing operations in this state – the "Presence" and "Alarm" parameter values should not be read by the network
Test			The device does not perform its normal sensing operations. The "Presence" and "Alarm" parameter values are set to one (1)
TRAN-SITION	**SOURCE STATE**	**TARGET STATE**	**EVENT**
01	Initializing	Normal	Device initialised and ready for normal operation
02	Automatic	Configure	"Device mode" parameter is commanded to change from zero (0) to one (1), or service "Set configure mode" is invoked
03	Configure	Automatic	"Device mode" parameter is commanded to change from one (1) to zero (0), or service "Set automatic mode" is invoked
04	Normal	Test	"Test" parameter is commanded to change from zero (0) to one (1), or service "Enter test mode" is invoked
05	Test	Normal	"Test" parameter is commanded to change from one (1) to zero (0), or service "Exit test mode" is invoked

IEC 2241/07

Figure 12 – State transition table for the photoelectric switch example

STATE NAME			STATE DESCRIPTION
Init_0			Self test; initialisation of variables and values.
Not_Ready_1			All required conditions for operating are being prepared. NOTE "Monitoring" may still be possible, even if the switching device is in the “Not_Ready_1” state.
Ready_FRC_2			Starter is ready for remote control by the host controller.
Off_3			Starter in the “‘Off” state; main contacts are opened.
On_4			Starter in the “On” state; main contacts are closed.
Tripped_5			Starter in the “Off” state; main contacts are opened; trip reset required.
No_Warning_6			No warning condition exists.
Warning_7			Warning condition exists.
Fallback_position_8			A communication fault has occured. The starter is forced at a pre-configured “Fallback_position” (“Off” state or “On” state).
TRAN-SITION	**SOURCE STATE**	**TARGET STATE**	**EVENT**
01	Init_0	Not_Ready_1	Initial resume conditions made; not all required conditions for remote control by the host controller are fulfilled.
02	Not_Ready_1	Ready_FRC_2	All required conditions for remote control by the host controller are fulfilled.
03	Init_0	Ready_FRC_2	All required conditions for remote control by the host controller are fulfilled.
04	Off_3	On_4	On command executed.
05	On_4	Off_3	Off command executed.
06	On_4	Tripped_5	Tripping conditions exist; tripping happens (protection).
07	Tripped_5	Off_3	Tripping condition removed; trip reset proceeded.
08	No_Warning_6	Warning_7	Warning condition occurs.
09	Warning_7	No_Warning_6	Warning condition no longer exists.
10	Ready_FRC_2	Fallback_position_8	Communication with the network has failed.
11	Fallback_position_8	Ready_FRC_2	Communication with the network is re-established. Communication fault acknowledged.

IEC 2242/07

Figure 13 – State transition table for the motor starter example

5.8.4.2 State name

The “State name” field shall contain the text name of the state (maximum 32 characters).

5.8.4.3 State description

The “State description” field shall contain a text description of the state and/or its use.

5.8.4.4 Transitions

Each transition shall be numbered. The source state and the target state of each transition shall be specified. The “Event” field describes the events and conditions that cause the transition to occur.

5.9 Services (root device profile)

5.9.1 General

A root device profile may contain one or more services. Services may be provided to allow a user or an application to execute specific actions such as initiate a state transition, trigger a command, configure or program the device. A service may be associated with an exchange of parameters with the device, or with specific events of the state chart diagram.

NOTE 1 The service may be provided by the device or one of its functional elements.

NOTE 2 Read/write operations are not considered as services for the purpose of 5.9.

EXAMPLE Examples of services are fault reset, calibrate, identify, diagnostics.

5.9.2 Service name (mandatory)

The “Service name” field shall contain a text string (maximum 32 characters) specified by the IEC product committee.

5.9.3 Request parameter group (optional)

The “Request parameter group” field shall, if used, reference the name of a parameter group to identify parameters sent to the device in conjunction with a service request.

5.9.4 Response parameter group (optional)

The “Response parameter group” field shall, if used, reference the name of a parameter group to identify parameters sent from the device in conjunction with a service response.

5.9.5 Required (mandatory)

A root device profile shall specify whether the device is required to support each service. The root device profile's “Required” field shall contain either

M for mandatory services, i.e. those required to be supported by the device; or

O for optional services.

5.9.6 Description (optional)

The “Description” field shall, if used, contain a text description of the service and/or its use.

5.9.7 Additional information (optional)

The “Additional information” field shall, if used, contain a text description providing additional information on the use of the service.

NOTE This may include specific conditions for using the service (e.g. can only be invoked via secure access, or from certain state chart diagram states), or a reference to an external file containing the additional information required to use the service (e.g. executable file, description file).

6 Creating a manufacturer's device profile using a root device profile

6.1 General

Profile developers (device manufacturers or other organizations) shall add manufacturer-specific information to the following sections of the root device profile (see Annex A) when applicable in order to create the manufacturer's device profile:

– manufacturer's device profile header (see 6.2);
– implementation of root device profile parameters (see 6.3);
– parameters (manufacturer-specific) (see 6.4);
– implementation of root device profile complex data types (see 6.5);
– complex data types (manufacturer-specific) (see 6.6);
– implementation of root device profile parameter assemblies (see 6.7);
– parameter assemblies (manufacturer-specific) (see 6.8);
– implementation of root device profile parameter groups (see 6.9);
– parameter groups (manufacturer-specific) (see 6.10);
– implementation of root device profile functional elements (see 6.11);
– functional elements (manufacturer-specific) (see 6.12);
– state model (manufacturer-specific) (see 6.13);
– implementation of root device profile services (see 6.14);
– services (manufacturer-specific) (see 6.15).

6.2 Manufacturer's device profile header

6.2.1 General

The manufacturer's device profile header contains identification information for the manufacturer's device profile: the manufacturer's device profile ID, manufacturer's device profile description, manufacturer's device profile version, manufacturer's device profile release date, manufacturer ID, model compatibility, software compatibility and hardware compatibility.

If the specific device profile is based on a generic device profile (e.g. from a users' organization), a second manufacturer's device profile header may be added below the first one to identify the generic device profile.

NOTE In this case, the field "Profile type" should be completed with "Device" in the first header, and "Generic" in the second header (see 6.2.10).

6.2.2 Manufacturer's device profile ID (mandatory)

The profile developer shall insert the manufacturer's device profile ID.

NOTE If the manufacturer's device profile is supplied as an electronic file, then the filename should contain the manufacturer's device profile ID.

6.2.3 Manufacturer's device profile description (optional)

The profile developer may insert text describing the device profile.

6.2.4 Manufacturer's device profile version (mandatory)

A profile version number shall be assigned by the profile developer for each manufacturer's device profile. A different version number shall be assigned when any changes occur to a manufacturer's device profile.

The initial release of a manufacturer's device profile shall be version 001. All profiles with a version number of 000 shall be considered unreleased.

The format for a profile version consists of a text string using the format VAAA, where

V is always the upper case letter V;

AAA is the version number.

6.2.5 Manufacturer's device profile release date (mandatory)

Each manufacturer's device profile shall contain a version release date. The release date of the manufacturer's device profile shall be a text string using the format YYYY-MM-DD (a 10-character string including the dashes), where

YYYY is the year;

MM is the month of the year (01-12);

DD is the day of the month (01-31).

6.2.6 Manufacturer ID (mandatory)

The manufacturer ID identifies the developer of the device profile. Each profile developer shall be responsible for specifying their manufacturer ID.

NOTE The manufacturer ID should be unique to each profile developer and should be the same for all manufacturer's device profiles the developer provides. It will usually be the trademarked company or brand name.

6.2.7 Model compatibility (optional)

The manufacturer may insert text indicating which of his models are compatible with this profile.

6.2.8 Software compatibility (optional)

The manufacturer may insert text indicating which of his software/firmware is compatible with this profile.

6.2.9 Hardware compatibility (optional)

The manufacturer may insert text indicating which of his hardware is compatible with this profile.

6.2.10 Profile type (mandatory)

Two main types of manufacturer's device profiles may be defined:

– a generic device profile for a family of similar devices (e.g. similar device types with differing feature levels),

– a specific device profile for a single device (e.g. a specific catalogue model).

The profile developer shall insert a text string to specify the profile type. This text string shall be “Generic” if the manufacturer’s device profile is for a family of similar devices, and “Device” if the manufacturer’s device profile is for a single device.

6.2.11 Profile availability (mandatory)

The profile developer shall insert a text string to specify whether the device profile information can be read from the device. This text string shall be “Yes” if it can be read, and “No” if this feature is not implemented.

6.2.12 Additional information (optional)

The profile developer may insert text providing additional information on the use of the device.

NOTE This may include specific conditions for using the device (e.g. requires management of secure access), or a reference to an external file containing the additional information required to use the device (e.g. executable file, description file).

6.3 Implementation of root device profile parameters

All parameters specified as mandatory by the root device profile shall be included in the manufacturer's device profile, and implemented in the corresponding manufacturer’s device(s). The profile developer shall insert the upper case “M” in the “Required” field of the manufacturer's device profile to indicate that the root device profile specifies this parameter as being mandatory.

Each parameter specified as optional by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each parameter specified as optional by the root device profile shall not be included in a specific device profile unless this parameter is actually implemented in the device.

For each parameter specified as optional by the root device profile, but included in the manufacturer's device profile, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the parameter is mandatory for the generic device profile) or upper case “O” (if it is still optional);

– if creating a specific device profile, the profile developer shall insert the upper case “A” (“applied”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the parameter is designated mandatory in the generic device profile) or the upper case “A” (if the parameter is designated optional in the generic device profile).

NOTE An “M” or “m” in the “Required” field of the specific device profile identifies mandatory parameters which the manufacturer is required to implement. An “A” indicates optional parameters which the manufacturer has chosen to implement.

If no description has been assigned by the IEC product committee, then the profile developer may insert a description of the parameter in the manufacturer’s device profile.

6.4 Parameters (manufacturer-specific)

Profile developers may create additional manufacturer-specific parameters in accordance with 5.3.

For each manufacturer-specific parameter, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the parameter is mandatory for the generic device profile) or upper case “O” (if it is optional);

– if creating a specific device profile, the profile developer shall insert the upper case “D” (“device-specific”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the parameter is designated mandatory in the generic device profile), or upper case “A” (if the parameter is designated optional in the generic device profile), or upper case “D” (if the parameter is not defined in the generic device profile).

All manufacturer-specific parameters specified as mandatory in a generic device profile shall be implemented in the device by the manufacturer.

6.5 Implementation of root device profile complex data types

Complex data types which are used by mandatory parameters in the root device profile are also mandatory. Others are optional.

Each complex data type specified as optional by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each complex data type specified as optional by the root device profile shall not be included in a specific device profile unless this complex data type is actually used in the device.

If no additional information has been assigned by the IEC product committee, then the profile developer may insert additional information in the manufacturer’s device profile.

6.6 Complex data types (manufacturer-specific)

Profile developers may create additional manufacturer-specific complex data types in accordance with 5.4.

6.7 Implementation of root device profile parameter assemblies

Parameter assemblies are optional within a root device profile.

Each parameter assembly specified by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each parameter assembly specified by the root device profile shall not be included in a specific device profile unless this parameter assembly is actually implemented in the device.

