ZIGBEE CLUSTER LIBRARY SPECIFICATION

ZigBee Document 075123r01ZB

October 19, 2007 7:34 pm

Sponsored by: ZigBee Alliance

Accepted by	This document has been accepted for release by the ZigBee Alliance Board of Directors.
Abstract	This document defines the ZigBee Cluster Library.
Keywords	ZigBee, Profile, Application Framework, ZCL

October 19, 2007

PARTICIPANTS

The following is a list of those who were members of the ZigBee Alliance Application Framework Working Group leadership when this document was released:

Phil Jamieson:	Chair
Don Sturek:	Editor-in-chief
Drew Gislason:	Secretary
Matt Maupin:	Project manager
David Clark:	ZCL technical editor

Contributions were made to this document from the following members:

Table 1:

David Clark	Ted Humpal	Jens Klostergaard Lyngsø	Zachary Smith
Ettore Colicchio	Phil Jamieson	Marco Naeve	Don Sturek
Kent Crouse	William Keith	Philip Orlik	Urban Wicklander
Tim Gillman	Tom Klein	Isaac Pinhas	Walter Young
Drew Gislason	Jared Lemke	Phil Rudland	

NOTICE OF USE AND DISCLOSURE

The ZigBee Specification is available to individuals, companies and institutions free of charge for all non-commercial purposes (including university research, technical evaluation, and development of non-commercial software, tools, or documentation). No part of this specification may be used in development of a product for sale without becoming a member of ZigBee Alliance.

Elements of ZigBee Alliance specifications may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a member of ZigBee). ZigBee is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

This document and the information contained herein are provided on an “AS IS” basis and ZigBee DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. IN NO EVENT WILL ZIGBEE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. All Company, brand and product names may be trademarks that are the sole property of their respective owners.

The above notice and this paragraph must be included on all copies of this document that are made.

ZigBee Alliance, Inc.
2400 Camino Ramon, Suite 375
San Ramon, CA 94583

This page is intentionally blank

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Scope and Purpose

This document specifies the ZigBee Cluster Library (ZCL). The ZCL is a repository for cluster functionality that is developed by the ZigBee Alliance, and, as a consequence, it will be a working library with regular updates as new functionality is added.

A developer constructing a new application profile should use the ZCL to find relevant cluster functionality that can be incorporated into the new profile. Correspondingly, new clusters that are defined for application profiles should be considered for inclusion in the ZCL.

The ZCL consists of the ZCL Foundation, a set of elements that apply across the entire library (such as frame structures, attribute access commands and data types), and a number of sets of clusters. Clusters that are generally useful across many application domains are included in the General set. Clusters that are intended for use mainly in specific application domains are grouped together in domain oriented sets.

1.2 Acronyms and Abbreviations

Table 1:

ACE	Ancillary Control Equipment
CIE	Control and Indicating Equipment
HVAC	Heating, Ventilation, Air Conditioning
IAS	Intruder Alarm System

Table 1:

PIR	Pyroelectric Infra-Red (a type of motion detection sensor)
RSSI	Received Signal Strength Indication
WD	Warning Device
ZCL	ZigBee Cluster Library

1.3 Definitions

1.3.1 ZigBee Definitions

Cluster:- A related collection of attributes and commands, which together define a communications interface between two devices. The devices implement server and client sides of the interface respectively.

Client:- A cluster interface which is listed in the output cluster list of the simple descriptor on an endpoint. Typically this interface sends commands that manipulate the attributes on the corresponding server cluster.

Server:- A cluster interface which is listed in the input cluster list of the simple descriptor on an endpoint. Typically this interface supports all or most of the attributes of the cluster.

1.3.2 Application Domain Definitions

4-pipes:- In a 4-pipe HVAC fan coil system, heated and chilled water each have their own supply and return pipes, while in a 2 pipe system they share the same supply and return. With a 4-pipes system, heating and cooling can take place at the same time in different locations of a building. With a 2-pipes system, only heating or cooling can take place in the whole building.

Ballast factor:- A measure of the light output (lumens) of a ballast and lamp combination in comparison to an ANSI standard ballast operated with the same lamp. Multiply the ballast factor by the rated lumens of the lamp to get the light output of the lamp/ballast combination.

HSV:- Hue, Saturation, Value. A color space, also known as HSB (Hue, Saturation, Brightness). This is a well-known transformation of the RGB (Red, Green, Blue) color space. For more information see e.g. http://en.wikipedia.org/wiki/HSV_color_space.

Illuminance:- The density of incident luminous flux on a surface. Illuminance is the standard metric for lighting levels, and is measured in lux (lx).

Precooling:- Cooling a building in the early (cooler) part of the day, so that the thermal mass of the building decreases cooling needs in the later (hotter) part of the day.

1.4 Conformance Levels

- **expected:** A key word used to describe the behavior of the hardware or software in the design models *assumed* by this Draft. Other hardware and software design models may also be implemented.
- **may:** A key word that indicates flexibility of choice with *no implied preference*.
- **shall:** A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.
- **should:** A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.

1.5 References

The following standards and specifications contain provisions, which through reference in this document constitute provisions of this specification. All the standards and specifications listed are normative references. At the time of publication, the editions indicated were valid. All standards and specifications are subject to revision, and parties to agreements based on this specification are encouraged to investigate the possibility of applying the most recent editions of the standards and specifications indicated below.

1.5.1 Zigbee Alliance Documents

[B1] ZigBee document 053474, ZigBee Specification, ZigBee Alliance.

1.5.2 European Standards Documents

[B2] EN 50131 European Standards Series for Intruder Alarm Systems

1.5.3 IEEE Documents

[B3] IEEE Standards 802, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE, October 2003.

[B4] IEEE 754-1985, IEEE Standard for Binary Floating-Point Arithmetic, IEEE, 1985.

1.5.4 ASHRAE Documents

[B5] ASHRAE 135-2004 standard, Data Communication Protocol for Building Automation and Control Networks

CHAPTER 2

FOUNDATION SPECIFICATION

2.1 Scope and Purpose

This chapter provides an entry point into the documentation for the ZigBee cluster library (ZCL), and specifies the elements that are general across the entire library.

The ZCL frame structure is specified along with ZCL wide commands used to manipulate attributes from all the clusters defined throughout the ZCL. In addition, a set of data types is defined that can be used to represent attributes and a common set of status values returned by commands throughout the ZCL.

An overview is included which lists all the domains specified in the ZCL and the clusters contained therein.

2.2 Cluster Library Overview

The ZigBee Cluster Library (ZCL) is intended to act as a repository for cluster functionality that is developed by ZigBee and, as a consequence, it will be a working library with regular updates as new functionality is added. A developer constructing a new application profile should use the ZCL to find relevant cluster functionality that can be incorporated into the new profile so as not to “re-invent the wheel”. This also allows ZigBee profiles to be developed with more of an object oriented style approach.

2.2.1 Client/Server Model

Throughout the ZCL, a client/server model is employed. This model is illustrated in Figure 2.1

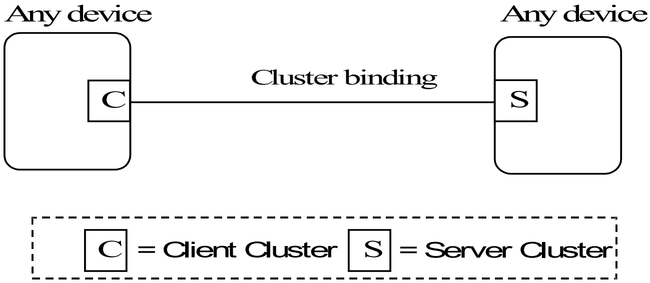


Figure 2.1 The ZCL client server model

A cluster is a related collection of commands and attributes, which together define an interface to specific functionality. Typically, the entity that stores the attributes of a cluster is referred to as the server of that cluster and an entity that affects or manipulates those attributes is referred to as the client of that cluster. However, if required, attributes may also be present on the client of a cluster.

