17B/1022/CD

COMMITTEE DRAFT (CD)

IEC/TC or SC: SC 17B	Project number 61915 Ed.1	
Title: Low-voltage switchgear and controlgear	Date of circulation **1999-10-15**	Closing date for comments **2000-01-17**
Also of interest to the following committees SC 22G, SC 65C	Supersedes document 17B/845/CD and 17B/875A/CC	
Horizontal functions concerned: ☐ Safety ☐ EMC ☐ Environment ☐ Quality assurance		
Secretary: Marcel DELAPLACE	THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES. RECIPIENTS OF THIS DOCUMENT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.	

Title:
Draft IEC 61915: Low-voltage switchgear and controlgear – Profiles for networked industrial devices

(Titre) :
Projet 61915: Appareillage à basse tension – Profils pour les appareils industriels interconnectés

Introductory note

This draft IEC 61915, prepared by SC 17B/WG3, deals with general rules which will be used for the development of specific device profiles. Obviously, these specific device profiles will be prepared by appropriate SC 17B/WGs.

With their comments, National Committees are kindly invited to answer the following question:

Should specific device profiles be issued as annexes of this future IEC 61915 or as annex(es) of dedicated product standards IEC 60947-2, 60947-3, 60947-4-1, etc. ?

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

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

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61915 has been prepared by subcommittee 17B: Low-voltage switchgear and controlgear, of IEC technical committee 17: Switchgear and controlgear.

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

FDIS	Report on voting
17B/XX/FDIS	17B/XX/RVD

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

This publication has been drafted in accordance with ISO/IEC Directives, Part 3.

The committee has decided that this publication remains valid until…….. At this date, in accordance with the committee's decision, the publication will be

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

INTRODUCTION

The purpose of this standard is to provide a framework within which IEC Product Committees can define profiles for devices within their scope. It is intended that each Committee may define a ‘root profile’ for their devices from which manufacturers can define a specific profile for their products.

This standard will enable Committees to decide the amount of information which their products should make available through a controller-device interface in a common representation. It will give manufacturers a common framework to represent their controller-device interface capable devices and allow for manufacturer specific extensions to the device profiles.

It will also aid the writing of network independent application software. The use of this standard allows for the utilization of a profile exchange language for representation of the device profile facilitating the profile’s use by controller-device interface tools.

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

1 Scope

This standard defines a common representation of networked industrial devices and provides a template for documenting that representation independent of the controller-device interface used.

Product standard committees should use this International Standard in the preparation of root profiles for their products.

Product manufacturers should use the root device profiles developed by product committees together with the rules described in this standard in order to create manufacturer's device profiles to represent and describe their devices. This standard allows for manufacturers to make manufacturer-specific extensions to their manufacturer's device profile that are not included in the root device profiles developed by product standard committees.

This standard allows for the utilization of a profile exchange language for representation of the device profile information that will facilitate the profile's use by controller-device interface tools and application software.

NOTE – 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 normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 60559:1989, *Binary floating–point arithmetic for microprocessor system*

IEC 61131-3:1992, *Programmable controllers – Part 3: Programming languages*

ISO/IEC 4873:1991, *Information technology – ISO 8-bit code for information interchange – Structure and rules for implementation*

ISO 19502-1, *Unified Modelling Language*

ISO/IEC 8879:1986, *Information processing text and office systems Standard Generalised Mark-up Language (SGML)*

3 Definitions

For the purposes of this International Standard, the following definitions apply.

3.1
device profile
representation of a device in terms of its parameters, parameter assemblies and state model that describes the device's data and behaviour as viewed through a controller-device interface

3.2
device profile framework

set of rules contained in this standard for generating device profiles

3.3
manufacturer's device profile

profile containing the mandatory part and the relevant optional parts of the root device profile. It is declared by a manufacturer and may also include manufacturer-specific extensions

3.4
manufacturer-specific extensions

data contained within a manufacturer's device profile which is specified by a particular manufacturer and is in addition to the mandatory and optional parts of the root profile

3.5
parameter

data element that can be read from or written to a device

3.6
parameter assembly

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

3.7
root device profile

data required to be included in a manufacturer's device profile and which comprises mandatory and optional elements

NOTE – The root device profile is defined by the relevant IEC product committee.

4 Device profiles

4.1 General

A device profile is determined by the information (parameters and parameter assemblies) in, and behaviour (state model) of, the device and is independent of the controller-device interface. When designing an automation application, it is this profile that is used to represent the device as seen through the controller-device interface.

A root device profile can be created for each device (see table 1). It is recommended that product committees, where possible, use common parameters within a device type. The common parameters should use the same bit order and byte position, e.g. the “start” bit should be in the same position for each type of motor starter.

The root device profile shall define the format and content of any run-time, configuration, and device management information that is received from and/or sent to the device.

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

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

A device profile may also contain configuration information. Configuration information is data that is not normally exchanged at run time, such as device setup data. Some devices may have no configuration data. Some controller-device interfaces do not support configuration of devices across the communication system.

Data that does not change during device operation e.g. profile ID, may be stored either in the device or external to the device.

4.2 Mandatory information

The root device profile shall specify the mandatory information that can be exchanged with the device.

4.3 Optional information

A root device profile may also contain optional run time, configuration, or device management information

The root device profile shall define the format of the optional information.