For each parameter assembly specified by the root device profile, and included in the manufacturer's device profile, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the parameter assembly is mandatory for the generic device profile) or upper case “O” (if it is still optional);

– if creating a specific device profile, the profile developer shall insert the upper case “A” (“applied”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the parameter assembly is designated mandatory in the generic device profile) or the upper case “A” (if the parameter assembly is designated optional in the generic device profile).

NOTE An “m” in the “Required” field of the specific device profile identifies mandatory parameter assemblies which the manufacturer is required to implement. An “A” indicates optional parameter assemblies which the manufacturer has chosen to implement.

6.8 Parameter assemblies (manufacturer-specific)

Profile developers may create additional manufacturer-specific parameter assemblies in accordance with 5.5.

For each manufacturer-specific parameter assembly, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the parameter assembly is mandatory for the generic device profile) or upper case “O” (if it is optional);

– if creating a specific device profile, the profile developer shall insert the upper case “D” (“device-specific”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the parameter assembly is designated mandatory in the generic device profile), or upper case “A” (if the parameter assembly is designated optional in the generic device profile), or upper case “D” (if the parameter assembly is not defined in the generic device profile).

All manufacturer-specific parameter assemblies specified as mandatory in a generic device profile shall be implemented in the device by the manufacturer.

6.9 Implementation of root device profile parameter groups

Parameter groups are optional within a root device profile.

Each parameter group specified by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each parameter group specified by the root device profile shall not be included in a specific device profile unless this parameter group is actually implemented in the device.

For each parameter group specified by the root device profile, and included in the manufacturer's device profile, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the parameter group is mandatory for the generic device profile) or upper case “O” (if it is still optional);

– if creating a specific device profile, the profile developer shall insert the upper case “A” (“applied”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the parameter group is designated mandatory in the generic device profile) or the upper case “A” (if the parameter group is designated optional in the generic device profile).

NOTE An “m” in the “Required” field of the specific device profile identifies mandatory parameter groups which the manufacturer is required to implement. An “A” indicates optional parameter groups which the manufacturer has chosen to implement.

If no description or additional information has been assigned by the IEC product committee, then the profile developer may insert a description of the parameter group or additional information in the manufacturer’s device profile.

6.10 Parameter groups (manufacturer-specific)

Profile developers may create additional manufacturer-specific parameter groups in accordance with 5.6.

For each manufacturer-specific parameter group, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the parameter group is mandatory for the generic device profile) or upper case “O” (if it is optional);

– if creating a specific device profile, the profile developer shall insert the upper case “D” (“device-specific”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the parameter group is designated mandatory in the generic device profile), or upper case “A” (if the parameter group is designated optional in the generic device profile), or upper case “D” (if the parameter group is not defined in the generic device profile).

All manufacturer-specific parameter groups specified as mandatory in a generic device profile shall be implemented in the device by the manufacturer.

6.11 Implementation of root device profile functional elements

Functional elements are optional within a root device profile.

Each functional element specified by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each functional element specified by the root device profile shall not be included in a specific device profile unless this functional element is actually implemented in the device.

For each functional element specified by the root device profile, and included in the manufacturer's device profile, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the functional element is mandatory for the generic device profile) or upper case “O” (if it is still optional);

– if creating a specific device profile, the profile developer shall insert the upper case “A” (“applied”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the functional element is designated mandatory in the generic device profile) or the upper case “A” (if the functional element is designated optional in the generic device profile).

NOTE An “m” in the “Required” field of the specific device profile identifies mandatory functional elements which the manufacturer is required to implement. An “A” indicates optional functional elements which the manufacturer has chosen to implement.

6.12 Functional elements (manufacturer-specific)

Profile developers may create additional manufacturer-specific functional elements in accordance with 5.7.

For each manufacturer-specific functional element, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the functional element is mandatory for the generic device profile) or upper case “O” (if it is optional);

– if creating a specific device profile, the profile developer shall insert the upper case “D” (“device-specific”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the functional element is designated mandatory in the generic device profile), or upper case “A” (if the functional element is designated optional in the generic device profile), or upper case “D” (if the functional element is not defined in the generic device profile).

All manufacturer-specific functional elements specified as mandatory in a generic device profile shall be implemented in the device by the manufacturer.

6.13 State model (manufacturer-specific)

The manufacturer’s device profile shall use the state models of the root device profile.

The profile developer may

– define sub-states of simple states (i.e. states that do not already contain sub-states) that are already specified in the root device profile state model;

– define states concurrent to those already specified in the root device profile state model.

The profile developer may not

– modify any states already defined in the root device profile state model (except as specified above);

– add, delete or modify any transitions which connect to states already defined in the root device profile state model.

Any alteration of the root device profile state models other than that allowed in this clause shall be considered to be a new device type, and a new manufacturer’s device profile shall be developed in accordance with Clause 7.

However, the profile developer may add auxiliary state models to describe the behaviour of complex manufacturer-specific functional elements.

NOTE Network-specific states are outside the scope of IEC 61915.

6.14 Implementation of root device profile services

All services specified as mandatory by the root device profile shall be included in the manufacturer's device profile, and implemented in the corresponding manufacturer’s device(s). The profile developer shall insert the upper case “M” in the “Required” field of the manufacturer's device profile to indicate that the root device profile specifies this service as being mandatory.

Each service specified as optional by the root device profile may be omitted, or included without alteration (as mandatory or optional) in a generic device profile. Each service specified as optional by the root device profile shall not be included in a specific device profile unless this service is actually implemented in the device.

For each service specified as optional by the root device profile, but included in the manufacturer's device profile, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the service is mandatory for the generic device profile) or upper case “O” (if it is still optional);

– if creating a specific device profile, the profile developer shall insert the upper case “A” (“applied”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the service is designated mandatory in the generic device profile) or the upper case “A” (if the service is designated optional in the generic device profile).

NOTE An “M” or “m” in the “Required” field of the specific device profile identifies mandatory services which the manufacturer is required to implement. An “A” indicates optional services which the manufacturer has chosen to implement.

If no description or additional information has been assigned by the IEC product committee, then the profile developer may insert a description of the service or additional information in the manufacturer’s device profile.

6.15 Services (manufacturer-specific)

Profile developers may create additional manufacturer-specific services in accordance with 5.9.

For each manufacturer-specific service, the profile developer shall complete the “Required” field as follows:

– if creating a generic device profile, the profile developer shall insert the lower case “m” (if the service is mandatory for the generic device profile) or upper case “O” (if it is optional);

– if creating a specific device profile, the profile developer shall insert the upper case “D” (device-specific”);

– if creating a specific device profile, but based on a generic device profile, the profile developer shall insert the lower case “m” (if the service is designated mandatory in the generic device profile), or upper case “A” (if the service is designated optional in the generic device profile), or upper case “D” (if the service is not defined in the generic device profile).

All manufacturer-specific services specified as mandatory in a generic device profile shall be implemented in the device by the manufacturer.

7 Creating a manufacturer's device profile without using a root device profile

7.1 General

In the case where a suitable root device profile does not exist, then the profile developer may create a manufacturer's device profile using the rules given in this part of IEC 61915.

7.2 Manufacturer's device profile header

7.2.1 General

In accordance with 6.2.1.

7.2.2 Manufacturer's device profile ID (mandatory)

In accordance with 6.2.2.

7.2.3 Manufacturer's device profile description (optional)

In accordance with 6.2.3.

7.2.4 Manufacturer's device profile version (mandatory)

In accordance with 6.2.4.

7.2.5 Manufacturer's device profile release date (mandatory)

In accordance with 6.2.5.

7.2.6 Manufacturer ID (mandatory)

In accordance with 6.2.6.

7.2.7 Model compatibility (optional)

In accordance with 6.2.7.

7.2.8 Software compatibility (optional)

In accordance with 6.2.8.

7.2.9 Hardware compatibility (optional)

In accordance with 6.2.9.

7.2.10 Profile type (optional)

In accordance with 6.2.10.

7.2.11 Profile availability (optional)

In accordance with 6.2.11.

7.2.12 Additional information (optional)

In accordance with 6.2.12.

7.3 Root device profile header

7.3.1 Root device profile ID

No root device profile exists, therefore this field shall contain “na”.

7.3.2 Root device profile version

No root device profile exists, therefore this field shall contain “na”.

7.3.3 Root device profile release date

No root device profile exists, therefore this field shall contain “na”.

7.3.4 Device description (optional)

The manufacturer may insert a text string which describes the type of device.

7.4 Parameters (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.5 Parameters (manufacturer-specific)

In accordance with 6.4.

7.6 Complex data types (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.7 Complex data types (manufacturer-specific)

In accordance with 6.6.

7.8 Parameter assemblies (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.9 Parameter assemblies (manufacturer-specific)

In accordance with 6.8.

7.10 Parameter groups (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.11 Parameter groups (manufacturer-specific)

In accordance with 6.10.

7.12 Functional elements (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.13 Functional elements (manufacturer-specific)

In accordance with 6.12.

7.14 State model (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.15 State model (manufacturer-specific)

The manufacturer shall define the state model using the rules given in 5.8.

7.16 Services (root device profile)

No root device profile exists, therefore these fields shall remain empty.

7.17 Services (manufacturer-specific)

In accordance with 6.15.

Annex A
(normative)

Device profile template

The following Figure A.1 shows the device profile template.

The entire template is used as a basis for both root device profiles and manufacturer's device profiles. Unless otherwise instructed in this part of IEC 61915, unused fields shall remain empty.

NOTE 1 If some main template sections are completely empty (e.g. the manufacturer's header for a root device profile), these sections may be omitted in the profile.

NOTE 2 The shaded areas indicate those fields that will be completed by either the IEC product committee or the manufacturer.

NOTE 3 Optional fields of the template are in italics.

NOTE 4 Symbols are listed in 3.2 and their use is further detailed in the relevant clauses.

Table A.1 summarizes the possible contents of the "Required" fields included in several sections of the device profile template, depending on the definition process for a given device profile.

Table A.1 – Contents of the "Required" field in a device profile

Initial profile item definition	Contents of the "Required" fields		
	Root device profile	Generic device profile	Specific device profile
Profile items defined in the root device profile	M		M
	O		A or —
	M	M	M
	O	m	m
		O	A or —
		—	
Profile items defined in the generic device profile		m	m
		O	A or —
Profile items defined in the specific device profile			D

M = mandatory (from a root device profile)
m = mandatory (from a generic device profile)
O = optional
A = applied
D = device-specific
"—" = discarded, omitted

Cells with a grey background correspond to profile items which are not relevant for the considered relationship between profile types.