Commands that allow devices to manipulate attributes, e.g. in this document the read attribute (see 2.4.1) or write attribute (see 2.4.3) commands, are (typically) sent from a client device and received by the server device. Any response to those commands, e.g. in this document the read attribute response (see 2.4.2) or the write attribute response (see 2.4.5) commands, are sent from the server device and received by the client device.

Conversely, the command that facilitates dynamic attribute reporting, i.e. the report attribute command (see 2.4.11) is (typically) sent from the server device (as typically this is where the attribute data itself is stored) and sent to the client device that has been bound to the server device.

The clusters supported by an application object within an application profile are identified through the simple descriptor (see [B1]), specified on each active endpoint of a device. In the simple descriptor, the application input cluster list shall contain the list of server clusters supported on the device and the application output cluster list shall contain the list of client clusters supported on the device.

2.2.2 Functional Domains

The ZCL is divided into a number of functional domains, each domain addressing clusters relating to specific functionality. The functional domains defined in the ZCL are listed in Table 2.1.

Table 2.1 Functional Domains Defined in the ZCL

Functional Domain	Cluster ID Range
General	0x0000 – 0x00ff
Closures	0x0100 – 0x01ff
HVAC	0x0200 – 0x02ff
Lighting	0x0300 – 0x03ff
Measurement and sensing	0x0400 – 0x04ff
Security and safety	0x0500 – 0x05ff
Protocol interfaces	0x0600 – 0x06ff

The structure of each of these functional domains is described in the following sub-clauses.

2.2.2.1 General

The general functional domain contains clusters and information that provides generally applicable functions and attributes that are not specific to other functional domains.

This functional domain specifies the clusters listed in Table 2.2.

Table 2.2 Clusters Specified by the General Functional Domain

Cluster ID	Cluster Name	Description
0x0000	Basic	Attributes for determining basic information about a device, setting user device information such as location, and enabling a device.
0x0001	Power configuration	Attributes for determining more detailed information about a device’s power source(s), and for configuring under/over voltage alarms.
0x0002	Device Temperature Configuration	Attributes for determining information about a device’s internal temperature, and for configuring under/over temperature alarms.
0x0003	Identify	Attributes and commands for putting a device into Identification mode (e.g. flashing a light)

Table 2.2 Clusters Specified by the General Functional Domain

Cluster ID	Cluster Name	Description
0x0004	Groups	Attributes and commands for group configuration and manipulation.
0x0005	Scenes	Attributes and commands for scene configuration and manipulation.
0x0006	On/off	Attributes and commands for switching devices between ‘On’ and ‘Off’ states.
0x0007	On/off Switch Configuration	Attributes and commands for configuring On/Off switching devices
0x0008	Level Control	Attributes and commands for controlling devices that can be set to a level between fully ‘On’ and fully ‘Off’.
0x0009	Alarms	Attributes and commands for sending notifications and configuring alarm functionality.
0x000a	Time	Attributes and commands that provide a basic interface to a real-time clock.
0x000b	RSSI Location	Attributes and commands that provide a means for exchanging location information and channel parameters among devices.
0x000c – 0x00ff	-	Reserved.

2.2.2.2 Closures

The closures functional domain contains clusters and information to build devices in the closure domain, e.g. shade controllers.

This functional domain specifies the clusters listed in Table 2.3.

Table 2.3 Clusters Specified by the Closures Functional Domain

Cluster ID	Cluster Name	Description
0x0100	Shade Configuration	Attributes and commands for configuring a shade.
0x0101 – 0x01ff	-	Reserved.

2.2.2.3 HVAC

The HVAC functional domain contains clusters and information to build devices in the HVAC domain, e.g. pumps.

This functional domain specifies the clusters listed in Table 2.4

Table 2.4 Clusters Specified by the HVAC Functional Domain

Cluster ID	Cluster Name	Description
0x0200	Pump Configuration and Control	An interface for configuring and controlling pumps.
0x0201	Thermostat	An interface for configuring and controlling the functionality of a thermostat.
0x0202	Fan Control	An interface for controlling a fan in a heating / cooling system.
0x0203	Dehumidification Control	An interface for controlling dehumidification.
0x0204	Thermostat User Interface Configuration	An interface for configuring the user interface of a thermostat (which may be remote from the thermostat).
0x0205 – 0x02ff	-	Reserved

2.2.2.4 Lighting

The lighting functional domain contains clusters and information to build devices in the lighting domain, e.g. ballast units.

This functional domain specifies the clusters listed in Table 2.5

Table 2.5 Clusters Specified by the Lighting Functional Domain

Cluster ID	Cluster Name	Description
0x0300	Color control	Attributes and commands for controlling the hue and saturation of a color-capable light
0x0301	Ballast Configuration	Attributes and commands for configuring a lighting ballast
0x0302 – 0x03ff	-	Reserved.

2.2.2.5 Measurement and Sensing

The measurement and sensing functional domain contains clusters and information to build devices in the measurement and sensing domain, e.g. a temperature sensor or an occupancy sensor.

This functional domain specifies the clusters listed in Table 2.6.

Table 2.6 Clusters Specified by the Measurement and Sensing Functional Domain

Cluster ID	Cluster Name	Description
0x0400	Illuminance measurement	Attributes and commands for configuring the measurement of illuminance, and reporting illuminance measurements.
0x0401	Illuminance level sensing	Attributes and commands for configuring the sensing of illuminance levels, and reporting whether illuminance is above, below, or on target.
0x0402	Temperature measurement	Attributes and commands for configuring the measurement of temperature, and reporting temperature measurements.
0x0403	Pressure measurement	Attributes and commands for configuring the measurement of pressure, and reporting pressure measurements.
0x0404	Flow measurement	Attributes and commands for configuring the measurement of flow, and reporting flow rates.
0x0405	Relative humidity measurement	Attributes and commands for configuring the measurement of relative humidity, and reporting relative humidity measurements.
0x0406	Occupancy sensing	Attributes and commands for configuring occupancy sensing, and reporting occupancy status.
0x0407 – 0x04ff	-	Reserved.

2.2.2.6 Security and Safety

The security and safety functional domain contains clusters and information to build devices in the security and safety domain, e.g. alarm units.

This functional domain specifies the clusters listed in Table 2.7.

Table 2.7 Clusters Specified by the Security and Safety Functional Domain

Cluster ID	Cluster Name	Description
0x0500	IAS Zone	Attributes and commands for IAS security zone devices.
0x0501	IAS ACE	Attributes and commands for IAS Ancillary Control Equipment.
0x0502	IAS WD	Attributes and commands for IAS Warning Devices.
0x0503 – 0x05ff	-	Reserved.

2.3 Command Frame Formats

All commands, defined in this specification, shall be transmitted to the ZigBee stack using the message service.

2.3.1 General ZCL Frame Format

The ZCL frame format is composed of a ZCL header and a ZCL payload. The general ZCL frame shall be formatted as illustrated in Figure 2.2.

Bits: 8	**0/16**	**8**	**8**	**Variable**
Frame control	Manufacturer code	Transaction sequence number	Command identifier	Frame payload
ZCL header				ZCL payload

Figure 2.2 Format of the General ZCL Frame

2.3.1.1 Frame Control Field

The frame control field is 8-bits in length and contains information defining the command type and other control flags. The frame control field shall be formatted as illustrated in Figure 2.3.

Bits: 0-1	**2**	**3**	**4**	**5-7**
Frame type	Manufacturer specific	Direction	Disable default response	Reserved

Figure 2.3 Format of the Frame Control Field

2.3.1.1.1 Frame Type Sub-field

The frame type sub-field is 2 bits in length and shall be set to one of the non-reserved values listed in Figure 2.4.

Frame Type Value b_1b_0	Description
00	Command acts across the entire profile
01	Command is specific to a cluster
10-11	Reserved

Figure 2.4 Values of the Frame Type Sub-field

2.3.1.1.2 Manufacturer Specific Sub-field

The manufacturer specific sub-field is 1 bit in length and specifies whether this command refers to a manufacturer specific extension to a profile. If this value is set to 1, the manufacturer code field shall be present in the ZCL frame. If this value is set to 0, the manufacturer code field shall not be included in the ZCL frame.