4.4 Manufacturer-specific information

A root device profile may be extended by a manufacturer to include additional parameters, parameter assemblies and state models using this standard’s profile framework.

4.5 Profile template

4.5.1 General

Figure 1 shows the template for device profiles. It represents the formatting rules of the profile framework. A device profile shall contain a header, parameter list, state model, and an optional parameter assembly list. Once a template is completed with the device-specific information, it becomes a device profile.

Profile ID: P-SB-SN-PN Version: VAAA Profile classification: CFXX Date: YYYY-MM-DD

Automation Function Name: <AUTOMATION FUNCTION NAME> Device Type Name: <DEVICE TYPE NAME>

Description: <DESCRIPTION OF DEVICE>

PARAMETERS:

Parameter Name	Date Type	Engr. Units	Units Multiplier	Range	Access	Required	Description

TEMPLATE STATE MODEL:

STATE TRANSITION TABLE STATECHART

Parameter assembly Name: Access: Required:

Byte	Bits 7	6	5	4	3	2	1	0
0								
1								
2								

Parameter assembly Name: Access: Required:

Byte	Bits 7	6	5	4	3	2	1	0
0								

Figure 1 – Profile template

4.5.2 Template header

4.5.2.1 General

The template header shall contain the profile identifier, version, classification, creation date, automation function, device type and description information for the device profile.

4.5.2.2 Root device profile identifier

A root profile identifier shall be assigned for each device profile. The profile identifier includes the location in the appropriate product standard that defines the profile.

The format for a profile identifier consists of a text string of up to 9 characters using the format P-SB-SN-PN where:

P is always the capital letter ‘P’ followed by a dash ‘-‘;

SB is always a text string that identifies the standards body followed by a dash ‘-‘;

SN is always a text string that identifies the standard body’s standard number followed by a dash ‘-‘. The standard number may include multiple dashes;

PN is always a text string that identifies the subclause number of the relevant standard.

4.5.2.3 Profile version

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

- parameters; or
- the state model; or
- list of optional parameters; or
- list of optional parameter assemblies.

The initial release of a 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 of 4 characters using the format VAAA where:

V is always the capital letter “V”;

AAA is the version number.

4.5.2.4 Profile classification

A profile classification shall be assigned for each device profile. The profile classification includes automation function number and device type identifier currently defined in table 1. Only numerical characters shall be used for the automation function and device type.

Profile classification shall be assigned by the appropriate product committee using the automation function and device type identifier as listed in table 1.

The format for a profile classification consists of a text string of 5 characters using the format CFXX where:

C is always the capital letter ‘C’;

FF is the automation function specified in table 1;

XX is the device type identifier specified in table 1.

NOTE – An example of a profile header’s profile identifier, version and classification is *P-IEC-60947-5-2-F3 V004 C0202*, which means this is an IEC profile for a photoelectric sensor described by IEC Standard 60947-5-2. The product profile would be found in paragraph F3 of annex F. It is the fourth version or third revision. The device comes under the automation function “detection” and device type “photoelectric sensors” in table 1.

Table 1 – Profile automation function and device type identifiers

Automation function number (FF)	Device type identifier (XX)	Automation Function Name Device Type Name	Examples
00		Power Distribution	
	00	Switchboards	
	01	Circuit-Breakers	
	02	Power Monitoring	
	03	Distribution Panel	
	04-99	Reserved for future assignments	
01		Motor Control	
	00	Protection	
	01	Starters	Motor Starter
	02	Soft Starters	
	03	Drives	
	04	Axis Control	
	05	Motor Control Centre	
	06	Motor Monitoring	
	07-99	Reserved for future assignments	
02		Detection	
	00	Limit Switches	
	01	Inductive Sensors	Proximity Switch
	02	Photoelectric Sensors	Photoelectric Switch
	03	Capacitive Sensors	
	04	Ultrasonic Sensors	
	05	Magnetic Sensors	
	06-99	Reserved for future assignments	
03		Human-Machine Interfaces (HMI)	
	00	Push Buttons	
	01	Joysticks	
	02	Keypads	
	03	Pilot Lights	
	04	Stacklights	
	05	Displays	
	06	Combined Buttons/Lights	
	07	Operator Stations	Message Display, Label Printer
	08-99	Reserved for future assignments	
04		Logic / I/O Modules & Controllers	
	00	General Input	Discrete Input Block
	01	General Output	Discrete Output Block
	02	Combined Input/Output	Discrete Input/Output Block

	03	Actuators	Relay, Contactor, Solenoid Valves
	04	Timers	
	05	Scanners	Bar Code Reader
	06	PLC	
	07	Encoders	
	08-99	Reserved for future assignments	
05-99		Reserved for future use	

4.5.2.5 Release date

Each profile shall contain a version release date. The release date of the device profile shall be shown as YYYY-MM-DD where:

YYYY is the year;

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

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

4.5.2.6 Automation function name

Each profile header shall contain an automation function name, which is the relevant text string from column 3 of table 1 that describes the F automation function number.

4.5.2.7 Device type name

Each profile header shall contain a device type name, which is the relevant text string from column 3 of table 1 that describes the XX device type identifier.

4.5.2.8 Device description

Each profile header may contain a description of the device. The device description field is a text string which may be defined by the relevant product committee.