MANUFACTURER'S DEVICE PROFILE HEADER			
Manufacturer's device profile ID: <?????????> [See 6.2.2]	Manufacturer's device profile description: <?????????> [See 6.2.3]	Manufacturer's device profile version: VAAA [See 6.2.4]	Manufacturer's device profile release date: YYYY-MM-DD [See 6.2.5]
Manufacturer ID: <?????????> [See 6.2.6]	*Model compatibility:* <?????????> [See 6.2.7]	*Software compatibility:* <?????????> [See 6.2.8]	*Hardware compatibility:* <?????????> [See 6.2.9]
Profile type: <?????????> [See 6.2.10]	*Profile availability:* <?????????> [See 6.2.11]	*Additional information:* <?????????> [See 6.2.12]	

ROOT DEVICE PROFILE HEADER		
Root device profile ID: P(SB SN)PN [See 5.2.2]	Root device profile version: VAAA [See 5.2.3]	Root device profile release date: YYYY-MM-DD [See 5.2.4]
Device description: <DESCRIPTION OF DEVICE> [See 5.2.5]		

PARAMETERS (ROOT DEVICE PROFILE)

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
[See 5.3.2]	[See 5.3.3]	[See 5.3.4]	[See 5.3.5]	[See 5.3.5]	[See 5.3.6]	[See 5.3.7]	[See 5.3.8 or 6.3]	[See 5.3.9 or 6.3]

PARAMETERS (MANUFACTURER-SPECIFIC)

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
[See 6.4]	[See 6.4]	[See 6.4]	[See 6.4]	[See 6.4]	[See 6.4]	[See 6.4]	[See 6.4]	[See 6.4]

COMPLEX DATA TYPES (ROOT DEVICE PROFILE)

Data type name	Category	Number of elements or element names	Element data type	*Additional information*
[See 5.4.2.2, 5.4.3.2 or 5.4.4.2]	[See 5.4.2.3, 5.4.3.3 or 5.4.4.3]	[See 5.4.2.4, 5.4.3.4 or 5.4.4.4]	[See 5.4.2.5, 5.4.3.5 or 5.4.4.5]	[See 5.4.2.6, 5.4.3.6 or 5.4.4.6]

COMPLEX DATA TYPES (MANUFACTURER-SPECIFIC)

Data type name	Category	Number of elements or element names	Element data type	*Additional information*
[See 6.6]	[See 6.6]	[See 6.6]	[See 6.6]	[See 6.6]

PARAMETER ASSEMBLIES (ROOT DEVICE PROFILE)

Parameter assembly name: <???????????> [See 5.5.2]	Access: RW [See 5.5.3]	Required: O/m/A [See 5.5.4 or 6.7]

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions) [See 5.5.5]							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0								
1								
n								

PARAMETER ASSEMBLIES (MANUFACTURER-SPECIFIC)

Parameter assembly name: <???????????> [See 6.8]	Access: RW [See 6.8]	Required: m/O/A/D [See 6.8]

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions) [See 5.5.5]							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0								
1								
n								

PARAMETER GROUPS (ROOT DEVICE PROFILE)

Group name	Group type	Number of members	Required	*Description*	*Additional information*
[See 5.6.2]	[See 5.6.3]	[See 5.6.4]	[See 5.6.5 or 6.9]	[See 5.6.6]	[See 5.6.7]
Member names					
[See 5.6.8]					

PARAMETER GROUPS (MANUFACTURER-SPECIFIC)

Group name	Group type	Number of members	*Description*	*Additional information*
[See 6.10]	[See 6.10]	[See 6.10]	[See 6.10]	[See 6.10]
Member names				
[See 6.10]				

DEVICE FUNCTIONAL STRUCTURE (ROOT DEVICE PROFILE)

FUNCTIONAL ELEMENTS (ROOT DEVICE PROFILE)

FUNCTIONAL STRUCTURE DIAGRAM

[See 5.7.2]

FUNCTIONAL ELEMENT LIST [See 5.7.3]

Functional element name	Required	*Parameter group*	*State model*	Description
[See 5.7.3.2]	[See 5.7.3.3 or 6.11]	[See 5.7.3.4]	[See 5.7.3.5]	[See 5.7.3.6]

DEVICE BEHAVIOUR (ROOT DEVICE PROFILE)

STATE MODEL (ROOT DEVICE PROFILE)

***State model name*: [see 5.8.2]**

STATE CHART DIAGRAM

[See 5.8.3]

STATE TRANSITION TABLE [See 5.8.4]

STATE NAME	STATE DESCRIPTION
[See 5.8.4.2]	[See 5.8.4.3]

TRANSITION	SOURCE STATE	TARGET STATE	EVENT (events and conditions that cause the transition to occur)
[See 5.8.4.4]	[See 5.8.4.4]	[See 5.8.4.4]	[See 5.8.4.4]

SERVICES (ROOT DEVICE PROFILE)

Service name	*Request parameter group*	*Response parameter group*	Required	*Description*	*Additional information*
[See 5.9.2]	[See 5.9.3]	[See 5.9.4]	[See 5.9.5 or 6.14]	[See 5.9.6]	[See 5.9.7]

DEVICE FUNCTIONAL STRUCTURE (MANUFACTURER-SPECIFIC)				
FUNCTIONAL ELEMENTS (MANUFACTURER-SPECIFIC)				
FUNCTIONAL STRUCTURE DIAGRAM				
[See 6.12 or 7.13]				
FUNCTIONAL ELEMENT LIST [See 6.12 or 7.13]				
Functional element name	Required	*Parameter group*	*State model*	Description
[See 6.12 or 7.13]	[See 6.12 or 7.13]	[See 6.12 or 7.13]	[See 6.12 or 7.13]	[See 6.12 or 7.13]

DEVICE BEHAVIOUR (MANUFACTURER-SPECIFIC)

STATE MODEL (MANUFACTURER-SPECIFIC)

***State model name*: [See 6.13 or 7.15]**

STATE CHART DIAGRAM

[See 6.13 or 7.15]

STATE TRANSITION TABLE [See 6.13 or 7.15]

STATE NAME	STATE DESCRIPTION
[See 6.13 or 7.15]	[See 6.13 or 7.15]

TRANSITION	SOURCE STATE	TARGET STATE	EVENT (events and conditions that cause the transition to occur)
[See 6.13 or 7.15]	[See 6.13 or 7.15]	[See 6.13 or 7.15]	[See 6.13 or 7.15]

SERVICES (MANUFACTURER-SPECIFIC)

Service name	*Request parameter group*	*Response parameter group*	Required	*Description*	*Additional information*
[See 6.15]	[See 6.15]	[See 6.15]	[See 6.15]	[See 6.15]	[See 6.15]

Figure A.1 – Device profile template

Annex B
(informative)

Device profile examples

B.1 General

This annex contains the following examples of device profiles:

– root device profiles as developed by an IEC product committee in accordance with Clause 5 (see Clause B.2);

– manufacturer's device profile created in accordance with Clause 6 using the aforementioned root device profile (see Clause B.3);

– manufacturer's device profile created in accordance with Clause 7 without using a root device profile (see clause B.4).

The examples given in this annex are not actual profiles, they are only included here to show how the various fields can be filled. Actual root device profiles will be developed by IEC product committees, and actual manufacturer's device profiles will be developed by other organizations.

NOTE Symbols are listed in 3.2 and their use is further detailed in the relevant clauses.

B.2 Root device profile examples

Figure B.1 and Figure B.2 are only examples of root device profiles as may be developed by an IEC product committee. Actual root device profiles will be included in the subsequent parts of this standard.

NOTE 1 The shaded areas indicate those fields that are completed by the IEC product committee.

NOTE 2 Optional fields of the template are in italics.

B.2.1 Photoelectric switch

ROOT DEVICE PROFILE HEADER		
Root device profile ID: P(IEC 60947-5-2)23068	Root device profile version: V000	Root device profile release date: 2000-09-01
Device description: Photoelectric switch with mode control		

PARAMETERS (ROOT DEVICE PROFILE)								
Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Presence	BOOL	na	na	na	na	R	M	Indicates if the photoelectric switch has detected the presence of an object. "0=No presence sensed" "1=Presence sensed"
Alarm	BOOL	na	na	na	na	R	O	Indicates a marginal or failed sensing condition. "0=Alarm condition detected" "1=No alarm condition detected"
Device mode	BOOL	na	na	na	na	RW	M	Toggles device between automatic and configure states. "0=Command device state to automatic" "1=Command device state to configure"
Operate mode	BOOL	na	na	na	na	RW	M	Controls if a light or dark condition indicates presence. "0=Light signal indicates presence" "1=Dark signal indicates presence"
Test	BOOL	na	na	na	na	RW	O	Toggles device between normal and test states. "0=Command device state to normal" "1=Command device state to test"

PARAMETER ASSEMBLIES (ROOT DEVICE PROFILE)

Parameter assembly name: Presence and alarm	Access: R	Required: O

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	r	Alarm	Presence

Parameter assembly name: Configure	Access: RW	Required: O

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	r	Operate mode	Device mode

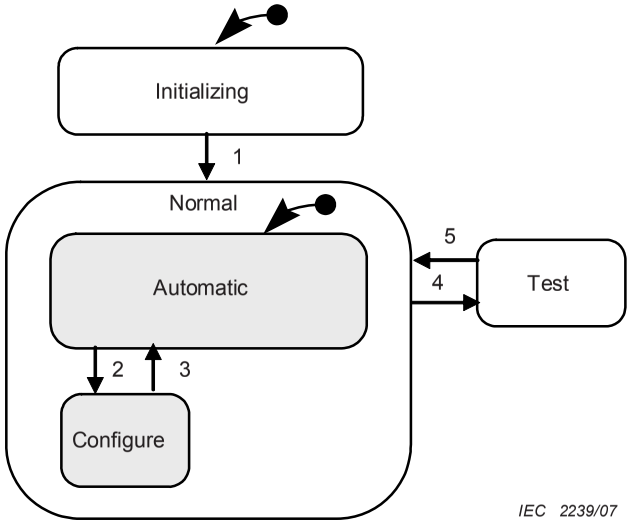
PARAMETER GROUPS (ROOT DEVICE PROFILE)

Group name	Group type	Number of members	Required	*Description*	*Additional information*
Mode changes	P	2	O	Parameters used for mode change	
Member names					
Device mode					
Test					

DEVICE BEHAVIOUR (ROOT DEVICE PROFILE)

STATE MODEL (ROOT DEVICE PROFILE)

STATE CHART DIAGRAM



STATE TRANSITION TABLE

STATE NAME	STATE DESCRIPTION
Initializing	Initial state of the device upon power-up. The device is not yet available for normal operation
Normal	The device is available for automatic operation
Automatic	"Presence" and "Alarm" parameter values are available to the network
Configure	When in this state, the device can be commanded by the "Operate mode" parameter to alter its operation accordingly (light or dark signal indicates presence) The device will not perform its normal sensing operations in this state, the "Presence" and "Alarm" parameter values should not be read by the network
Test	The device does not perform its normal sensing operations. The "Presence" and "Alarm" parameter values are set to one (1)

TRANSITION	SOURCE STATE	TARGET STATE	EVENT (events and conditions that cause the transition to occur)
01	Initializing	Normal	Device initialized and ready for normal operation
02	Automatic	Configure	“Device mode” parameter is commanded to change from zero (0) to one (1), or service “Set configure mode” is invoked
03	Configure	Automatic	“Device mode” parameter is commanded to change from one (1) to zero (0), or service “Set automatic mode” is invoked
04	Normal	Test	“Test” parameter is commanded to change from zero (0) to one (1)), or service “Enter test mode” is invoked
05	Test	Normal	“Test” parameter is commanded to change from one (1) to zero (0)), or service “Exit test mode” is invoked