2.3.1.1.3 Direction Sub-field

The direction sub-field specifies the client/server direction for this command. If this value is set to 1, the command is being sent from the server side of a cluster to the client side of a cluster. If this value is set to 0, the command is being sent from the client side of a cluster to the server side of a cluster.

2.3.1.1.4 Disable Default Response Sub-field

The disable default response sub-field is 1 bit in length. If it is set to 0, the Default response command will be returned as a response, under the conditions specified in 2.4.12.2. If it is set to 1, the Default response command will not be returned.

2.3.1.2 Manufacturer Code Field

The manufacturer code field is 16-bits in length and specifies the ZigBee assigned manufacturer code for proprietary extensions to a profile. This field shall only be included in the ZCL frame if the manufacturer specific sub-field of the frame control field is set to 1.

2.3.1.3 Transaction Sequence Number

The transaction sequence number field is 8-bits in length and specifies an identification number for the transaction so that a response-style command frame can be related to a request-style command frame. The application object itself shall maintain an 8-bit counter that is copied into this field and incremented by one for each command sent. When a value of 0xff is reached, the next command shall re-start the counter with a value of 0x00.

The transaction sequence number field can be used by a controlling device, which may have issued multiple commands, so that it can match the incoming responses to the relevant command.

2.3.1.4 Command Identifier Field

The command identifier field is 8-bits in length and specifies the cluster command being used. If the frame type sub-field of the frame control field is set to 0b00, the command identifier corresponds to one of the non-reserved values of Table 2.9. If the frame type sub-field of the frame control field is set to 0b01, the command identifier corresponds to a cluster specific command. The cluster specific command identifiers can be found in each individual document describing the clusters (see also 2.2.1).

2.3.1.5 Frame Payload Field

The frame payload field has a variable length and contains information specific to individual command types. The maximum payload length for a given command is limited by the stack profile in use, in conjunction with the applicable cluster specification and application profile. Fragmentation will be used where available.

2.4 General Command Frames

General command frames are used for manipulating attributes and other general tasks that are not specific to an individual cluster.

The command frames defined in this document are listed in Table 2.8. Each command frame shall be constructed with the frame type sub-field of the frame control field set to 0b00.

All clusters (server and client) shall support generation, reception and execution of the Default response command.

Each cluster (server or client) that implements attributes shall support reception of, execution of, and response to all commands to discover, read, write, report, configure reporting of, and read reporting configuration of these attributes. Generation of these commands is application dependent.

Table 2.8 ZCL Command Frames

Command Identifier Field Value	Description	Reference
0x00	Read attributes	7.1
0x01	Read attributes response	7.2
0x02	Write attributes	7.3
0x03	Write attributes undivided	7.4
0x04	Write attributes response	7.5
0x05	Write attributes no response	7.6
0x06	Configure reporting	7.7
0x07	Configure reporting response	7.8
0x08	Read reporting configuration	7.9
0x09	Read reporting configuration response	7.10
0x0a	Report attributes	7.11
0x0b	Default response	7.12
0x0c	Discover attributes	7.13
0x0d	Discover attributes response	7.14
0x0e – 0xff	Reserved	-

2.4.1 Read Attributes Command

2.4.1.1 Read Attributes Command Frame Format

The read attributes command frame shall be formatted as illustrated in Figure 2.5.

Octets: Variable	2	2	…	2
ZCL header	Attribute identifier 1	Attribute identifier 2	…	Attribute identifier *n*

Figure 2.5 Format of the Read Attributes Command Frame

2.4.1.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to read attributes defined for any cluster in the ZCL or 1 if this command is being used to read manufacturer specific attributes.

The command identifier field shall be set to indicate the read attributes command (see Table 2.9).

2.4.1.1.2 Attribute Identifier Field

The attribute identifier field is 16-bits in length and shall contain the identifier of the attribute that is to be read.

2.4.1.2 When Generated

The read attributes command is generated when a device wishes to determine the values of one or more attributes located on another device. Each attribute identifier field shall contain the identifier of the attribute to be read.

2.4.1.3 Effect on Receipt

On receipt of this command, the device shall process each specified attribute identifier and generate a read attributes response command. The read attributes response command shall contain as many read attribute status records as attribute identifiers included in this command frame. Each read attribute status record shall contain the corresponding attribute identifier from this command frame, a status value evaluated as described below, and, depending on the status value, the value of the attribute itself.

For each attribute identifier included in the command frame, the device shall first check that it corresponds to an attribute that exists on this device. If it does not, the device shall set the status field of the corresponding read attribute status record to UNSUPPORTED_ATTRIBUTE and shall not include an attribute value field. The device shall then move on to the next attribute identifier.

If the attribute identified by the attribute identifier is supported, the device shall set the status field of the corresponding read attribute status record to SUCCESS and shall set the attribute value field to its current value. The device shall then move on to the next attribute identifier.

2.4.2 Read Attributes Response Command

2.4.2.1 Read Attributes Response Command Frame Format

The read attributes response command frame shall be formatted as illustrated in Figure 2.6.

Octets: Variable	**Variable**	**Variable**	**…**	**Variable**
ZCL header	Read attribute status record 1	Read attribute status record 2	…	Read attribute status record *n*

Figure 2.6 Format of the Read Attributes Response Command Frame

Each read attribute status record shall be formatted as illustrated in Figure 2.7

Octets: 2	**1**	**0 / 1**	**0 / Variable**
Attribute identifier	Status	Attribute data type	Attribute data

Figure 2.7 Format of the Read Attributes Status Record Field

2.4.2.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used as a response to reading attributes defined for any cluster in the ZCL or 1 if this command is being used as a response to reading manufacturer specific attributes.

The command identifier field shall be set to indicate the read attributes response command (see Table 2.9).

2.4.2.1.2 Attribute Identifier Field

The attribute identifier field is 16-bits in length and shall contain the identifier of the attribute that has been read. This field shall contain the same value that was included in the corresponding attribute identifier field of the original read attributes command.

2.4.2.1.3 Status Field

The status field is 8-bits in length and specifies the status of the read operation on this attribute. This field shall be set to SUCCESS, if the operation was successful, or an error code, as specified in 2.4.1.3, if the operation was not successful.

2.4.2.1.4 Attribute Data Type Field

The attribute data type field shall contain the data type of the attribute in the same read attributes status record (see Table 2.14). This field shall only be included if the associated status field contains a value of SUCCESS.

2.4.2.1.5 Attribute Data Field

The attribute data field is variable in length and shall contain the current value of this attribute. This field shall only be included if the associated status field contains a value of SUCCESS.

2.4.2.2 When Generated

The read attributes response command is generated in response to a read attributes command. The command frame shall contain a read attribute status record for each attribute identifier specified in the original read attributes command. For each read attribute status record, the attribute identifier field shall contain the identifier specified in the original read attributes command. The status field shall contain a suitable status code, as detailed in Clause 2.4.1.3. The attribute data field shall only be included in the read attribute status record if the associated status field contains a value of SUCCESS and shall contain the current value of the attribute that was read.

Only as many read attribute status records that will fit in the frame shall be returned.

2.4.2.3 Effect on Receipt

On receipt of this command, the originator is notified of the results of its original read attributes attempt and, for each successful request, the value of the requested attribute.

If some trailing attribute status records have not been returned, due to space limitations in the frame, the originator may issue a further read attributes command to obtain their values.

2.4.3 Write Attributes Command

2.4.3.1 Write Attributes Command Frame Format

The write attributes command frame shall be formatted as illustrated in Figure 2.8.

Octets: Variable	Variable	…	Variable	Variable
ZCL header	Write attribute record 1	Write attribute record 2	…	Write attribute record *n*

Figure 2.8 Format of the Write Attributes Command Frame

Each write attribute record shall be formatted as illustrated in Figure 2.9.