4.5.3 Template parameters

4.5.3.1 General

A device template shall contain one or more parameters. Parameters are indivisible data elements that represent device information that can be viewed through the network.

NOTE – Parameters can be classified as either:

Control (e.g., run-time data exchange)

Management (e.g., diagnostics, configuration).

4.5.3.2 Control parameters

Control parameters are application run-time information that is exchanged with the control application, controller and other devices connected to the controller-device interface. The parameters may include application diagnostics information. It is the information that is to control the application.

4.5.3.3 Management parameters

Management parameters are device diagnostics, state status and configuration information. Diagnostic information is used to provide information about the device's condition. Configuration information is used to set the device up during application development, control equipment and process start-up. A device state parameter assembly may be defined to provide information about the behavioural state of the device.

4.5.3.4 Parameter name (mandatory)

Parameter name shall be a maximum 32 character text string that contains the text name of the parameter.

4.5.3.5 Data type (mandatory)

Data type shall be a text string that contains the text name of the parameter data type. Valid data types are listed in table 3 and are derived from the data types defined in IEC 61131-3.

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

EXAMPLE: STRING10

Table 3 – Standard data types

Type Name	Description	Definition and Range
BOOL	Bit or Boolean	Represented by a 0 or a 1
BYTE	Byte	Bit string of 8 bits
WORD	Word	Bit string of 16 bits
DWORD	Double Word	Bit string of 32 bits
LWORD	Long Word	Bit string of 64 bits
SINT	Short Integer	-128 to 127
USINT	Unsigned Short Integer	0 to 255
INT	Single Integer	-32768 to 32767
UINT	Unsigned Integer	0 to 65534
DINT	Double Integer	-2^{31} to $2^{31}-1$
UDINT	Unsigned Double Integer	0 to $2^{32}-1$
LINT	Long Integer	-2^{63} to $2^{63}-1$
ULINT	Unsigned Long Integer	0 to $2^{64}-1$
REAL	Single Real	IEC 60559 basic single floating point. Allows an approximate range of - $1.2*10^{-38}$ to $1.8*10^{38}$
LREAL	Double Real	IEC 60559 basic double floating point. Allows an approximate range of - $1.2*10^{-308}$ to $1.8*10^{308}$
STRING	Text String	1 byte per character
UNICODE	Unicode	2 bytes per character

4.5.3.6 Engineering units (mandatory)

Engineering unit shall be a text field that specifies the engineering units of the parameter in SI units. Engineering units help define how the parameter value should be interpreted. When no engineering units are defined or required, the text string “<na>” (not applicable) shall be used.

4.5.3.7 Multiplier (mandatory)

Unit multiplier shall be a number that defines how the parameter value is interpreted. It is only used to define the resolution of the engineering units. When no units are specified, the unit multiplier shall be the value 1.

The unit multiplier is a scalar (floating point or integer) number, without units.

EXAMPLE: a parameter of value 100 with engineering units of ^{o}C and a unit multiplier of 0,1 corresponds to 10,0 ^{o}C

4.5.3.8 Range (mandatory)

Range shall specify the valid range of data values of the parameter. The range value may be more limited than the range specified by the parameter type e.g., the parameter type may be BYTE (0…255), while the valid range may only be 40…200. The range shall be inclusive of the specified values and any exceptions shall be described in the description field.

The range shall be specified as the minimum value followed by an ellipsis (…) followed by the maximum value with no spaces. When no range is defined or required the text string “<na>” (not applicable) shall be used.

EXAMPLE: 40…200

NOTE – Ellipsis is a character that contains three dots as in (…).

4.5.3.9 Access (mandatory)

Access shall specify the access to the parameter allowed through the controller-device interface. Access shall be specified as:

Read (for parameters readable from the connected device);

Write (for parameters writable to the connected device).

NOTE – A parameter that is Write accessible should also be Read accessible.

4.5.3.10 Required (mandatory)

A device profile shall specify whether the device supports the parameter. The template’s “Required” field shall be specified as:

No (for parameters not required to be supported);

Yes (for parameters required to be supported).

4.5.3.11 Description (optional)

Description shall contain a text description of the parameter and/or its use. It may also contain a description of any specific values the parameter may have.

EXAMPLE: 0 = No object sensed

1 = Object sensed

4.5.4 Template state model

4.5.4.1 General

All profiles shall define a state model. A state model shall include a state chart 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 the application can influence the state of the device or when the internal state of the device affects its observable behaviour.

Any device state that is visible through the network shall be defined.

EXAMPLES: on, off, target present, test, normal operation, or error condition.

Controller-device interface specific states are outside the scope of this standard.

NOTE – The state models defined in annex B may be used as the basis for a specific device state model.

4.5.4.2 State chart

State charts shall be used to describe the dynamic behaviour of the devices defined. Unified Modelling Language (UML) shall be the state model language used by device profiles.

An example of a state chart is shown in figure 2.