SERVICES (ROOT DEVICE PROFILE)

Service name	*Request parameter group*	*Response parameter group*	Required	*Description*	*Additional information*
Set configure mode			M	Allows to enter configure mode	
Set automatic mode			M	Allows to exit configure mode	
Enter test mode			O	Allows to enter test mode	
Exit test mode			O	Allows to exit test mode	

Figure B.1 – Example of a root device profile – Photoelectric switch

B.2.2 Motor starter

ROOT DEVICE PROFILE HEADER

Root device profile ID:	Root device profile version:	Root device profile release date:
P(IEC 60947-4-2)12345	V000	2001-09-20
Device description: Motor starter		

PARAMETERS (ROOT DEVICE PROFILE)

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Ready	BOOL	na	na	na	na	R	O	All the conditions that will permit the operation of a starter by the remote host controller have been fulfilled "0=Not ready" "1=Ready" “Not ready” is the inverse of “Ready” NOTE 1 The conditions that have to be fulfilled are determined by the manufacturer of the starter. NOTE 2 Monitoring may still be present even if the starter is not ready, e.g. current can be transmitted because the starter is being operated under local control. NOTE 3 Examples of the conditions which have to be fulfilled include: – resetting of trips, – remote operation selected.
On	BOOL	na	na	na	na	R	O	The main circuit contacts are closed or the semiconductor starter is in the conducting state "0=Not on" "1=On" “Not on” is the inverse of “On” and is generally equivalent to the “Off” condition NOTE 1 In simple systems such as certain direct-on-line starters, the assumption may need to be made that when the starter is “On”, power is available at the incoming connections and the motor is running. In more complex systems, the status of the “On” input may be combined with current data to show if the motor is properly connected and running. NOTE 2 In semiconductor starters, the semiconductor “On” state implies that current is flowing to the motor.
Running forward	BOOL	na	na	na	na	R	O	Energy is being supplied to the motor so that its direction of rotation is forward "0=Not running" "1=Running forward" NOTE In unidirectional systems, this parameter signifies that the motor is running.

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Running reverse	BOOL	na	na	na	na	R	O	Energy is being supplied to the motor so that its direction of rotation is reverse "0=Not running" "1=Running reverse" NOTE In unidirectional systems, this parameter is not used.
Fault	BOOL	na	na	na	na	R	O	A fault condition exists "0=No fault" "1=Fault" “No fault” is the normal condition NOTE A fault condition is any condition which is abnormal and which required the tripping of the starter and the disconnection of the motor from the supply.
Alarm	BOOL	na	na	na	na	R	O	An alarm condition exists "0=No alarm" "1=Alarm" An alarm condition may develop into a fault condition if remedial action is not taken. “No alarm” is the normal condition NOTE An alarm condition is any condition which is abnormal but which does not require the immediate disconnection of the starter or the motor from the supply.
Local control	BOOL	na	na	na	na	R	O	The indication to a remote host controller that, as a result of operator intervention, commands received will not be accepted or acted upon "0=Remote control" "1=Local control" NOTE 1 Correlation between “Local control” and “Ready” should be explained. NOTE 2 This indication is supplementary to the “Ready” indication and may not be available in all instances.
Network control	BOOL	na	na	na	na	R	O	The indication to a remote host controller that, as a result of operator intervention, commands received will be accepted or acted upon "0=Network control" "1=Local control" NOTE This indication is supplementary to the “Ready” indication and may not be available in all instances.

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Ramping	BOOL	na	na	na	na	R	O	The condition when a semiconductor controller is accelerating or decelerating the motor by altering the voltage at the motor terminals to increase or decrease the motor torque "0=Not ramping" "1=Ramping" NOTE Refers to the “Soft starting” and “Soft stopping” modes of operation, and not to any voltage variation activity occurring while the motor is running at full-speed.
At reference	BOOL	na	na	na	na	R	O	The condition when a semiconductor controller is neither accelerating or decelerating the motor by altering the voltage at the motor terminals to increase or decrease the motor torque, but is at an intended intermediate voltage "0=Ramping" "1=Reference reached" NOTE Refers to the “Soft starting” and “Soft stopping” modes of operation, and not to any voltage variation activity occurring while the motor is running at full-speed such as “Optimising”.
Input N	BOOL	na	na	na	na	R	O	Signal of a digital input. "0=No signal on input N" "1=Signal on input N"
Motor current	UNIT	A	0	0,1	0 – 32767	R	O	The average instantaneous value of the current in the motor NOTE The average value of current can be derived in a variety of ways, e.g. by a computed true average value or from a line current chosen by an operator or manufacturer as being representative of the average instantaneous motor current.
Motor current (%)	USINT	%	0	3,125	0 – 63	R	O	The motor current expressed as a percentage of the motor rated current, I_e NOTE The maximum engineering value is 196,875 %.
Line current L1 (%)	USINT	%	0	3,125	0 – 63	R	O	The motor current expressed as a percentage of the motor rated current, I_e, in a specific phase NOTE The maximum engineering value is 196,875 %.
Line current L2 (%)	USINT	%	0	3,125	0 – 63	R	O	The motor current expressed as a percentage of the motor rated current, I_e, in a specific phase NOTE The maximum engineering value is 196,875 %.
Line current L3 (%)	USINT	%	0	3,125	0 – 63	R	O	The motor current expressed as a percentage of the motor rated current, I_e, in a specific phase NOTE The maximum engineering value is 196,875 %.

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Run forward	BOOL	na	na	na	na	RW	O	Instructs the starter to energise the motor in the forward direction "0=Stop" "1=Run forward" NOTE In the case of single-direction starters, this command is used to energise the motor.
Run reverse	BOOL	na	na	na	na	RW	O	Instructs the starter to energise the motor in the reverse direction "0=Stop" "1=Run reverse" NOTE In the case of single-direction starters, this command is not used.
Brake	BOOL	na	na	na	na	RW	O	Instructs the starter to release the brake to free the motor "0=Brake" "1=Release"
Fault reset	BOOL	na	na	na	na	RW	O	Instructs the starter to reset all re-settable faults "0=Fault reset inactive" "1=Fault reset"
Emergency start	BOOL	na	na	na	na	RW	O	Instructs the starter to override any fault condition and allows starting "0=Inactive" "1=Emergency Start" NOTE This command is required in process and other industries where sacrificial demands can be made on motors and associated equipment in order to achieve an orderly outcome in emergency circumstances.
Low speed	BOOL	na	na	na	na	RW	O	Instructs the starter to select slow speed running of a two-speed motor "0=High/normal speed" "1=Low speed" NOTE This command is used to select the motor to the low speed.
Self test	BOOL	na	na	na	na	RW	O	Instructs the starter to initiate an internal test routine within the starter "0=Inactive" "1=Test" NOTE The routine depends on the application and is at the discretion of the operator, user or manufacturer.
NOTE Configuration parameters may be added in actual versions of the device profile.								

PARAMETER ASSEMBLIES (ROOT DEVICE PROFILE)

Parameter assembly name: Monitoring type 1	Access: R	Required: O

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	Input 4	Input 3	Input 2	Input 1	Alarm	Fault	On	Ready

Parameter assembly name: Monitoring type 2	Access: R	Required: O

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	Input 4	Input 3	Input 2	Input 1	Alarm	Fault	On	Ready
1	Ramping	Local control	Motor current (%)					

Parameter assembly name: Monitoring type 3	Access: R	Required: O

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	Input 4	Input 3	Input 2	Input 1	Alarm	Fault	On	Ready
1	Ramping	Local control	Line current L1 (%)					
2	r	r	Line current L2 (%)					
3	r	r	Line current L3 (%)					

Parameter assembly name: Basic motor starter input	Access: R	Required: O

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	Running forward	r	Fault

Parameter assembly name: Extended motor starter input 1			Access: R		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	Network control	Ready	r	Running forward	Alarm	Fault

Parameter assembly name: Extended motor starter input 2			Access: R		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	Network control	Ready	Running reverse	Running forward	Alarm	Fault

Parameter assembly name: Basic softstart input			Access: R		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	At reference	r	r	r	r	Running forward	r	Fault

Parameter assembly name: Extended softstart input			Access: R		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	At reference	r	Network control	Ready	Running reverse	Running forward	Alarm	Fault

Parameter assembly name: Command type 1			Access: RW		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	Low speed	Self test	Emergency start	Fault reset	Brake	Run reverse	Run forward

Parameter assembly name: Command type 2			Access: RW		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	Low speed	Self test	Emergency start	Fault reset	Brake	Run reverse	Run forward
1	(Manufacturer specific)	(Manufacturer specific)	(Manufacturer specific)	(Manufacturer specific)	r	r	r	r

Parameter assembly name: Basic motor starter output			Access: RW		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	Reset	r	Run forward

Parameter assembly name: Extended contactor output			Access: RW		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	r	Run reverse	Run forward

Parameter assembly name: Extended motor starter output			Access: RW		Required: O			
Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	Reset	Run reverse	Run forward

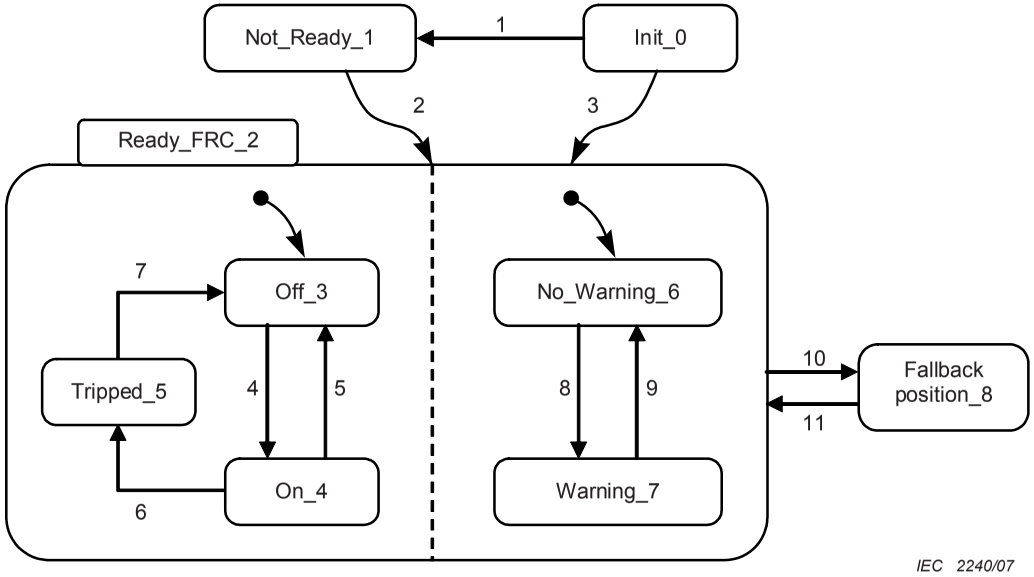
PARAMETER GROUPS (ROOT DEVICE PROFILE)