Octets: 2	1	Variable
Attribute identifier	Attribute data type	Attribute data

Figure 2.9 Format of the Write Attribute Record Field

2.4.3.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to write attributes defined for any cluster in the ZCL or 1 if this command is being used to write manufacturer specific attributes.

The command identifier field shall be set to indicate the write attributes command (see Table 2.9).

2.4.3.1.2 Attribute Identifier Field

The attribute identifier field is 16-bits in length and shall contain the identifier of the attribute that is to be written.

2.4.3.1.3 Attribute Data Type Field

The attribute data type field shall contain the data type of the attribute that is to be written.

2.4.3.1.4 Attribute Data Field

The attribute data field is variable in length and shall contain the actual value of the attribute that is to be written.

2.4.3.2 When Generated

The write attributes command is generated when a device wishes to change the values of one or more attributes located on another device. Each write attribute record shall contain the identifier and the actual value of the attribute to be written.

2.4.3.3 Effect on Receipt

On receipt of this command, the device shall attempt to process each specified write attribute record and shall construct a write attribute response command (2.4.5). Each write attribute status record of the constructed command shall contain the identifier from the corresponding write attribute record and a status value evaluated as described below.

For each write attribute record included in the command frame, the device shall first check that it corresponds to an attribute that is implemented on this device. If it does not, the device shall set the status field of the corresponding write attribute status record to UNSUPPORTED_ATTRIBUTE and move on to the next write attribute record.

If the attribute identified by the attribute identifier is supported, the device shall check whether the attribute data type field is correct. If not, the device shall set the status field of the corresponding write attribute status record to INVALID_DATA_TYPE and move on to the next write attribute record.

If the attribute data type is correct, the device shall check whether the attribute is writable. If the attribute is designated as read only, the device shall set the status field of the corresponding write attribute status record to READ_ONLY and move on to the next write attribute record.

If the attribute is writable, the device shall check that the supplied value in the attribute data field is within the specified range of the attribute. If the supplied value does not fall within the specified range of the attribute, the device shall set the status field of the corresponding write attribute status record to INVALID_VALUE and move on to the next write attribute record.

If the value supplied in the attribute data field is within the specified range of the attribute, the device shall write the supplied value to the identified attribute and shall move on to the next write attribute record. In this (successful) case, a write attribute status record shall not be generated.

When all write attribute records have been processed, the device shall generate the constructed write attributes response command. If there are no write attribute status records in the constructed command, indicating that all attributes were written successfully, a single write attribute status record shall be included in the command, with the status field set to SUCCESS and the attribute identifier field omitted.

2.4.4 Write Attributes Undivided Command

The write attributes undivided command is generated when a device wishes to change the values of one or more attributes located on another device, in such a way that if any attribute cannot be written (e.g. if an attribute is not implemented on the device, or a value to be written is outside its valid range), no attribute values are changed.

In all other respects, including generation of a write attributes response command, the format and operation of the command is the same as that of the write attributes command, except that the command identifier field shall be set to indicate the write attributes undivided command (see Table 2.9).

2.4.5 Write Attributes Response Command

2.4.5.1 Write Attributes Response Command Frame Format

The write attributes response command frame shall be formatted as illustrated in Figure 2.10.

Octets: Variable	**3**	**3**	**…**	**3**
ZCL header	Write attribute status record 1	Write attribute status record 2	…	Write attribute status record *n*

Figure 2.10 Format of the Write Attributes Response Command Frame

Each write attribute status record shall be formatted as illustrated in Figure 2.11.

Octets: 1	**2**
Status	Attribute identifier

Figure 2.11 Format of the Write Attribute Status Record Field

2.4.5.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used as a response to writing attributes defined for any cluster in the ZCL or 1 if this command is being used as a response to writing manufacturer specific attributes.

The command identifier field shall be set to indicate the write attributes response command (see Table 2.9).

2.4.5.1.2 Status Field

The status field is 8-bits in length and specifies the status of the write operation attempted on this attribute, as detailed in 2.4.3.3.

Note that write attribute status records are not included for successfully written attributes, in order to save bandwidth. In the case of successful writing of all attributes, only a single write attribute status record shall be included in the command, with the status field set to SUCCESS and the attribute identifier field omitted.

2.4.5.1.3 Attribute Identifier Field

The attribute identifier field is 16-bits in length and shall contain the identifier of the attribute on which the write operation was attempted.

2.4.5.2 When Generated

The write attributes response command is generated in response to a write attributes command.

2.4.5.3 Effect on Receipt

On receipt of this command, the device is notified of the results of its original write attributes command.

2.4.6 Write Attributes No Response Command

2.4.6.1 Write Attributes No Response Command Frame Format

The write attributes no response command frame shall be formatted as illustrated in Figure 2.12.

Octets: Variable	Variable	Variable	…	Variable
ZCL header	Write attribute record 1	Write attribute record 2	…	Write attribute record *n*

Figure 2.12 Format of the Write Attributes No Response Command Frame

Each write attribute record shall be formatted as illustrated in Figure 2.9.

2.4.6.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to write attributes defined for any cluster in the ZCL or 1 if this command is being used to write manufacturer specific attributes.

The command identifier field shall be set to indicate the write attributes no response command (see Table 2.9).

2.4.6.1.2 Write Attribute Records

Each write attribute record shall be formatted as illustrated in Figure 2.9 Its fields have the same meaning and contents as the corresponding fields of the Write attributes command.

2.4.6.2 When Generated

The write attributes no response command is generated when a device wishes to change the value of one or more attributes located on another device but does not require a response. Each write attribute record shall contain the identifier and the actual value of the attribute to be written.

2.4.6.3 Effect on Receipt

On receipt of this command, the device shall attempt to process each specified write attribute record.

For each write attribute record included in the command frame, the device shall first check that it corresponds to an attribute that is implemented on this device. If it does not, the device shall ignore the attribute and move on to the next write attribute record.

If the attribute identified by the attribute identifier is supported, the device shall check whether the attribute is writable. If the attribute is designated as read only, the device shall ignore the attribute and move on to the next write attribute record.

If the attribute is writable, the device shall check that the supplied value in the attribute data field is within the specified range of the attribute. If the supplied value does not fall within the specified range of the attribute, the device shall ignore the attribute and move on to the next write attribute record.

If the value supplied in the attribute data field is within the specified range of the attribute, the device shall write the supplied value to the identified attribute and move on to the next write attribute record.

2.4.7 Configure Reporting Command

The Configure Reporting command is used to configure the reporting mechanism for one or more of the attributes of a cluster.

The individual cluster definitions specify which attributes shall be available to this reporting mechanism, however specific implementations of a cluster may make additional attributes available.

2.4.7.1 Configure Reporting Command Frame Format

The Configure Reporting command frame shall be formatted as illustrated in Figure 2.13.

Octets: Variable	Variable	Variable	…	Variable
ZCL header	Attribute reporting configuration record 1	Attribute reporting configuration record 2	…	Attribute reporting configuration record n

Figure 2.13 Format of the Configure Reporting Command Frame

There shall be one attribute reporting configuration record for each attribute to be configured. Each such record shall be formatted as illustrated in Figure 2.14.

Octets: 1	2	0/1	0/2	0/2	0/Variable	0/2
Direction	Attribute identifier	Attribute data type	Minimum reporting interval	Maximum reporting interval	Reportable change	Timeout period

Figure 2.14 Format of the Attribute Reporting Configuration Record

2.4.7.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to configure attribute reports defined for any cluster in the ZCL or 1 if this command is being used to configure attribute reports for manufacturer specific attributes.

The command identifier field shall be set to indicate the report configuration command (see Table 2.9).

2.4.7.1.2 Direction Field

The direction field specifies whether values of the attribute are be reported, or whether reports of the attribute are to be received.

If this value is set to 0x00, then the attribute data type field, the minimum reporting interval field and the maximum reporting interval field are included in the payload, and the timeout period field is omitted. The record is sent to a cluster server (or client) to configure how it sends reports to a client (or server) of the same cluster.