(editor's note: this drawing will be changed to UML format)

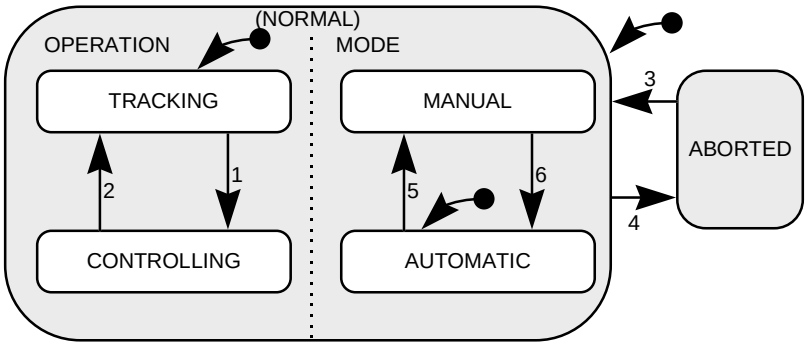


Figure 2 – Example state chart

4.5.4.3 State name

State name shall be a maximum 32 character text string that contains the text name of the state.

4.5.4.4 State description

Description shall contain a text description of the state and/or its use. It may also contain a description of the value of the state.

EXAMPLE: 0 = Not in state
1 = In state

4.5.4.5 State transition

A state transition tables shall describe the nature of each state and describe the rules of each transition between states and events that cause the transition to take place. State transition tables complement and describe the state chart diagrams.

The state transition tables shall be used in conjunction with the state chart. They define the events that cause state transitions. The events may be commands sent through the network, via parameters or parameter assembly values, or may be internally generated events or conditions detected by the connected device.

Figure 3 illustrates a state transition table and describes transition conditions for the concurrent state chart in figure 2.

State	***Description***
ABORTED	The device has failed (and not the network interface) and is not available to perform its normal operation.
NORMAL	The substate of the device, manual or automatic.
MANUAL	The control value setpoint is defined externally from the device.
AUTOMATIC	The control value is calculated by the device.
TRACKING	The device is not controlling its output, but is tracking the control variable for bumpless transfer to CONTROLLING.
CONTROLLING	The device is controlling its outputs.

Transition	***From State***	***To State***	***Transition Condition(s)***
1	TRACKING	CONTROLLING	Device commanded to start controlling.
2	CONTROLLING	TRACKING	Device commanded to stop controlling.
3	ABORTED	NORMAL	Network reset of device
4	NORMAL	ABORTED	Device detected abnormal condition.
5	AUTOMATIC	MANUAL	Device commanded to manual mode
6	MANUAL	AUTOMATIC	Device commanded to automatic mode

Figure 3 – State transition table

4.5.4.6 State transition description

The transition description field describes the events and conditions that cause the transition to occur.

4.5.5 Template parameter assemblies

4.5.5.1 General

Parameter assemblies shall define the data structures for information exchange of one or more parameters and shall be independent of the operating system and controller-device interface. Parameter assemblies do not necessarily represent the ordering of the data within the controller-device interface message. Parameter assemblies shall specify the bit, byte ordering of data, and the specific bit ordering within the bytes. They shall also specify the length in bits or length in bytes of the data. Parameter assemblies shall either read information from or write information to a device. A profile may contain multiple parameter assemblies. Individual parameters may be represented in multiple parameter assemblies within a profile.

Parameter assemblies shall be optional. Parameter assembly information listed as mandatory shall be provided when parameter assemblies are included in manufacturer' device profile.

Within each parameter assembly, those bits or bytes which are not part of the parameter assembly's data shall be labelled as (<r>), for *reserved* and shall be considered undefined. Bits and bytes that are labelled reserved are undefined data. Reserved fields shall be set to zero (0), for write parameter assemblies.

4.5.5.2 Name (mandatory)

Parameter assembly name shall be a maximum 32 character text string that contains a descriptive text name of the parameter assembly.

4.5.5.3 Access (mandatory)

Access shall specify the access to the parameter assembly allowed through the controller-device interface. Access shall be specified as:

Read (for parameter assemblies readable from the connected device);

Write (for parameter assemblies writable to the connected device).

4.5.5.4 Required (mandatory)

A root device profile shall specify that device support for the parameter assembly is optional. The template's "Required" field shall be specified as "No".

For an parameter assembly, not all parameters need to be provided. Non-provided parameters shall be set to null values such as zero or a blank string, or a "not provided" value. The device profile shall define what parameter values that indicate the "not provided" value for the parameter.

4.5.5.5 Parameter reference (mandatory)

Parameters within a parameter assembly shall contain the name of the parameters. 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.

4.5.5.6 Bit and byte ordering (mandatory)

Bit ordering and byte ordering in an parameter assembly shall start at zero. Bits shall be defined such that the least significant bit in a byte is bit 0 and the most significant bit is bit 7.

4.5.5.7 Multi-byte ordering

Multiple byte parameters within an parameter assembly shall start on byte boundaries, bit zero. Multiple byte data types within an parameter assembly shall utilise the ordering according to table 4.

Table 4 - Standard data type sizes

Data Type Keyword	Size of Type	Byte Ordering
WORD	Two (2) bytes	Lower byte first Upper byte last
DWORD	Four (4) bytes	Lower byte first Upper byte last
LWORD	Eight (8) bytes	Lower byte first Upper byte last
INT	Two (2) bytes	Lower byte first Upper byte last
UINT	Two (2) bytes	Lower byte first Upper byte last
DINT	Four (4) bytes	Lower byte first Upper byte last
UDINT	Four (4) bytes	Lower byte first Upper byte last
LINT	Eight (8) bytes	Lower byte first Upper byte last
ULINT	Eight (8) bytes	Lower byte first Upper byte last
REAL	Four (4) bytes	Mantissa in the lower bytes, the exponent and sign in the upper byte.
LREAL	Eight (8) bytes	Mantissa in the lower bytes, the exponent and sign in the upper bytes.
STRING	Size specified by parameter	First character in string in the lowest byte, last character in string the highest byte
UNICODE	Size specified by parameter	First character in string in the lowest byte, last character in string the highest byte

4.6 Rules of use

4.6.1 General

This clause defines the general rules that shall be used in the development of device profiles. It is recognized 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 shall be applied, where appropriate, to the definition of a device profile.