Group name	Group type	Number of members	Required	*Description*	*Additional information*
Monitoring signals	P	10	O	Control parameters	
Member names					
Ready					
On					
Running forward					
Running reverse					
Fault					
Alarm					
Local control					
Network control					
Ramping					
At reference					
Group name	Group type	Number of members	Required	*Description*	*Additional information*
Measurements	P	5	O	Control parameters	
Member names					
Motor current					
Motor current (%)					
Line current L1 (%)					
Line current L2 (%)					
Line current L3 (%)					
Group name	Group type	Number of members	Required	*Description*	*Additional information*
Control commands	P	7	O	Control parameters	
Member names					
Run forward					
Run reverse					
Brake					
Fault reset					
Emergency start					
Low speed					
Self test					

DEVICE BEHAVIOUR (ROOT DEVICE PROFILE)

STATE MODEL (ROOT DEVICE PROFILE)

STATE CHART DIAGRAM



STATE TRANSITION TABLE			
STATE NAME	STATE DESCRIPTION		
Init_0	Self test; initialisation of variables and values.		
Not_Ready_1	All required conditions for operating are being prepared. NOTE "Monitoring" may still be possible, even if the switching device is in the “Not_Ready_1” state.		
Ready_FRC_2	Starter is ready for remote control by the host controller.		
Off_3	Starter in the “‘Off” state; main contacts are opened.		
On_4	Starter in the “On” state; main contacts are closed.		
Tripped_5	Starter in the “Off” state; main contacts are opened; trip reset required.		
No_Warning_6	No warning condition exists.		
Warning_7	Warning condition exists.		
Fallback_position_8	A communication fault has occurred. The starter is forced at a pre-configured “Fallback_position” (“Off” state or “On” state).		
TRANSITION	SOURCE STATE	TARGET STATE	EVENT (events and conditions that cause the transition to occur)
01	Init_0	Not_Ready_1	Initial resume conditions made; not all required conditions for remote control by the host controller are fulfilled.
02	Not_Ready_1	Ready_FRC_2	All required conditions for remote control by the host controller are fulfilled.
03	Init_0	Ready_FRC_2	All required conditions for remote control by the host controller are fulfilled.
04	Off_3	On_4	On command executed.
05	On_4	Off_3	Off command executed.
06	On_4	Tripped_5	Tripping conditions exist; tripping happens (protection).
07	Tripped_5	Off_3	Tripping condition removed; trip reset proceeded.
08	No_Warning_6	Warning_7	Warning condition occurs.
09	Warning_7	No_Warning_6	Warning condition no longer exists.
10	Ready_FRC_2	Fallback_position_8	Communication with the network has failed.
11	Fallback_position_8	Ready_FRC_2	Communication with the network is re-established. Communication fault acknowledged.

Figure B.2 – Example of a root device profile – Motor starter

B.3 Generic device profile example (created using a root device profile)

The following Figure B.3 is an example of a manufacturer’s device profile for a family of devices (generic device profile) created using the root device profile defined in B.2.1.

NOTE The shaded areas indicate those fields that are completed by the profile developer.

MANUFACTURER'S DEVICE PROFILE HEADER			
Manufacturer's device profile ID: Photo-Prox	*Manufacturer's device profile description:* Photoelectric switch with time delay and target sensitivity	Manufacturer's device profile version: V002	Manufacturer's device profile release date: 2000-09-05
Manufacturer ID: Acme Sensors Inc.	*Model compatibility:* P-100-DS A	*Software compatibility:*	*Hardware compatibility:*
Profile type: Generic	*Profile availability:* No	*Additional information:*	

ROOT DEVICE PROFILE HEADER		
Root device profile ID: P(IEC 60947-5-2)23068	Root device profile version: V000	Root device profile release date: 2000-09-01
Device description: Photoelectric switch with mode control		

PARAMETERS (ROOT DEVICE PROFILE)

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Presence	BOOL	na	na	na	na	R	M	Indicates if the photoelectric switch has detected the presence of an object "0=No presence sensed" "1=Presence sensed"
Alarm	BOOL	na	na	na	na	R	m	Indicates a marginal or failed sensing condition "0=Alarm condition detected" "1=No alarm condition detected"
Device mode	BOOL	na	na	na	na	RW	M	Toggles device between automatic and configure states "0=Command device state to automatic" "1=Command device state to configure"
Operate mode	BOOL	na	na	na	na	RW	M	Controls if a light or dark condition indicates presence "0=Light signal indicates presence" "1=Dark signal indicates presence"
Test	BOOL	na	na	na	na	RW	m	Toggles device between normal and test states "0=Command device state to normal" "1=Command device state to test"

PARAMETERS (MANUFACTURER-SPECIFIC)

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Output mode	BOOL	na	na	na	na	RW	m	The output mode inverts the definition of the output switch "0=Normally open" "1=Normally closed"
On delay	UINT	ms	0	1	0… 65535	RW	m	Amount of time in ms that the output is held “Off” after an object is detected
Off delay	UINT	ms	0	1	0… 65535	RW	m	Amount of time in ms that the output is held “On” after no object is detected
One shot delay	UINT	ms	0	1	0… 65535	RW	m	Amount of time in ms that the output is held “On” after an object is detected
Sensitivity	USINT	na	0	1	0…255	RW	m	The sensitivity is the threshold setting of the detection device

PARAMETER ASSEMBLIES (ROOT DEVICE PROFILE)

Parameter assembly name: Presence and alarm	Access: R	Required: m

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	r	Alarm	Presence

PARAMETER ASSEMBLIES (MANUFACTURER-SPECIFIC)

Parameter assembly name: Advanced configure	Access: RW	Required: m

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	Output mode	Operate mode	Device mode
1-2	On delay							
3-4	Off delay							
5-6	One shot delay							
7	Sensitivity							

PARAMETER GROUPS (ROOT DEVICE PROFILE)

Group name	Group type	Number of members	Required	*Description*	*Additional information*
Mode changes	P	2	m	Parameters used for mode change	
Member names					
Device mode					
Test					

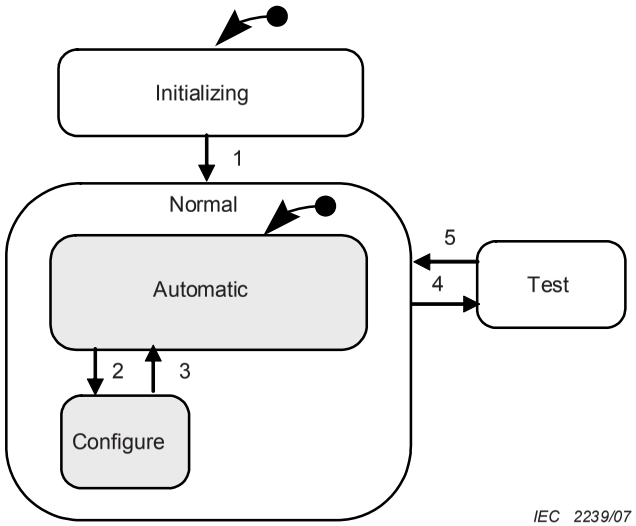
PARAMETER GROUPS (MANUFACTURER-SPECIFIC)

Group name	Group type	Number of members	Required	*Description*	*Additional information*
Delays	P	3	m	Parameters used for setting various delays	
Member names					
ON delay					
OFF delay					
One shot delay					

DEVICE BEHAVIOUR (ROOT DEVICE PROFILE)

STATE MODEL (ROOT DEVICE PROFILE)

STATE CHART DIAGRAM



STATE TRANSITION TABLE	
STATE NAME	STATE DESCRIPTION
Initializing	Initial state of the device upon power-up. The device is not yet available for normal operation.
Normal	The device is available for automatic operation.
Automatic	“Presence” and “Alarm” parameter values are available to the network.
Configure	When in this state, the device can be commanded by the “Operate mode” parameter to alter its operation accordingly (light or dark signal indicates presence). The device will not perform its normal sensing operations in this state, the “Presence” and “Alarm” parameter values should not be read by the network.
Test	The device does not perform its normal sensing operations. The “Presence” and “Alarm” parameter values are set to one (1)

TRANSITION	SOURCE STATE	TARGET STATE	EVENT (events and conditions that cause the transition to occur)
01	Initializing	Normal	Device initialized and ready for normal operation
02	Automatic	Configure	“Device mode” parameter is commanded to change from zero (0) to one (1), or service “Set configure mode” is invoked
03	Configure	Automatic	“Device mode” parameter is commanded to change from one (1) to zero (0), or service “Set automatic mode” is invoked
04	Normal	Test	“Test” parameter is commanded to change from zero (0) to one (1)), or service “Enter test mode” is invoked
05	Test	Normal	“Test” parameter is commanded to change from one (1) to zero (0)), or service “Exit test mode” is invoked

SERVICES (ROOT DEVICE PROFILE)					
Service name	*Request parameter group*	*Response parameter group*	Required	*Description*	*Additional information*
Set configure mode			M	Allows to enter configure mode	
Set automatic mode			M	Allows to exit configure mode	
Enter test mode			m	Allows to enter test mode	
Exit test mode			m	Allows to exit test mode	

Figure B.3 – Example of a generic device profile created using a root device profile

B.4 Specific device profile example (created without using a root device profile)

The following Figure B.4 is an example of a manufacturer’s device profile for a single device (specific device profile), created without using a root device profile.

NOTE The shaded areas indicate those fields that are completed by the manufacturer.

MANUFACTURER'S DEVICE PROFILE HEADER			
Manufacturer's device profile ID: Learn-Prox 1000	*Manufacturer's device profile description:* Photoelectric switch with learning and target sensitivity	Manufacturer's device profile version: V002	Manufacturer's device profile release date: 2000-09-05
Manufacturer ID: Sensors Inc.	*Model compatibility:* Smart-Prox Series A	*Software compatibility:* V001	*Hardware compatibility:*
Profile type: Device	*Profile availability:* No	*Additional information:*	

ROOT DEVICE PROFILE HEADER		
Root device profile ID: na	Root device profile version: na	Root device profile release date: na
Device description: Photoelectric switch with learning		

PARAMETERS (MANUFACTURER-SPECIFIC)

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Presence	BOOL	na	na	na	na	R	D	Indicates if the photoelectric switch has detected the presence of an object "0=No presence sensed" "1=Presence sensed"
Alarm	BOOL	na	na	na	na	R	D	Indicates a marginal or failed sensing condition "0=Alarm condition detected" "1=No alarm condition detected"
Device mode	BOOL	na	na	na	na	RW	D	Toggles device between automatic and configure states "0=Command device state to automatic" "1=Command device state to configure"

Parameter name	Data type	Units	Offset	Multiplier	Range	Access	Required	*Parameter description*
Operate mode	BOOL	na	na	na	na	RW	D	Controls if a light or dark condition indicates presence "0=Light signal indicates presence" "1=Dark signal indicates presence"
Output mode	BOOL	na	na	na	na	RW	D	The output mode inverts the definition of the output switch "0=Normally open" "1=Normally closed"
Fault	BOOL	na	na	na	na	RW	D	Application fault "0=Normal" "1=Faulted"
Sensitivity	USINT	na	0	1	0…255	RW	D	The sensitivity is the threshold setting of the detection device
Test	BOOL	na	na	na	na	RW	D	Toggles device between normal and test states "0=Command device state to normal" "1=Command device state to test"