If this value is set to 0x01, then the timeout period field is included in the payload, and the attribute data type field, the minimum reporting interval field and the maximum reporting interval field are omitted. The record is sent to a cluster client (or server) to configure how it should expect reports from a server (or client) of the same cluster.

All other values of this field are reserved.

2.4.7.1.3 Attribute Identifier Field

If the direction field is 0x00, this field contains the identifier of the attribute that is to be reported. If instead the direction field is 0x01, the device shall expect reports of values of this attribute.

2.4.7.1.4 Attribute Data Type Field

The Attribute data type field contains the data type of the attribute that is to be reported.

2.4.7.1.5 Minimum Reporting Interval Field

The minimum reporting interval field is 16-bits in length and shall contain the minimum interval, in seconds, between issuing reports of the specified attribute.

If this value is set to 0x0000, then there is no minimum limit, unless one is imposed by the specification of the cluster using this reporting mechanism or by the applicable profile.

2.4.7.1.6 Maximum Reporting Interval Field

The maximum reporting interval field is 16-bits in length and shall contain the maximum interval, in seconds, between issuing reports of the specified attribute.

If this value is set to 0xffff, then the device shall not issue reports for the specified attribute, and the configuration information for that attribute need not be maintained. (Note:- in an implementation using dynamic memory allocation, the memory space for that information may then be reclaimed).

2.4.7.1.7 Reportable Change Field

The reportable change field shall contain the minimum change to the attribute that will result in a report being issued. This field is of variable length. For attributes with 'analog' data type (see Table 2.14) the field has the same data type as the attribute. The sign (if any) of the reportable change field is ignored.

For attributes of 'discrete' data type (see Table 2.14) this field is omitted.

2.4.7.1.8 Timeout Period Field

The timeout period field is 16-bits in length and shall contain the maximum expected time, in seconds, between received reports for the attribute specified in the attribute identifier field. If more time than this elapses between reports, this may be an indication that there is a problem with reporting.

If this value is set to 0x0000, reports of the attribute are not subject to timeout.

Note that, for a server/client connection to work properly using automatic reporting, the timeout value set for attribute reports to be received by the client (or server) cluster must be set somewhat higher than the maximum reporting interval set for the attribute on the server (or client) cluster.

2.4.7.2 When Generated

The report configuration command is generated when a device wishes to configure a device to automatically report the values of one or more of its attributes, or to receive such reports.

2.4.7.3 Effect on Receipt

On receipt of this command, the device shall attempt to process each attribute reporting configuration record and shall construct a configure reporting response command. Each attribute status record of the constructed command shall contain an identifier from an attribute reporting configuration record and a status value evaluated as described below.

If the direction field is 0x00, indicating that the reporting intervals and reportable change are being configured, then

- If the attribute specified in the attribute identifier field is not implemented on this device, the device shall construct an attribute status record with the status field set to UNSUPPORTED_ATTRIBUTE.
- Else, if the attribute identifier in this field cannot be reported (because it is not in the list of mandatory reportable attributes in the relevant cluster specification, and support has also not been implemented as a manufacturer option), the device shall construct an attribute status record with the status field set to UNREPORTABLE_ATTRIBUTE.
- Else, if the attribute data type field is incorrect, the device shall construct an attribute status record with the status field set to INVALID_DATA_TYPE.

Else, if the minimum reporting interval field is less than any minimum set by the relevant cluster specification or application profile, or the value of the maximum reporting interval field is non-zero and is less than that of the minimum reporting interval field, the device shall construct an attribute status record with the status field set to INVALID_VALUE.

Else the device shall set the minimum and maximum reporting intervals and the reportable change for the attribute to the values contained in the corresponding fields.

Else the direction field is 0x01, indicating that the timeout period is being configured, then

If reports of values of the attribute identifier specified in the attribute identifier field cannot be received (because it is not in the list of mandatory reportable attributes in the relevant cluster specification, and support has also not been implemented as a manufacturer option), or the timeout feature is not spported, the device shall construct an attribute status record with the status field set to UNSUPPORTED_ATTRIBUTE.

Else the device shall set the timeout value for the attribute identifier specified in the attribute identifier field to the value of the timeout period field. Note that the action to be taken by the device if the timeout period is exceeded is cluster and device dependent, including optionally taking no actionl.

When all attribute reporting configuration records have been processed, the device shall generate the constructed configure reporting response command. If there are no attribute status records in the constructed command, indicating that all attributes were configured successfully, a single attribute status record shall be included in the command, with the status field set to SUCCESS and the attribute identifier field omitted.

The device shall then proceed to generate attribute reports according the configuration just set up, by means of the Report attributes command (see 2.4.11.2.1 through 2.4.11.2.4).

2.4.8 Configure Reporting Response Command

The Configure Reporting Response command is used to respond to a Configure Reporting command.

2.4.8.1 Configure Reporting Response Command Frame Format

The Configure Reporting Response command frame shall be formatted as illustrated in Figure 2.15

Octets: Variable	**4**	**4**	**…**	**4**
ZCL header	Attribute status record 1	Attribute status record 2	…	Attribute status record *n*

Figure 2.15 Format of the Configure Reporting Response Command Frame

Each attribute status record shall be formatted as illustrated in Figure 2.16.

Octets: 1	**1**	**2**
Status	Direction	Attribute identifier

Figure 2.16 Format of the Attribute Status Record Field

2.4.8.1.1 ZCL Header Fields

The frame control field is specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used as a response to configuring attribute reports defined for any cluster in the ZCL or 1 if this command is being used as a response to configuring attribute reports for manufacturer specific attributes.

The command identifier field shall be set to indicate the report configuration response command (see Table 2.9).

2.4.8.1.2 Direction Field

The direction field specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01).

All other values of this field are reserved.

2.4.8.1.3 Status Field

The status field specifies the status of the configure reporting operation attempted on this attribute, as detailed in 2.4.7.3.

Note that attribute status records are not included for successfully configured attributes, in order to save bandwidth. In the case of successful configuration of all attributes, only a single attribute status record shall be included in the command, with the status field set to SUCCESS and the attribute identifier field omitted.

2.4.8.2 When Generated

The Configure Reporting Response command is generated in response to a Configure Reporting command.

2.4.8.3 Effect on Receipt

On receipt of this command, the device is notified of the success (or otherwise) of its original configure reporting command, for each attribute.

2.4.9 Read Reporting Configuration Command

The Read Reporting Configuration command is used to read the configuration details of the reporting mechanism for one or more of the attributes of a cluster.

2.4.9.1 Read Reporting Configuration Command Frame Format

The Read Reporting Configuration command frame shall be formatted as illustrated in Figure 2.17

Octets: Variable	**3**	**3**	**…**	**3**
ZCL header	Attribute record 1	Attribute record 2	…	Attribute record *n*

Figure 2.17 Format of the Read Reporting Configuration Command Frame

Each attribute record shall be formatted as illustrated in Figure 2.18.

Octets: 1	**2**
Direction	Attribute identifier

Figure 2.18 Format of the Attribute Status Record Field

2.4.9.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to read the reporting configuration of attributes defined for any cluster in the ZCL or 1 if this command is being used to read the reporting configuration of manufacturer specific attributes.

The command identifier field shall be set to indicate the read reporting configuration command (see Table 2.9).

2.4.9.1.2 Direction Field

The direction field specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01).

All other values of this field are reserved.

2.4.9.1.3 Attribute Identifier Field

The attribute identifier field shall contain the identifier of the attribute whose reporting configuration details are to be read.

2.4.9.2 Effect on Receipt

On receipt of this command, a device shall generate a read reporting configuration response command containing the details of the reporting configuration for each of the attributes specified in the command (see 2.4.10).

2.4.10 Read Reporting Configuration Response Command

The Read Reporting Configuration Response command is used to respond to a Read Reporting Configuration command.