4.6.2 Inheritance

Inheritance of parameters, state models, and parameter assemblies shall not be permitted. Each profile shall stand alone without reference to other profiles. Device profile development supports multiple inheritance definitions, even though device profiles do not.

However, inheritance may be used to develop profiles. Inheritance is a way to simplify the use of devices. Devices that are extensions of simpler devices may extend the parameter list, state model, and parameter assemblies rather than redefining this information. The new profile parameter assemblies would use the reserved bits and bytes defined in the simpler device. For example, an illuminated push button with light sensing feedback may be considered as derived from a simple push button that only contains the expected status of the button. The extra information could be placed into one of the reserved bits of the status parameter assembly as illustrated below in figure 4 and figure 5.

Figure 4 is an example of an parameter assembly for a simple illuminated push button with light.

7	6	5	4	3	2	1	0
<r>	<r>	<r>	<r>	<r>	<r>	<r>	Button status

Figure 4 – Simple push button parameter assembly

Figure 5 is an example of an parameter assembly for simple push button with light and feedback.

7	6	5	4	3	2	1	0
<r>	<r>	<r>	<r>	<r>	<r>	Feedback	Button status

Figure 5 – Push button parameter assembly with light and feedback

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

4.7 Common device information

A manufacturer's device profile shall contain common device information as shown in figure 6. This information shall be stored in either the device or an external file.

Manufacturer Device Profile ID: Version: VAAA Profile classification: CFFXX Date: YYYY-MM-DD

Automation Function Name:<AUTOMATION FUNCTION NAME> Device Type Name:<DEVICE TYPE NAME>

Description: <DESCRIPTION OF DEVICE>

PARAMETERS:

Parameter Name	Date Type	Engr Units	Unit	Range	Access	Required	Description
Root Device Profile ID	String 9	<na>	<na>	<na>	Read	Yes	Defines the device profile to which this device adheres. Formatted according to the definition in this standard.
Version Number	String 4	<na>	<na>	<na>	Read	Yes	Identifies the version of the device profile to which the device adheres. Formatted according to the definition in this standard.
Manufacturer ID	String 32	<na>	<na>	<na>	Read	Yes	Designates the manufacturer of the device. Each manufacturer is responsible for defining this variable. It is unique to a manufacturer and is the same for all devices they provide. Usually the trademarked company name.
Model Number	String 32	<na>	<na>	<na>	Read	Yes	Designates the basic model identification number, defined by the manufacturer.
Software Revision	String 8	<na>	<na>	<na>	Read	Yes	Designates the software or firmware version of microprocessor code that is contained in the device, as defined by the manufacturer.
Hardware Revision	String 8	<na>	<na>	<na>	Read	No	Designates the version of the device, excluding the microprocessor code, as defined by the manufacturer.
Serial Number	String 32	<na>	<na>	<na>	Read	No	Designates the number or string, defined and assigned by the manufacturer that uniquely identifies each individual device or batch of devices produced.
Additional Info	String 64	<na>	<na>	<na>	Read	No	Designates any additional device information.

Parameter assembly Name: Device Management Information Access: Read Required: No

Byte	Parameter Name
0-23	Profile ID
24-27	Version Number
28-59	Manufacturer ID
60-91	Model Number
92-99	Software Revision
100-107	Hardware Revision
108-139	Serial Number

Parameter assembly Name: Additional Information Access: READ Required: NO

Byte	Parameter Name
0-63	Additional Info

Figure 6 – Example of common profile information

Annex A
(informative)

Device profile example

A.1 Example photoelectric device profile

The following profile example represents a device with a single running operating state (see figure A.1). This is typical of a simple detecting devices. No common profile information is included.

Profile ID : *P-IEC-60947-5-2-F1* Version: V000 Profile classification: *C0202* Date: 1996-05-07

Automation function: Detection Device type: Photoelectric sensor
Description: Photoelectric switch with alarm and light sensing control configuration

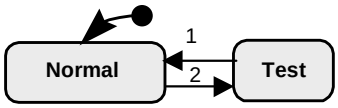
PARAMETERS:

Parameter Name	Data Type	Engr. Units	Unit Multiplier	Range	Access	Required	Description
Presence	BOOL	<na>	<na>	0 .. 1	Read	Yes	Indicates if the photoelectric switch has detected the presence of an object 0 = No presence sensed 1 = Presence sensed
Alarm	BOOL	<na>	<na>	0 .. 1	Read	Yes	Alarm condition is indicated by a marginal or failed sensing condition. 0 = Alarm condition detected 1 = No alarm condition detected
Mode Control	BOOL	<na>	<na>	0 .. 1	Write	Yes	Controls if a light or dark condition indicates presence 0 = Light signal indicates presence 1 = Dark signal indicates presence
Test	BOOL	<na>	<na>	0 .. 1	Read	Yes	Toggles device between Normal and Test state. 0 = Command device state to Normal 1 = Command device state to Test

TEMPLATE STATE MODEL:
TRANSACTION RULES FOR STATECHART

State	*Description*
Normal	The device is performing its normal sensing operations. Parameters indicate their sensing and alarm values.
Test	The device is not performing its normal sensing operations. Presence and Alarm parameters are set to one (1).