PARAMETER ASSEMBLIES (MANUFACTURER-SPECIFIC)

Parameter assembly name: Advanced configure	Access: RW	Required: D

Byte	Bits: (0-7 for bit and byte constructions; 0-15 for word constructions)							
	7	6	5	4	3	2	1	0
	15	14	13	12	11	10	9	8
0	r	r	r	r	r	Output mode	Operate mode	Device mode

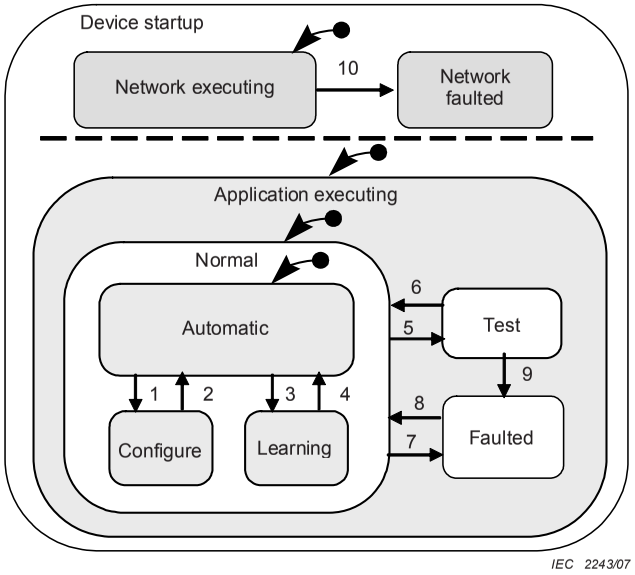
PARAMETER GROUPS (MANUFACTURER-SPECIFIC)

Group name	Group type	Number of members	Required	*Description*	*Additional information*
Problems	P	2	D	Parameters used to indicate problems	
Member names					
Alarm					
Fault					

DEVICE BEHAVIOUR (MANUFACTURER-SPECIFIC)

STATE MODEL (MANUFACTURER-SPECIFIC)

STATE CHART DIAGRAM



STATE TRANSITION TABLE

STATE NAME	STATE DESCRIPTION
Network executing	Initial concurrent state (with application executing) of the device upon power-up. The device is capable of responding to network commands for normal operation. The processor is running
Network faulted	The device is not available for normal communication operation and requires an external reset or power cycle to return to normal operation
Application executing	Initial concurrent state (with network executing) of the device upon power-up. The device is available for normal operation
Normal	The device is available for automatic operation
Automatic	“Presence” and “Alarm” parameter values are available to the network
Configure	When in this state, the device can be commanded by the “Operate mode”, “Output mode”, or “Sensitivity” parameters to alter its operation accordingly The device will not perform its normal sensing operations in this state – the “Presence” and “Alarm” parameter values should not be read by the network
Learning	In this state the device learns object sense level
Faulted	The device is not available for normal operation and requires an external reset or power cycle to return to normal operation
Test	The device does not perform its normal sensing operations. The “Presence” and “Alarm” parameter values are set to one (1)

TRANSITION	SOURCE STATE	TARGET STATE	EVENT (events and conditions that cause the transition to occur)
01	Automatic	Configure	"Device mode" parameter is commanded to change from zero (0) to one (1)
02	Configure	Automatic	"Device mode" parameter is commanded to change from one (1) to zero (0)
03	Automatic	Learning	Commanded when the device-adjusting button is pushed
04	Learning	Automatic	Commanded when the device-adjusting button is released
05	Normal	Test	"Test" parameter is commanded to change from zero (0) to one (1)
06	Test	Normal	"Test" parameter is commanded to change from one (1) to zero (0)
07	Normal	Faulted	An application fault is detected – "Fault" parameter is set to one (1)
08	Faulted	Normal	"Fault" parameter is commanded to change from one (1) to zero (0)
09	Test	Faulted	A fault is detected during test – "Fault" parameter is set to one (1)
10	Network executing	Network faulted	Communication port or processor failed

SERVICES (MANUFACTURER-SPECIFIC)

Service name	*Request parameter group*	*Response parameter group*	Required	*Description*	*Additional information*
Set configure mode			D	Allows to enter configure mode	
Set automatic mode			D	Allows to exit configure mode	
Enter test mode			D	Allows to enter test mode	
Exit test mode			D	Allows to exit test mode	

Figure B.4 – Example of a specific device profile created without using a root device profile

Annex C
(informative)

Profile creation guidelines

This annex defines the guidelines that should be used in the development of device profiles. It is recognised that no single set of rules will cover all cases of profile definition, therefore flexibility in definition of specific device profiles is implicit. The rules of use are a collection of general good engineering examples that should be applied, where appropriate, to the definition of a device profile.

IEC product committees and other profile developers should consider the complete product family in order that simple device profiles can be subsets of complex device profiles. Simpler device profiles should be subsets of the parameter list, state model and parameter assemblies of more complex device profiles, rather than redefining this information.

For example, a simple proximity switch device profile that reports the presence of an object can be derived from a diagnostic proximity switch device profile with margin alarm.

Figure C.1 is an example of a parameter assembly from a device profile for a simple proximity switch.

7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	Presence

IEC 2244/07

Figure C.1 – Simple proximity switch parameter assembly

Figure C.2 is an example of a parameter assembly for a diagnostic proximity switch device profile with margin alarm.

7	6	5	4	3	2	1	0
r	r	r	r	r	r	Margin alarm	Presence

IEC 2245/07

Figure C.2 – Diagnostic proximity switch parameter assembly

The parameters in multiple parameter assemblies should be in the same bit order and byte position, such that one parameter assembly is a subset of a larger parameter assembly.

Annex D
(informative)

Profile exchange language

D.1 General

This annex presents recommendations for the documentation and transfer of device profiles in the form of XML documents, using reference schemas. This device profile documentation can be used for the electronic distribution of profiles and is defined such that a software program can interpret the profile to make it usable in an automation application.

The XML syntax is defined by ISO/IEC 8879 and W3C recommendations (REC-xml-20001006, REC-xmlschema-1-20010502, and REC-xmlschema-2-20010502). The characters used in the profile exchange language are in accordance with ISO/IEC 4873.

The XML syntax allows the device profile to be extracted from the XML document and schema, so that the device profile can be read and viewed.

ISO 15745 has defined an application integration framework, i.e. a set of elements and rules for describing integration models and application interoperability profiles. ISO 15745-1 includes XML schemas for the master profile template and the main profile templates, including device profiles, while subsequent parts of ISO 15745 define technology specific templates.

Therefore, it is recommended to use ISO 15745 as a basis for any device profile template. Clause D.2 gives the correspondence between elements of the device profile template defined in Annex A and the schemas of ISO 15745-1. Clause D.3 provides examples of XML definitions for the various elements of the device profile template, in case no technology specific equivalent is available in the ISO 15745 series.

D.2 Correspondence with ISO 15745

Figure D.1 shows the general structure of an ISO 15745 based device profile.

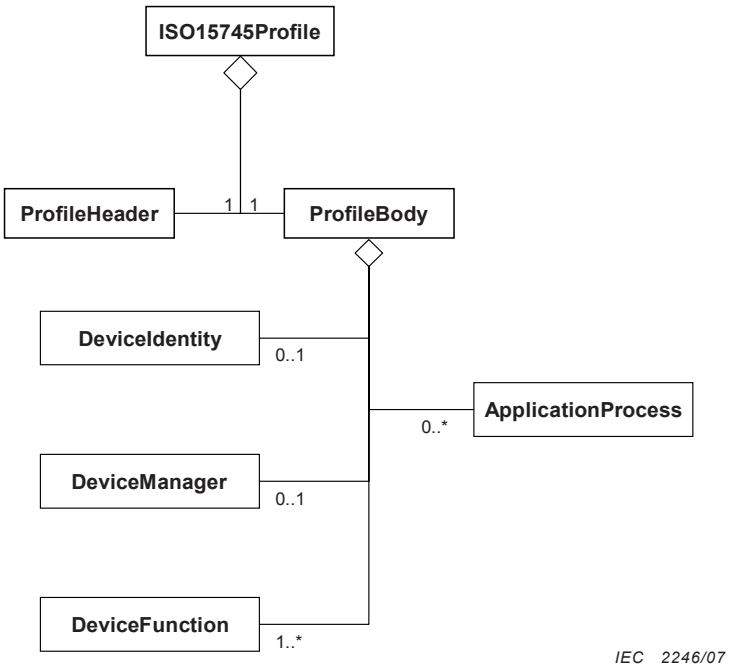


Figure D.1 – Overview of an ISO 15745 device profile

Table D.1 provides guidance on how to fill the attributes defined in the header of the ISO 15745-1 master template schema for a root device profile, based on the elements defined in the device profile template of Annex A.

Table D.1 – Mapping for a root device profile (ProfileHeader)

ISO 15745-1 ProfileHeader attributes	Contents (IEC 61915-1 device profile template elements)	Description
ProfileIdentification	Root device profile ID	In accordance with 5.2.2
ProfileRevision	Root device profile version	In accordance with 5.2.3
ProfileName	Device description	In accordance with 5.2.5
ProfileSource	“IEC” (or “IEC SC17B”, or organization’s name)	
ProfileClassID	“Device”	Fixed
ProfileDate	Root device profile release date	In accordance with 5.2.4
AdditionalInformation	—	In accordance with ISO 15745-1
ISO 15745Reference	ISO 15745Part= “1” by default ISO 15745Edition = “1” by default ProfileTechnology = “None” by default	In accordance with ISO 15745-1
IASInterfaceType	—	In accordance with ISO 15745-1

Table D.2 provides advice on how to fill the attributes defined in the body of the ISO 15745-1 master template schema for a root device profile, based on the elements defined in the device profile template of Annex A.

Table D.2 – Example mapping for a root device profile (ProfileBody)

ISO 15745-1 ProfileBody attributes	Contents (IEC 61915-1 device profile template elements)	Description
DeviceIdentity	—	May duplicate information in the ProfileHeader
DeviceManager	—	In accordance with ISO 15745-1
DeviceFunction	—	In accordance with ISO 15745-1
ApplicationProcess	Parameters	In accordance with 5.3
	Complex data types	In accordance with 5.4
	Parameter assemblies	In accordance with 5.5
	Parameter groups	In accordance with 5.6
	Functional elements	In accordance with 5.7
	State model (state transition table)	In accordance with 5.8
	Services	In accordance with 5.9
ExternalProfileHandle		In accordance with ISO 15745-1
NOTE Examples schema elements for these attributes are provided in Clause D.3.		

Table D.3 provides guidance on how to fill the attributes defined in the header of the ISO 15745-1 master template schema for a manufacturer's device profile, based on the elements defined in the device profile template of Annex A.