2.4.10.1 Read Reporting Configuration Response Command Frame Format

The read reporting configuration response command frame shall be formatted as illustrated in Figure 2.19

Octets: Variable	Variable	Variable	…	Variable
ZCL header	Attribute reporting configuration record 1	Attribute reporting configuration record 2	…	Attribute reporting configuration record n

Figure 2.19 Format of the Read Reporting Configuration Response Command Frame

There shall be one attribute reporting configuration record for each attribute record of the received read reporting configuration command. Each such record shall be formatted as illustrated in Figure 2.20.

Octets: 1	1	2	0/1	0/2	0/2	0/Variable	0/2
Status	Direction	Attribute identifier	Attribute data type	Minimum reporting interval	Maximum reporting interval	Reportable change	Timeout period

Figure 2.20 Format of the Attribute Reporting Configuration Record Field

2.4.10.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to for attributes specified in the ZCL or 1 if this command is being used for manufacturer specific attributes.

The command identifier field shall be set to indicate the Read reporting configuration response command (see Table 2.9).

2.4.10.1.2 Status Field

If the attribute is not implemented on the sender or receiver of the command, whichever is relevant (depending on direction), this field shall be set to UNSUPPORTED_ATTRIBUTE. If the attribute is supported, but is not capable of being reported, this field shall be set to UNREPORTABLE_ATTRIBUTE. Otherwise, this field shall be set to SUCCESS.

If the status field is not set to SUCCESS, all fields except the direction and attribute identifier fields shall be omitted.

2.4.10.1.3 Direction Field

The direction field specifies whether values of the attribute are reported (0x00), or whether reports of the attribute are received (0x01).

If this value is set to 0x00, then the attribute data type field, the minimum reporting interval field, the maximum reporting interval field and the reportable change field are included in the payload, and the timeout period field is omitted. If this value is set to 0x01, then the timeout period field is included in the payload, and the attribute data type field, the minimum reporting interval field, the maximum reporting interval field and the reportable change field are omitted.

All other values of this field are reserved.

2.4.10.1.4 Attribute Identifier Field

The attribute identifier field is 16-bits in length and shall contain the identifier of the attribute that the reporting configuration details apply to.

2.4.10.1.5 Minimum Reporting Interval Field

The minimum reporting interval field is 16-bits in length and shall contain the minimum interval, in seconds, between issuing reports for the attribute specified in the attribute identifier field. If the minimum reporting interval has not been configured, this field shall contain the value 0xffff.

2.4.10.1.6 Maximum Reporting Interval Field

The maximum reporting interval field is 16-bits in length and shall contain the maximum interval, in seconds, between issuing reports for the attribute specified in the attribute identifier field. If the maximum reporting interval has not been configured, this field shall contain the value 0xffff.

2.4.10.1.7 Reportable Change Field

The reportable change field shall contain the minimum change to the attribute that will result in a report being issued. For attributes with 'analog' data type (see Table 2.14) the field has the same data type as the attribute. If the reportable change has not been configured, this field shall contain the invalid value for the relevant data type.

For attributes of 'discrete' data type (see Table 2.14) this field is omitted.

2.4.10.1.8 Timeout Period Field

The timeout period field is 16-bits in length and shall contain the maximum expected time, in seconds, between received reports for the attribute specified in the attribute identifier field. If the timeout period has not been configured, this field shall contain the value 0xffff.

2.4.10.2 When Generated

The read reporting configuration response command is generated in response to a read reporting configuration command. Only as many attribute reporting configuration records as will fit in the frame shall be returned.

2.4.10.3 Effect on Receipt

On receipt of this command, the originator is notified of the results of its original read reporting configuration command.

If some trailing attribute reporting configuration records have not been returned, due to space limitations in the frame, the originator may issue a further read reporting configuration command to obtain their values.

2.4.11 Report Attributes Command

The report attributes command is used by a device to report the values of one or more of its attributes to another device, bound a priori. Individual clusters, defined elsewhere in the ZCL, define which attributes are to be reported and at what interval.

2.4.11.1 Report Attributes Command Frame Format

The report attributes command frame shall be formatted as illustrated in Figure 2.21.

Octets: Variable	**Variable**	**Variable**	**…**	**Variable**
ZCL header	Attribute report 1	Attribute report 2	…	Attribute report *n*

Figure 2.21 Format of the Report Attributes Command Frame

Each attribute report field shall be formatted as illustrated in Figure 2.22.

Octets: 2	**1**	**Variable**
Attribute identifier	Attribute data type	Attribute data

Figure 2.22 Format of the Attribute Report Fields

2.4.11.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being used to report attributes defined for any cluster in the ZCL or 1 if this command is being used to report manufacturer specific attributes.

The command identifier field shall be set to indicate the report attributes command (see Table 2.9).

2.4.11.1.2 Attribute Identifier Field

The attribute identifier field is 16-bits in length and shall contain the identifier of the attribute that is being reported.

2.4.11.1.3 Attribute Data Type Field

The attribute data type field contains the data type of the attribute that is being reported.

2.4.11.1.4 Attribute Data Field

The attribute data field is variable in length and shall contain the actual value of the attribute being reported.

2.4.11.2 When Generated

The report attributes command is generated when a device has been configured to report the values of one or more of its attributes to another device., and when the conditions that have been configured are satisfied. These conditions are detailed in the following sections.

2.4.11.2.1 Periodic Reporting

A report shall be generated when the time that has elapsed since the previous report of the same attribute is equal to the Maximum Reporting Interval for that attribute (see 2.4.7.1.6). The time of the first report after configuration is not specified.

If the Maximum Reporting Interval is set to 0x0000, there is no periodic reporting, but change based reporting is still operational.

If the Maximum Reporting Interval is set to 0xffff, no reports shall be generated, whatever other conditions are satisfied.

2.4.11.2.2 Changes to 'Discrete' Attributes

If the attribute has a 'discrete' data type, a report shall be generated when the attribute undergoes any change of value. Discrete types are general data types (which are often used as set of bit fields), logical types, bitmap types, enumerations, strings, identifiers and IEEE address (see Table 2.14).

Reporting is subject to the Minimum Reporting Interval for that attribute (see 2.4.7.1.5). After a report, no further reports are sent during this interval.

2.4.11.2.3 Changes to 'Analog' Attributes

If the attribute has an 'analog' data type, a report shall be generated when the attribute undergoes a change of value, in a positive or negative direction, equal to or greater than the Reportable Change for that attribute (see 2.4.7.1.7). The change is measured from the value of the attribute when the Reportable Change is configured, and thereafter from the previously reported value of the attribute.

Analog types are signed and unsigned integer types, floating point types and time types (see Table 2.14).

Reporting is subject to the Minimum Reporting Interval for that attribute (see 2.4.7.1.5). After a report, no further reports are sent during this interval.

2.4.11.2.4 Cluster Specific Conditions

The specification for a cluster may add additional conditions for specific attributes of that cluster.

2.4.11.2.5 Consolidation of Attribute Reporting

In order to reduce the resources (such as the number of timers) required for attribute reporting, a device may adapt the timing of reports by relaxing the configured minimum and maximum periods as described below. By employing these techniques a device may limit the number of timers required to any manufacturer specific value, including use of only a single timer, though at the cost of some side effects, such as increased network traffic in some cases.

In consolidating timers, a number of principles apply:-

1/ The maximum reporting interval of an attribute may be reduced, as it should not normally cause a problem to devices to receive reports more frequently than expected – typical reporting intervals are seconds to minutes. It may not be increased, as this may be incompatible with any timeout period set.

2/ The minimum reporting interval of an attribute may also be reduced. However, it may not be increased, as an application may be relying on receiving reports of changes to an attribute within a given delay time. Minimum values are generally used to reduce network traffic, but this is less important than ensuring that the application timing needs are satisfied.

3/ From (1), when consolidating the maximum reporting periods of two or more attributes together, the consolidated reporting period shall be equal to the lowest of the configured maximum intervals of the attributes to be reported.