STATECHART



Transition	*From State*	*To State*	*Description*
1	NORMAL	TEST	Commanded by Test parameter set to one (1).
2	TEST	NORMAL	Commanded by Test parameter set to zero (0).

Parameter assembly Name: Presence Access: Read Required: No

Byte	Bits 7	6	5	4	3	2	1	0
0	<r>	<r>	<r>	<r>	<r>	<r>	Alarm	Presence

Parameter assembly Number: Configure Access: Write Required: No

Byte	Bits: 7	6	5	4	3	2	1	0
0	<r>	<r>	<r>	<r>	<r>	<r>	Mode Control	<r>

Figure A.1 – Photoelectric device profile example

A.2 Example motor starter device profile

(Profile under development awaiting contribution from IEC SC17B WG2.)

Annex B
(informative)

Profile exchange language

B.1 General

This annex presents a recommended syntax for the documentation and transfer of a device profile in the form of an XML Document and an XML Document Type Definition (DTD) which the XML Document references. This syntax is defined by ISO/IEC 8879. 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 following information can be defined using this recommended syntax:

- the information contained in a root device profile;
- optional parameter assemblies, and parameters in optional parameter assemblies which are supported by a manufacturer's device profiles and its controller-device interface implementation;
- manufacturer-specific extensions of a device profile including additional parameters, parameter assemblies, or state information which are outside the scope of the root device profile.

The XML syntax is defined such that it is easy for software to parse, which allows the device profile to be recreated from the XML Document. The recreated device profile is possible to read and view as a simple text file.

The XML Document and Document Type Definition file may be contained within a disk file, transferred electronically, or otherwise shared. A text file may contain multiple profile definitions and option definitions and the format allows these to be differentiated in the file.

This standard will not define the XML syntax and formatting. This information can be found in ISO/IEC 8879. The characters used in the profile exchange language are in accordance with ISO/IEC 4873. Each sentence of the profile exchange language is enclosed in the '<' and '>' characters. Only the device profile and profile framework specific entities will be define.

B.2 Device profile document type definition

The document type definition provides the structure of the device profile. The XML document type definition for device profiles defined in this standard is shown in figure B.1. The device profile DTD defines the elements of device profile document. The first sentence of the DTD gives the file name for referencing in the XML Document and for use by an XML parser.

```
<!--profileexchangelanguage.dtd -->

<!ELEMENT Device
      (Header,Parameter+,ParameterValue+,Assembly+,AssemblyMap+,
      StateName+,(StateTransition+)?,(StateID+)?,
      (StateTransitionCmd+)?,OptionHeader?)>
<!ELEMENT Header
      (ProfileID,Version,ProfileClassification,ReleaseDate,
      AutomationFunctionName,DeviceTypeName,Description?)>
<!ELEMENT Parameter
      (ParmeterName,DataType,EngineeringUnits,UnitMultiplier,MinimumRa
      ngeValue,MaximumRangeValue,Access,Required,Description?)>
<!ELEMENT ParameterValue    (ParameterValue,ValueDescription)>
<!ELEMENT Assembly   (AssemblyName,Access,Required)>
<!ELEMENT AssemblyMap
      (AssemblyName,ParameterName,StartingByte,StartingBit,EndingByte,
      EndingBit)>
<!ELEMENT StateName
     (StateName,ParentStateName,InitialState,Description)>
<!ELEMENT StateTransition
     (TransitionNum,FromStateName,ToStateName,TransitionCondition)>
<!ELEMENT StateID           (StateName,ParameterName,Value)>
<!ELEMENT StateTransitionCmd(TransitionNum,ParameterName,Value)>
<!ELEMENT OptionHeader
      (HeaderDefinition,(AssemblySupported+)?,(ParameterSupported+)?)>
<!ELEMENT HeaderDefinition
     (ProfileID,VersionNum,VendorID,VendorVersionNum,Description)>
<!ELEMENT AssemblySupported AssemblyName>
```

Figure B.1 – Device profile DTD

B.2.1 Device entity

The device entity defines the main entities or parts of the device profile. The main entities are enclosed in parenthesis following the device entity. The Main entities of the device profile are:

- Header
- Parameter
- Parameter Value
- Parameter Assembly
- Parameter Assembly Map
- State Name
- State Transition
- State ID
- State Transition Cmd
- Option Header

B.2.1.1 Header entity

The main entity header includes the following subordinate entities:

- Profile ID. See 4.5.2.2;
- Version; See 4.5.2.3;
- Profile Classification. See 4.5.2.4;
- Release Date. See 4.5.2.5;
- Automation Function Name. See 4.5.2.6;
- Device Type Name. See 4.5.2.7;
- Description. See 4.5.2.8.