Table D.3 – Correspondence with ISO 15745 for a manufacturer's device profile

ISO 15745 attributes	Contents (IEC 61915-1 device profile template elements)	Description
ProfileIdentification	Manufacturer's device profile ID	In accordance with 6.2.2
ProfileRevision	Manufacturer's device profile version	In accordance with 6.2.4
ProfileName	Manufacturer's device profile description	In accordance with 6.2.3
ProfileSource	Manufacturer ID	In accordance with 6.2.6
ProfileClassID	"Device"	Fixed
ProfileDate	Manufacturer's device profile release date	In accordance with 6.2.5
AdditionalInformation		In accordance with ISO 15745-1
ISO 15745Reference	ISO 15745Part= "1" by default ISO 15745Edition = "1" by default ProfileTechnology = "None" by default	In accordance with ISO 15745-1
IASInterfaceType		In accordance with ISO 15745-1

Table D.4 provides advice on how to fill the attributes defined in the body of the ISO 15745-1 master template schema for a manufacturer's device profile, based on the elements defined in the device profile template of Annex A.

Table D.4 – Example mapping for a manufacturer's device profile (ProfileBody)

ISO 15745-1 ProfileBody attributes	Contents (IEC 61915-1 device profile template elements)	Description
DeviceIdentity	Information from the Manufacturer's device profile header: – Model compatibility – Software compatibility – Hardware compatibility – Profile type – Profile availability – Additional information Information from the Root device profile header: – Root device profile ID – Root device profile version – Root device profile release date – Device description	May also duplicate information in the ProfileHeader
DeviceManager	—	In accordance with ISO 15745-1
DeviceFunction	—	In accordance with ISO 15745-1
ApplicationProcess	Parameters Complex data types Parameter assemblies Parameter groups Functional elements State model (state transition table) Services	In accordance with 5.3 In accordance with 5.4 In accordance with 5.5 In accordance with 5.6 In accordance with 5.7 In accordance with 5.8 In accordance with 5.9
ExternalProfileHandle		In accordance with ISO 15745-1
NOTE Examples schema elements for these attributes are provided in Clause D.3.		

D.3 Device profile schema elements

The schema represents the structure of the device profile. Figure D.2 provides the main structure of an ISO 15745-based device profile, combined with examples of XML definitions for the various elements of the device profile template, in case no technology specific equivalent is available in the ISO 15745 series.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <!-- Target namespaces are not specified in this master template -->
    <xsd:annotation>
        <xsd:documentation>* ISO 15745 PROFILE STRUCTURE *</xsd:documentation>
    </xsd:annotation>
    <xsd:element name="ISO15745Profile">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="ProfileHeader"/>
                <xsd:element ref="ProfileBody"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:annotation>
        <xsd:documentation>* HEADER SECTION *</xsd:documentation>
    </xsd:annotation>
```

```
    <xsd:element name="ProfileHeader">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ProfileIdentification" type="xsd:string"/>
                <xsd:element name="ProfileRevision" type="xsd:string"/>
                <xsd:element name="ProfileName" type="xsd:string"/>
                <xsd:element name="ProfileSource" type="xsd:string"/>
                <xsd:element name="ProfileClassID" type="ProfileClassID_DataType" fixed="Device"/>
                <xsd:element name="ProfileDate" type="xsd:date" minOccurs="0"/>
                <xsd:element name="AdditionalInformation" type="xsd:anyURI" minOccurs="0"/>
                <xsd:element name="ISO15745Reference" type="ISO15745Reference_DataType"/>
                <xsd:element name="IASInterfaceType" type="IASInterface_DataType" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:annotation>
        <xsd:documentation>* BODY SECTION *</xsd:documentation>
    </xsd:annotation>
    <xsd:element name="ProfileBody">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="DeviceIdentity"/>
                <xsd:element ref="DeviceManager" minOccurs="0"/>
                <xsd:element ref="DeviceFunction" maxOccurs="unbounded"/>
                <xsd:element ref="ApplicationProcess" minOccurs="0"/>
                <xsd:element name="ExternalProfileHandle" type="ProfileHandle_DataType" minOccurs="0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:annotation>
        <xsd:documentation>* ISO 15745 HEADER DATA TYPES *</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleType name="ProfileClassID_DataType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="AIP"/>
            <xsd:enumeration value="Process"/>
            <xsd:enumeration value="InformationExchange"/>
            <xsd:enumeration value="Resource"/>
            <xsd:enumeration value="Device"/>
            <xsd:enumeration value="CommunicationNetwork"/>
            <xsd:enumeration value="Equipment"/>
            <xsd:enumeration value="Human"/>
            <xsd:enumeration value="Material"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="ISO15745Reference_DataType">
        <xsd:sequence>
            <xsd:element name="ISO15745Part" type="xsd:positiveInteger"/>
            <xsd:element name="ISO15745Edition" type="xsd:positiveInteger"/>
            <xsd:element name="ProfileTechnology" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="IASInterface_DataType">
        <xsd:union>
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="CSI"/>
                    <xsd:enumeration value="HCI"/>
                    <xsd:enumeration value="ISI"/>
                    <xsd:enumeration value="API"/>
                    <xsd:enumeration value="CMI"/>
                    <xsd:enumeration value="ESI"/>
                    <xsd:enumeration value="FSI"/>
                    <xsd:enumeration value="MTI"/>
                    <xsd:enumeration value="SEI"/>
                    <xsd:enumeration value="USI"/>
                </xsd:restriction>
            </xsd:simpleType>
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:length value="4"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:union>
    </xsd:simpleType>
    <xsd:annotation>
```

```
        <xsd:documentation>* ISO 15745 DEFINED DATA TYPES *</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType name="ProfileHandle_DataType">
        <xsd:sequence>
            <xsd:element name="ProfileIdentification" type="xsd:string"/>
            <xsd:element name="ProfileRevision" type="xsd:string"/>
            <xsd:element name="ProfileLocation" type="xsd:anyURI" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:annotation>
        <xsd:documentation>* IEC 61915 SPECIFIC ELEMENTS *</xsd:documentation>
        <xsd:documentation>* (Examples if no existing ISO 15745 technology-specific elements) *</xsd:documentation>
    </xsd:annotation>
    <xsd:element name="DeviceIdentity">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="RootDeviceProfileID" type="RootDeviceProfileIDType" minOccurs="0"/>
                <xsd:element name="RootDeviceProfileVersion" type="RootDeviceProfileVersionType" minOccurs="0"/>
                <xsd:element name="RootProfileReleaseDate" type="RootProfileReleaseDateType" minOccurs="0"/>
                <xsd:element name="DeviceDescription" type="xsd:string" minOccurs="0"/>
                <xsd:element name="ManufacturersDeviceProfileID" type="xsd:string"/>
                <xsd:element name="ManufacturersDeviceProfileDescription" type="xsd:string" minOccurs="0"/>
                <xsd:element name="ManufacturersDeviceProfileVersion">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:pattern value="V\d\d\d"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="ManufacturersDeviceProfileReleaseDate" type="xsd:date"/>
                <xsd:element name="ManufacturerID" type="xsd:string"/>
                <xsd:element name="ModelCompatibility" type="xsd:string" minOccurs="0"/>
                <xsd:element name="SoftwareCompatibility" type="xsd:string" minOccurs="0"/>
                <xsd:element name="HardwareCompatibility" type="xsd:string" minOccurs="0"/>
                <xsd:element name="ProfileType">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="Generic"/>
                            <xsd:enumeration value="Device"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element name="ProfileAvailability">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="Yes"/>
                            <xsd:enumeration value="No"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
                <xsd:element ref="AdditionalInformation" minOccurs="0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="DeviceManager">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="DeviceFunction">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="ApplicationProcess">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="RootDeviceProfileParameters">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element ref="Parameter" maxOccurs="unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
```

```
            </xsd:element>
            <xsd:element name="ManufacturersSpecificParameters">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="Parameter" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="RootDeviceProfileParameterAssemblies">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="ParameterAssembly" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="ManufacturersSpecificParameterAssemblies">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="ParameterAssembly" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="RootDeviceProfileParameterGroups">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="ParameterGroup" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="ManufacturersSpecificParameterGroups">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="ParameterGroup" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="RootDeviceProfileFunctionalElements">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="FunctionalElementList" minOccurs="0"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="ManufacturersSpecificFunctionalElements">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="FunctionalElementList" minOccurs="0"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="RootDeviceProfileStateModels">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="StateModel" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="RootDeviceProfileServices">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="Service" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="ManufacturersSpecificStateModels">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="StateModel" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="ManufacturersSpecificServices">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="Service" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
```

```
            </xsd:element>
            <xsd:element name="RootDeviceProfileComplexTypes">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="ComplexTypes" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="ManufacturersSpecificComplexTypes">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="ComplexTypes" minOccurs="0" maxOccurs="unbounded"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Parameter">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ParameterName" type="NameType"/>
            <xsd:element name="Length" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Units" type="unitsrangeType"/>
            <xsd:element name="Offset" type="scalingType"/>
            <xsd:element name="Multiplier" type="scalingType"/>
            <xsd:element name="Range" type="unitsrangeType"/>
            <xsd:element name="ParameterDescription" type="xsd:string" minOccurs="0"/>
            <xsd:element ref="ValuePair" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="Access" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="R"/>
                    <xsd:enumeration value="RW"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="DataType" type="ParameterDataType" use="required"/>
        <xsd:attribute name="Required" type="RequiredType" use="required"/>
        <xsd:attribute name="RootParameterFlag" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="Y"/>
                    <xsd:enumeration value="N"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<xsd:simpleType name="unitsrangeType">
    <xsd:union memberTypes="stringType naType"/>
</xsd:simpleType>
<xsd:simpleType name="stringType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="scalingType">
    <xsd:union memberTypes="floatType naType"/>
</xsd:simpleType>
<xsd:simpleType name="floatType">
    <xsd:restriction base="xsd:float"/>
</xsd:simpleType>
<xsd:simpleType name="ParameterDataType">
    <xsd:union memberTypes="SimpleDataType NameType"/>
</xsd:simpleType>
<!-- Parameter Assembly-->
<xsd:element name="ParameterAssembly">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ParameterAssemblyName" type="NameType"/>
            <xsd:element name="ParameterRef" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="ParameterID" type="NameType"/>
                        <xsd:element name="ParameterAssemblyStartByte" type="xsd:nonNegativeInteger"/>
                        <xsd:element name="ParameterAssemblyStartBit">
                            <xsd:simpleType>
```

```
                                <xsd:restriction base="xsd:nonNegativeInteger">
                                    <xsd:maxInclusive value="15"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:element>
                        <xsd:element name="ParameterAssemblyEndByte" type="xsd:nonNegativeInteger"/>
                        <xsd:element name="ParameterAssemblyEndBit">
                            <xsd:simpleType>
                                <xsd:restriction base="xsd:nonNegativeInteger">
                                    <xsd:maxInclusive value="15"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="Access" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="R"/>
                    <xsd:enumeration value="W"/>
                    <xsd:enumeration value="RW"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="Required" type="RequiredType" use="required"/>
        <xsd:attribute name="RootAssemblyFlag" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="Y"/>
                    <xsd:enumeration value="N"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<!--Parameter Group-->
<xsd:element name="ParameterGroup">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="GroupName" type="NameType"/>
            <xsd:element name="NumberOfMembers" type="xsd:positiveInteger"/>
            <xsd:element name="GroupDescription" type="xsd:string" minOccurs="0"/>
            <xsd:element ref="AdditionalInformation" minOccurs="0"/>
            <xsd:element name="MemberNames" type="NameType" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="GroupType" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="P"/>
                    <xsd:enumeration value="G"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="GroupUsedByOthersFlag" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="Y"/>
                    <xsd:enumeration value="N"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="Required" type="RequiredType" use="required"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="FunctionalElementList">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="FunctionalElement" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="FunctionalElement">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="FunctionalElementName" type="NameType"/>
```