4/ Similarly, from (2), when consolidating the minimum reporting periods of two or more attributes together, the consolidated reporting period shall be equal to the lowest of the configured minimum intervals of the attributes to be reported.

As a first step, timers for attributes on the same cluster may be consolidated. Such adaptations should aim to send attribute reports for different attributes of the same cluster at the same time, so that they can be consolidated into fewer attribute reports, thus reducing network traffic.

To reduce the number of timers further, timers may be consolidated across clusters and endpoints if needed.

(Note that it is not generally possible to consolidate timeout values (see 2.4.7.1.8) of received attribute reports.)

2.4.11.3 Effect on Receipt

On receipt of this command, a device is notified of the latest values of one or more of the attributes of another device.

2.4.12 Default Response Command

2.4.12.1 Default Response Command Frame Format

The default response command frame shall be formatted as illustrated in Figure 2.23.

Octets: Variable	1	1
ZCL header	Command identifier	Status code

Figure 2.23 Format of the Default Response Command Frame

2.4.12.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 if this command is being sent in response to a command defined for any cluster in the ZCL or 1 if this command is being sent in response to a manufacturer specific command.

The command identifier sub-field shall be set to indicate the default response command (see Table 2.9).

2.4.12.1.2 Command Identifier Field

The command identifier field is 8-bits in length and specifies the identifier of the received command to which this command is a response.

2.4.12.1.3 Status Code Field

The status code field is 8-bits in length and specifies either SUCCESS or the nature of the error that was detected in the received command. It shall be one of the status enumerations listed in Table 2.15.

2.4.12.2 When Generated

The default response command is generated when a device receives a unicast command with the Disable default response bit of its Frame control field set to 0 (see 2.3.1.1.4), and there is no other relevant response specified for the command. If a device receives a command in error through a broadcast or multicast transmission, the command shall be discarded and the default response command not generated.

If the identifier of the received command is not supported on the device, it shall set the command identifier field to the value of the identifier of the command received in error. The error code field shall be set to the either:-
UNSUP_CLUSTER_COMMAND,
UNSUP_GENERAL_COMMAND,
UNSUP_MANUF_CLUSTER_COMMAND or
UNSUP_MANUF_GENERAL_COMMAND, as appropriate.

The default response command shall be generated in response to reception of all commands, including response commands (such as the write attributes response command), under the conditions specified above. However, the default response command shall not be generated in response to reception of another default response command.

2.4.12.3 Effect on Receipt

On receipt of this command, the device is notified of the success or otherwise of the generated command with the same transaction sequence number (see 2.3.1.3).

2.4.13 Discover Attributes Command

2.4.13.1 Discover Attributes Command Frame Format

The discover attributes command frame shall be formatted as illustrated in Figure 2.24.

Octets: Variable	2	1
ZCL header	Start attribute identifier	Maximum attribute identifiers

Figure 2.24 Format of the Discover Attributes Command Frame

2.4.13.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to 0 to discover standard attributes in a ZigBee cluster or 1 to discover manufacturer specific attributes in either a standard or a manufacturer specific cluster.

The command identifier field shall be set to indicate the discover attributes command (see Table 2.9).

2.4.13.1.2 Start Attribute Identifier Field

The start attribute identifier field is 16-bits in length and specifies the value of the identifier at which to begin the attribute discovery.

2.4.13.1.3 Maximum Attribute Identifiers Field

The maximum attribute identifiers field is 8-bits in length and specifies the maximum number of attribute identifiers that are to be returned in the resulting discover attributes response command.

2.4.13.2 When Generated

The discover attributes command is generated when a remote device wishes to discover the identifiers and types of the attributes on a device which are supported within the cluster to which this command is directed.

2.4.13.3 Effect on Receipt

On receipt of this command, the device shall construct an ordered list of attribute information records, each containing a discovered attribute identifier and its data type, in ascending order of attribute identifiers. This list shall start with the first attribute that has an identifier that is equal to or greater than the identifier specified in the start attribute identifier field. The number of attribute identifiers included in the list shall not exceed that specified in the maximum attribute identifiers field.

The device shall then generate a discover attributes response command containing the discovered attributes and their types, and shall return it to the originator of the discover attributes command.

2.4.14 Discover Attributes Response Command

2.4.14.1 Discover Attributes Response Command Frame Format

The discover attributes response command frame shall be formatted as illustrated in Figure 2.25.

Octets: Variable	**1**	**3**	**3**	**…**	**3**
ZCL header	Discovery complete	Attribute information 1	Attribute information 2	…	Attribute information n

Figure 2.25 Format of the Discover Attributes Response Command Frame

Each attribute report field shall be formatted as illustrated in Figure 2.26.

Octets: 2	**1**
Attribute identifier	Attribute data type

Figure 2.26 Format of the Attribute Report Fields

2.4.14.1.1 ZCL Header Fields

The frame control field shall be specified as follows. The frame type sub-field shall be set to indicate a profile wide command (0b00). The manufacturer specific sub-field shall be set to the same value included in the original discover attributes command.

The command identifier field shall be set to indicate the discover attributes response command (see Table 2.9).

2.4.14.1.2 Discovery Complete Field

The discovery complete field is a boolean field. A value of 0 indicates that there are more attributes to be discovered. A value of 1 indicates that there are no more attributes to be discovered.

2.4.14.1.3 Attribute Identifier Field

The attribute identifier field shall contain the identifier of a discovered attribute. Attributes shall be included in ascending order, starting with the lowest attribute identifier that is greater than or equal to the start attribute identifier field of the received discover attributes command.

2.4.14.1.4 Attribute Data Type Field

The attribute data type field shall contain the data type of the attribute in the same attribute report field (see Table 2.14).

2.4.14.2 When Generated

The discover attributes response command is generated in response to a discover attributes command.

2.4.14.3 Effect on Receipt

On receipt of this command, the device is notified of the results of its attribute discovery request.

Following the receipt of this command, if the discovery complete field indicates that there are more attributes to be discovered, the device may choose to send subsequent discover attribute request commands to obtain the rest of the attribute identifiers. In this case, the start attribute identifier specified in the next attribute discovery request command should be set equal to one plus the last attribute identifier received in the discover attributes response command.

2.5 Addressing, Types and Enumerations

2.5.1 Addressing

ZigBee uses a number of concepts to address application profiles, clusters, device descriptions, attributes and commands, each with their own constraints. This sub-clause details these constraints.

2.5.1.1 Profile Identifier

A profile identifier is 16-bits long and specifies the application profile being used. A profile identifier shall be set to one of the non-reserved values listed in

Table 2.9. Within a manufacturer specific application profile, the full ranges of clusters, device, attribute and command identifiers can be used.

Table 2.9 Valid Profile Identifier Values

Profile Identifier	Description
0x0000 – 0x7fff	Standard ZigBee application profile.
0x8000 – 0xbfff	Reserved.
0xc000 – 0xffff	Manufacturer Specific application profile.

2.5.1.2 Device Identifier

A device identifier is 16-bits in length and specifies a specific device within a standard application profile. A device identifier shall be set to one of the non-reserved values listed in Table 2.10.

Table 2.10 Valid Device Identifier Values

Device Identifier	Description
0x0000 – 0xbfff	Standard ZigBee device description.
0xc000 – 0xffff	Reserved.

2.5.1.3 Cluster Identifier

A cluster identifier is 16-bits in length and specifies the set of related commands and attributes within a standard application profile. It shall be set to one of the non-reserved values listed in Table 2.11.

Table 2.11 Valid Cluster Identifier Values

Cluster Identifier	Description
0x0000 – 0x7fff	Standard ZigBee cluster.
0x8000 – 0xfbff	Reserved.
0xfc00 – 0xffff	Manufacturer specific cluster within a standard ZigBee profile.

2.5.1.4 Attribute Identifier

An attribute identifier is 16-bits in length and specifies a single attribute within a standard application profile. An attribute identifier, defined within the ZCL, shall be set to one of the non-reserved values listed in Table 2.12. Manufacturer specific attributes within a standard ZigBee cluster can be defined over the full 16-bit range but any manipulation shall be applied using the appropriate manufacturer code.