B.2.1.2 Parameter entity

The main entity parameter includes the following subordinate entities:

- Parameter Name. See 4.5.3.4;
- Data Type. See 4.5.3.5;
- Engineering Units. See 4.5.3.6;
- Unit Multiplier. See 4.5.3.7;
- Minimum Range Value. See 4.5.3.8;
- Maximum Range Value. See 4.5.3.8;
- Access. See 4.5.3.9;
- Required. See 4.5.3.10;
- Description. See 4.5.3.11.

B.2.1.3 Parameter value entity

The main entity parameter value includes the following subordinate entities:

- Parameter Value. See 4.5.3.11;
- Value Description. See 4.5.3.11.

B.2.1.4 Parameter assembly entity

The main entity parameter assembly includes the following subordinate entities:

- Parameter Assembly Name. See 4.5.5.2;
- Access. See 4.5.5.3;
- Required. See 4.5.5.4.

B.2.1.5 Parameter assembly map entity

The main entity parameter assembly map includes the following subordinate entities:

- Parameter Assembly Name. See 4.5.5.2;
- Parameter Name. See 4.5.3.4;
- Starting Byte. See 4.5.5.1;
- Starting Bit. See 4.5.5.1;
- Ending Byte. See 4.5.5.1;
- Ending Bit. See 4.5.5.1.

B.2.1.6 State name entity

The main entity state name map includes the following subordinate entities:

- State Name. See 4.5.4.3;
- Parent State Name. See 4.5.4.2;
- Initial State. See 4.5.4.2;
- Description. See 4.5.4.5.

B.2.1.7 State ID entity

The main entity state ID defines the mapping from a parameter to a state, if that mapping is defined in the profile. It contains the state name that is mapped, the parameter name that the state is mapped to, and the value of the parameter that indicates that the device is in the state. The entity state ID includes the following subordinate entities:

- State Name. See 4.5.4.3;
- Parameter Name. See 4.5.3.4;
- Value. Zero (0) = Not in state. One (1) = Device is in state.

B.2.1.8 State transition entity

The main entity state transition defines a state transition. It contains the number of the transition, the state the transition is from, the state the transition goes to, and the description of the transition conditions necessary for the state transition to occur. The entity state transition includes the following subordinate entities:

- Transition Num. See 4.5.4.2;
- From State Name. See 4.5.4.5;
- To State Name. See 4.5.4.5;
- Transition Condition. See 4.5.4.6.

B.2.1.9 State transition cmd entity

The main entity state transition defines the mapping from a parameter to a transition if that mapping is defined in the device profile. The appropriate parameter value commands the state transition to occur. The entity state transition includes the following subordinate entities:

- Transition Num. See 4.5.4.2.
- Parameter Name. See 4.5.3.4
- Value. Zero (0) = No transition is to occur. One (1) = Transition is to occur.

B.2.1.10 Option header entity

The additional main entity option header includes the following subordinate entities:

- Header Definition
- Parameter Assembly Supported
- Parameter Supported

B.2.1.10.1 Header definition entity

The additional main entity option header, header definition entity includes the following subordinate entities:

- Profile ID. See 4.5.2.2;
- Version. See 4.5.2.3;
- Vendor ID;
- Vendor Version;
- Description. See 4.5.2.8.

The vendor ID, vendor version and description are provided by the vendor. The description may use the definition in 4.5.2.8 as an example.

B.2.1.10.2 Parameter supported

The additional main entity option header, parameter supported entity defines an additional and/or an optional parameter that is supported within an optional parameter assembly. See 4.5.5.4 for parameter assembly description. When a vendor does not support all parameters within an parameter assembly the parameter assembly name and the parameters of the parameter assembly that are supported shall be listed:

- Parameter Assembly Name. See 4.5.5.2;
- Parameter Name. See 4.5.3.4.

B.2.1.10.3 Parameter assembly supported

The additional main entity option header, parameter assembly supported entity defines an additional and/or optional parameter assembly that is supported in its entirety includes the following subordinate entity:

- Parameter Assembly Name. See 4.5.5.2;

B.3 Device profile document

The document provides the actual device profile. It uses the referenced DTD in figure B.1 for the document structure and provides actual device profile entity data. Multiple instances of all device profile entities will be included in the device profile document.