```
            <xsd:element name="ParameterGroupName" type="NameType" minOccurs="0"/>
            <xsd:element name="StateModelName" type="NameType" minOccurs="0"/>
            <xsd:element name="FunctionalElementDescription" type="xsd:string"/>
            <xsd:element ref="FunctionalElement" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="Required" type="RequiredType" use="required"/>
    </xsd:complexType>
</xsd:element>
<!-- StateModel -->
<xsd:element name="StateTransitionTable">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="StateMapItem" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="StateName" type="NameType"/>
                        <xsd:element name="StateDescription" type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="TransitionMapItem" minOccurs="0" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="SourceState" type="NameType"/>
                        <xsd:element name="TargetState" type="NameType"/>
                        <xsd:element name="Event" type="xsd:string"/>
                    </xsd:sequence>
                    <xsd:attribute name="TransitionID" type="xsd:positiveInteger" use="required"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<!--Services-->
<xsd:element name="Service">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ServiceName" type="NameType"/>
            <xsd:element name="RequestParameterGroup" type="NameType" minOccurs="0"/>
            <xsd:element name="ResponseParameterGroup" type="NameType" minOccurs="0"/>
            <xsd:element name="ServiceDescription" type="xsd:string" minOccurs="0"/>
            <xsd:element ref="AdditionalInformation" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="Required" type="RequiredType" use="required"/>
        <xsd:attribute name="RootServiceFlag" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="Y"/>
                    <xsd:enumeration value="N"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<!-- Data types for RootDeviceProfileHeader elements-->
<xsd:simpleType name="RootIDType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="P\([A-Za-z0-9\-]+\s[A-Za-z0-9\-]+\)[1-9]\d\d\d\d"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RootDeviceProfileIDType">
    <xsd:union memberTypes="RootIDType naType"/>
</xsd:simpleType>
<xsd:simpleType name="RootVersionType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="V\d\d\d"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RootDeviceProfileVersionType">
    <xsd:union memberTypes="RootVersionType naType"/>
</xsd:simpleType>
<xsd:simpleType name="RootDateType">
    <xsd:restriction base="xsd:date"/>
</xsd:simpleType>
<xsd:simpleType name="RootProfileReleaseDateType">
    <xsd:union memberTypes="RootDateType naType"/>
</xsd:simpleType>
```

```
<!--Common elements-->
<xsd:simpleType name="SimpleDataType">
    <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="BOOL"/>
        <xsd:enumeration value="BYTE"/>
        <xsd:enumeration value="WORD"/>
        <xsd:enumeration value="DWORD"/>
        <xsd:enumeration value="LWORD"/>
        <xsd:enumeration value="SINT"/>
        <xsd:enumeration value="USINT"/>
        <xsd:enumeration value="INT"/>
        <xsd:enumeration value="UINT"/>
        <xsd:enumeration value="DINT"/>
        <xsd:enumeration value="UDINT"/>
        <xsd:enumeration value="LINT"/>
        <xsd:enumeration value="ULINT"/>
        <xsd:enumeration value="REAL"/>
        <xsd:enumeration value="LREAL"/>
        <xsd:enumeration value="STRING"/>
        <xsd:enumeration value="UNICODE"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="naType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="na"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="NameType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="32"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:element name="AdditionalInformation" type="xsd:string"/>
<xsd:element name="ComplexTypes">
    <xsd:complexType>
        <xsd:choice>
            <xsd:element name="ArrayType">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="DataTypeName" type="NameType"/>
                        <xsd:element name="NumberOfElements" type="xsd:unsignedInt"/>
                        <xsd:element name="ElementDataType">
                            <xsd:complexType>
                                <xsd:choice>
                                    <xsd:element name="SimpleTypes" type="SimpleDataType"/>
                                    <xsd:element name="DataTypeRef" type="NameType"/>
                                    <xsd:element ref="ComplexTypes"/>
                                </xsd:choice>
                            </xsd:complexType>
                        </xsd:element>
                        <xsd:element ref="AdditionalInformation" minOccurs="0"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="StructType">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="DataTypename" type="NameType"/>
                        <xsd:element name="NumberOfElements" type="xsd:unsignedInt"/>
                        <xsd:element ref="AdditionalInformation" minOccurs="0"/>
                        <xsd:element name="Element" maxOccurs="unbounded">
                            <xsd:complexType>
                                <xsd:sequence>
                                    <xsd:element name="ElementName" type="NameType" minOccurs="0"/>
                                    <xsd:element name="ElementDataType">
                                        <xsd:complexType>
                                            <xsd:choice>
                                                <xsd:element name="SimpleTypes" type="SimpleDataType"/>
                                                <xsd:element name="DataTypeRef" type="NameType"/>
                                                <xsd:element ref="ComplexTypes"/>
                                            </xsd:choice>
                                        </xsd:complexType>
                                    </xsd:element>
                                    <xsd:element ref="AdditionalInformation" minOccurs="0"/>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
```

```
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="EnumType">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="DataTypename" type="NameType"/>
                            <xsd:element name="NumberOfValues" type="xsd:unsignedInt"/>
                            <xsd:element name="ElementDataType" type="SimpleDataType" minOccurs="0"/>
                            <xsd:element ref="AdditionalInformation" minOccurs="0"/>
                            <xsd:element name="Element" maxOccurs="unbounded">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element ref="ValuePair" maxOccurs="unbounded"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="ValuePair">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ParameterValue" type="xsd:string"/>
                <xsd:element name="ValueDescription" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="StateModel">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="StateModelName" type="NameType" minOccurs="0"/>
                <xsd:element ref="StateTransitionTable" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:annotation>
        <xsd:documentation>* IEC 61915 SPECIFIC DATA TYPES *</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleType name="RequiredType">
        <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="M"/>
            <xsd:enumeration value="O"/>
            <xsd:enumeration value="m"/>
            <xsd:enumeration value="A"/>
            <xsd:enumeration value="D"/>
            <xsd:enumeration value="---"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

Figure D.2 – Device profile schema structure

Annex E
(informative)

Categories of parameters

E.1 General

The use of the parameter categories given below will ensure consistent terminology among different IEC product committees.

NOTE These categories may be used as a basis to define parameter groups in a device profile.

E.2 Control parameters

Control parameters are application information that is exchanged whilst the application is running with the control application, controller and other devices connected to the network. The control parameters are used by the application.

The control parameters may include application diagnostics information.

E.3 Management parameters

Management parameters include device diagnostic information, device state and device configuration information. Device diagnostic information is used to provide information about the device's condition. Device state parameters may be defined to provide information about the state of the device. Device configuration information is used to set up the device during application development, and when starting up the control equipment and the process.

NOTE 1 Some devices may have no configuration data.

NOTE 2 Some networks do not support configuration of devices across the communication system.

E.4 Parameter categories

Parameters can be classified into different categories such as the following:

1. Control parameters
    - Commands
        - Switching commands
        - Fault reset
        - Operating mode selection
    - Monitoring information
        - Operational signals
        - Fault signals
        - Warning signals
        - Maintenance signals
        - Network errors

- Measurements
  - Operational measurements
  - Maintenance measurements

2. Management parameters
   - Configuration
     - Operational levels
     - Fault levels
     - Warning levels
     - Maintenance levels
     - I/O assignment
     - Network configuration
   - Diagnostics
     - Operational diagnostics
     - Fault diagnostics
     - Warning diagnostics
     - Maintenance diagnostics
     - Network counters
     - Counters reset
   - Identification
     - Product identification
     - Manufacturer identification

Bibliography

IEC 60947 (all parts), *Low-voltage switchgear and controlgear*

NOTE Harmonized in EN 60947 series (not modified).

ISO/IEC 4873*, Information technology – ISO 8-bit code for information interchange -- Structure and rules for implementation* (available in English only)

ISO/IEC 8879, *Information processing – Text and office systems – Standard Generalised Markup Language (SGML)*

NOTE Harmonized as EN 28879:1990 (not modified).

ISO 15745-1, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*

REC-xml-20001006, *Extensible Markup Language (XML) 1.0 Second Edition – W3C Recommendation 6 October 2000*

REC-xmlschema-1-20010502*, XML Schema Part 1: Structures – W3C Recommendation 02 May 2001*

REC-xmlschema-2-20010502, *XML Schema Part 2: Datatypes – W3C Recommendation 02 May 2001*

———————

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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

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

Publication	Year	Title	EN/HD	Year
IEC 60559	1989	Binary floating-point arithmetic for microprocessor systems	HD 592 S1	1991
IEC 61131-3	2003	Programmable controllers - Part 3: Programming languages	EN 61131-3	2003
IEC/TR 62390	2005	Common automation device - Profile guideline	-	-
ISO 1000 A1	1992 1998	SI units and recommendations for the use of their multiples and of certain other units	-	-
ISO 15745	Series	Industrial automation systems and integration - Open systems application integration framework	-	-
ISO/IEC 10646	2003	Information technology - Universal multiple-octet coded character set (UCS)	-	-
ISO/IEC 19501	2005	Information technology - Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2	-	-

BSI — British Standards Institution

BSI is the independent national body responsible for preparing British Standards. It presents the UK view on standards in Europe and at the international level. It is incorporated by Royal Charter.

Revisions

British Standards are updated by amendment or revision. Users of British Standards should make sure that they possess the latest amendments or editions.

It is the constant aim of BSI to improve the quality of our products and services. We would be grateful if anyone finding an inaccuracy or ambiguity while using this British Standard would inform the Secretary of the technical committee responsible, the identity of which can be found on the inside front cover. Tel: +44 (0)20 8996 9000. Fax: +44 (0)20 8996 7400.

BSI offers members an individual updating service called PLUS which ensures that subscribers automatically receive the latest editions of standards.

Buying standards

Orders for all BSI, international and foreign standards publications should be addressed to Customer Services. Tel: +44 (0)20 8996 9001. Fax: +44 (0)20 8996 7001. Email: orders@bsi-global.com. Standards are also available from the BSI website at http://www.bsi-global.com.

In response to orders for international standards, it is BSI policy to supply the BSI implementation of those that have been published as British Standards, unless otherwise requested.

Information on standards

BSI provides a wide range of information on national, European and international standards through its Library and its Technical Help to Exporters Service. Various BSI electronic information services are also available which give details on all its products and services. Contact the Information Centre. Tel: +44 (0)20 8996 7111. Fax: +44 (0)20 8996 7048. Email: info@bsi-global.com.

Subscribing members of BSI are kept up to date with standards developments and receive substantial discounts on the purchase price of standards. For details of these and other benefits contact Membership Administration. Tel: +44 (0)20 8996 7002. Fax: +44 (0)20 8996 7001. Email: membership@bsi-global.com.

Information regarding online access to British Standards via British Standards Online can be found at http://www.bsi-global.com/bsonline.

Further information about BSI is available on the BSI website at http://www.bsi-global.com.

Copyright

BSI
389 Chiswick High Road
London
W4 4AL