The XML document for the example photoelectric device profile in B.1.1 is shown in figure B.2. All device profile entity data fields are filled with the actual photoelectric root profile device information in B.1.1. The device in B.1.1 supports a root device profile and does not support optional parameter assemblies and parameters. The device profile document will not contain an option header entity since the device in B.1.1 does not contain optional information or manufacturer-specific extensions.

```
<?xml version="1.0" ?>

<!DOCTYPE example photoelectric sensor SYSTEM
"profileexchangelanguage.dtd">

<example photoelectric sensor>
<Device>
<Header>
<ProfileID>P-IEC-947-5-2-F1</ProfileID>
<Version>V000</Version>
<ProfileClassification>C202</ProfileClassification>
```

```
<ReleaseDate>1996-12-01</ReleaseDate>
<AutomationFunctionName>Detection</AutomationFunctionName>
<DeviceTypeName>Photoelectric sensor</DeviceTypeName>
<Description>Photoelectric switch with alarm and light sensing control
configuration</Description>
</Header>
<Parameter>
<ParameterName>Presence</ParameterName>
<DataType>BOOL</DataType>
<EngineeringUnits>na</EngineeringUnits>
<UnitMultiplier>na</UnitMultiplier>
<MinimumRangeValue>0</MinimumRangeValue>
<MaximumRangeValue>1</MaximumRangeValue>
<Access>Read</Access>
<Required>Yes</Required>
<Description>Indicates if the photoelectric switch has detected the
presence of an object</Description>
</Parameter>
<Parameter>
<ParameterName>Alarm</ParameterName>
<DataType>BOOL</DataType>
<EngineeringUnits>na</EngineeringUnits>
<UnitMultiplier>na</UnitMultiplier>
<MinimumRangeValue>0</MinimumRangeValue>
<MaximumRangeValue>1</MaximumRangeValue>
<Access>Read</Access>
<Required>Yes</Required>
<Description>Alarm condition is indicated by a marginal or failed sensing
condition</Description>
</Parameter>
<ParameterName>Mode Control</ParameterName>
<DataType>BOOL</DataType>
<EngineeringUnits>na</EngineeringUnits>
<UnitMultiplier>na</UnitMultiplier>
<MinimumRangeValue>0</MinimumRangeValue>
<MaximumRangeValue>1</MaximumRangeValue>
<Access>Write</Access>
<Required>Yes</Required>
<Description>Controls if a light or dark condition indicates
presence</Description>
</Parameter>
<ParameterValue>
<ParameterName>Presence</ParameterName>
<ParameterValue>0</ParameterValue>
<ValueDescription>No presence sensed</ValueDescription>
</ParameterValue>
<ParameterValue>
<ParameterName>Presence</ParameterName>
<ParameterValue>1</ParameterValue>
<ValueDescription>Presence sensed</ValueDescription>
</ParameterValue>
<ParameterValue>
<ParameterName>Alarm</ParameterName>
<ParameterValue>0</ParameterValue>
<ValueDescription>Alarm condition detected</ValueDescription>
</ParameterValue>
<ParameterValue>
<ParameterName>Alarm</ParameterName>
<ParameterValue>1</ParameterValue>
<ValueDescription>No alarm condition detected</ValueDescription>
</ParameterValue>
<ParameterValue>
<ParameterName>Mode Control</ParameterName>
```

```
<ParameterValue>0</ParameterValue>
<ValueDescription>Light signal indicates presence</ValueDescription>
</ParameterValue>
<ParameterValue>
<ParameterName>Mode Control</ParameterName>
<ParameterValue>1</ParameterValue>
<ValueDescription>Dark signal indicates presence</ValueDescription>
</ParameterValue>
<ParameterAssembly>
<ParameterAssemblyName>Presence</ParameterAssemblyName>
<Access>Read</Access>
<Required>No</Required>
</ParameterAssembly>
<ParameterAssembly>
<ParameterAssemblyName>Configure</ParameterAssemblyName>
<Access>Write</Access>
<Required>No</Required>
</ParameterAssembly>
<ParameterAssemblyMap>
<ParameterAssemblyName>Presence</ParameterAssemblyName>
<ParameterName>Presence</ParameterName>
<StartingByte>0</StartingByte>
<StartingBit>0</StartingBit>
<EndingByte>0</EndingByte>
<EndingBit>0</EndingBit>
</ParameterAssemblyMap>
<ParameterAssemblyMap>
<ParameterAssemblyName>Presence</ParameterAssemblyName>
<ParameterName>Alarm</ParameterName>
<StartingByte>0</StartingByte>
<StartingBit>1</StartingBit>
<EndingByte>0</EndingByte>
<EndingBit>1</EndingBit>
</ParameterAssemblyMap>
<ParameterAssemblyMap>
<ParameterAssemblyName>Configure</ParameterAssemblyName>
<ParameterName>Mode Control</ParameterName>
<StartingByte>0</StartingByte>
<StartingBit>1</StartingBit>
<EndingByte>0</EndingByte>
<EndingBit>1</EndingBit>
</ParameterAssemblyMap>
<StateName>
<StateName>Normal</</StateName>
<ParentStateName></ParentStateName>
<InitialState>Yes</InitialState>
<Description>The device is performing its normal operations. Parameters
indicate their sensing and alarm values.</Description>
</StateName>
<StateName>
<StateName>Test</</StateName>
<ParentStateName></ParentStateName>
<InitialState>No</InitialState>
<Description>The device is not performing its normal operations. Presence
and Alarm parameters are set to one (1).</Description>
</StateName>
<StateTransition>
<TransitionNum>1</TransitionNum>
<FromStateName>Normal</FromStateName>
<ToStateName>Test</ToStateName>
<Description>Commanded by Test parameter set to one (1)</Description>
</StateTransition>
<StateTransition>
```

```
<TransitionNum>2</TransitionNum>
<FromStateName>Test</FromStateName>
<ToStateName>Normal</ToStateName>
<Description>Commanded by Test parameter set to zero (0)</Description>
</StateTransition>
<StateTransitionCmd>
<TransitionNum>1</TransitionNum>
<ParameterName>Test</ParameterName>
<Value>0</Value>
</StateTransitionCmd>
<StateTransitionCmd>
<TransitionNum>2</TransitionNum>
<ParameterName>Test</ParameterName>
<Value>1</Value>
</StateTransitionCmd>
```

Figure B.2 – Device profile document

_____