Table 2.12 Valid ZCL Defined Attribute Identifier Values

Attribute Identifier	Description
0x0000 – 0x3fff	Standard ZigBee attribute.
0x4000 – 0xffff	Reserved.

2.5.1.5 Command Identifier

A command identifier is 8-bits in length and specifies a specific command within the ZCL as a whole or within a specific cluster. A command identifier shall be set to one of the non-reserved values listed in Table 2.13. Manufacturer specific commands within a standard ZigBee cluster can be defined over the full 8-bit range but each shall use the appropriate manufacturer code.

Table 2.13 Valid ZCL Defined Command Identifier Values

Command Identifier	Description
0x00 – 0x7f	Standard ZigBee command.
0x80 – 0xff	Reserved.

2.5.2 Data Types

ZigBee devices, such as thermostats, lamps, etc., are defined in terms of the attributes they contain, which can be written, read or reported using the commands defined in clause 2.4. Table 2.14 details the data types and formats that can be used for these attributes. Note that individual clusters, which may use different or new types, show valid values, ranges, and units for the attributes they represent.

Each data type is allocated an 8-bit data type ID. The most significant 5 bits of this ID is used to divide the types into 32 type classes, and the least significant 3 bits specify a specific data type within this class.

Table 2.14 also indicates for each data type whether it is considered to be 'analog' or 'discrete'. Values of analog types may be added to or subtracted from other values of the same type, and are typically used to measure the value of properties in the real world that vary continuously over a range. Values of discrete data types only have meaning as individual values, and may not be added or subtracted.

Table 2.14 Data Types

Type Class	Data Type ID	Data Type	Length Of Data (Octets)	Invalid Number	Analog / Discrete
Null	0x00	No data	0	-	-
	0x01 – 0x7	Reserved	-	-	
General data	0x08	8-bit data	1	-	D
	0x09	16-bit data	2	-	
	0x0a	24-bit data	3	-	
	0x0b	32-bit data	4	-	
	0x0c – 0x0f	Reserved	-	-	
Logical	0x10	Boolean	1	0xff	D
	0x11 – 0x17	Reserved	-	-	
Bitmap	0x18	8-bit bitmap	1	-	D
	0x19	16-bit bitmap	2	-	
	0x1a	24-bit bitmap	3	-	
	0x1b	32-bit bitmap	4	-	
	0x1c – 0x1f	Reserved	-	-	
Unsigned integer	0x20	Unsigned 8-bit integer	1	0xff	A
	0x21	Unsigned 16-bit integer	2	0xffff	
	0x22	Unsigned 24-bit integer	3	0xffffff	
	0x23	Unsigned 32-bit integer	4	0xffffffff	
	0x24 – 0x27	Reserved	-	-	

Table 2.14 Data Types (Continued)

Signed integer	0x28	Signed 8-bit integer	1	0x80	A
	0x29	Signed 16-bit integer	2	0x8000	
	0x2a	Signed 24-bit integer	3	0x800000	
	0x2b	Signed 32-bit integer	4	0x80000000	
	0x2c – 0x2f	Reserved	-	-	
Enumeration	0x30	8-bit enumeration	1	0xff	D
	0x31	16-bit enumeration	2	0xffff	
	0x32 – 0x37	Reserved	-	-	
Floating point	0x38	Semi-precision	2	Not a Number	A
	0x39	Single precision	4	Not a Number	
	0x3a	Double precision	8	Not a Number	
	0x3b – 0x3f	Reserved	-	-	
String	0x40	Reserved	-	-	D
	0x41	Octet string	Defined in first octet	0xff in first octet	
	0x42	Character string	Defined in first octet	0xff in first octet	
	0x43 – 0x47	Reserved	-	-	
Array	0x48 – 0x4f	Reserved	-	-	-
List	0x50 – 0x57	Reserved	-	-	-
Reserved	0x58 – 0xdf	-	-	-	-
Time	0xe0	Time of day	4	0xffffffff	A
	0xe1	Date	4	0xffffffff	
	0xe2 – 0xe7	Reserved	-	-	

Table 2.14 Data Types (Continued)

Identifier	0xe8	Cluster ID	2	0xffff	D
	0xe9	Attribute ID	2	0xffff	
	0xea	BACnet OID	4	0xffffffff	
	0xeb – 0xef	Reserved	-	-	
Miscellaneous	0xf0	IEEE address	8	0xffffffffffffffff	D
	0xf1 – 0xfe	Reserved	-	-	-
Unknown	0xff	Unknown	0	-	-

2.5.2.1 No Data Type

The no data type is a special type to represent an attribute with no associated data.

2.5.2.2 General Data (8-bit, 16-bit, 24-bit and 32-bit)

This type has no rules about its use, and may be used when a data element is needed but its use does not conform to any of the standard types.

2.5.2.3 Boolean

The Boolean type represents a logical value, either TRUE (0x00) or FALSE (0x01). The value 0xff represents an invalid value of this type. All other values of this type are forbidden.

2.5.2.4 Bitmap (8-bit, 16-bit, 24-bit and 32-bit)

The Bitmap type holds 8, 16, 24 or 32 logical values, one per bit, depending on its length. There is no value to represent an invalid value of this type.

2.5.2.5 Unsigned Integer (8-bit, 16-bit, 24-bit and 32-bit)

This type represents an unsigned integer with a decimal range of 0 to 2^8-1, 0 to 2^{16}-1, 0 to 2^{24}-1, or 0 to 2^{32}-1, depending on its length. The values that represents an invalid value of this type are 0xff, 0xffff, 0xffffff and 0xffffffff respectively.

2.5.2.6 Signed Integer (8-bit, 16-bit, 24-bit and 32-bit)

This type represents a signed integer with a decimal range of -(2^7-1) to 2^7-1, -(2^{15}-1) to 2^{15}-1, -(2^{23}-1) to 2^{23}-1, or -(2^{31}-1) to 2^{31}-1, depending on its length. The values that represents an invalid value of this type are 0x80, 0x8000, 0x800000 and 0x80000000 respectively.

2.5.2.7 Enumeration (8-bit)

The Enumeration type represents an index into a lookup table to determine the final value. The value 0xff represents an invalid value of this type.

2.5.2.8 Semi-precision

The ZigBee semi-precision number format is based on the IEEE 754 standard for binary floating-point arithmetic [B4]. This number format should be used very sparingly, when absolutely necessary, keeping in mind the code and processing required supporting it.

The value is calculated as:

$$\text{Value} = -1^{\text{Sign}} * (\text{Hidden} + \text{Mantissa}/1024) * 2^{(\text{Exponent-15})}$$

	Sign	**Exponent**					**Hidden**		**Mantissa**									
	S	E_4	E_3	E_2	E_1	E_0	H	.	M_9	M_8	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0
bit	15					10			9									0

Figure 2.27 Format of the ZigBee Semi-precision Number

Note: The transmission order for the format in Figure 2.27 is bit 0 first.

For normalized numbers ($>2^{-14}$), the hidden bit = 1 and the resolution is constant at 11 bits (1 in 2048).

For un-normalized numbers, the hidden bit = 0. Note that this does not maintain 11-bit resolution and that the resolution becomes coarser as the number gets smaller.

The hidden bit is not sent over the link. It shall have the value ‘1’ (i.e. normalized) in order to be classified as a ZigBee semi-precision number.

The sign bit is set to 0 for positive values, 1 for negative.

The exponent is 5 bits. The actual exponent of 2 is calculated as (exponent – 15).

Certain values are reserved for specific purposes:

- **Not a Number**: this is used for undefined values (e.g. at switch-on and before initialization) and is indicated by an exponent of 31 with a non-zero mantissa.
- **Infinity:** this is indicated by an exponent of 31 and a zero mantissa. The sign bit indicates whether this represents + infinity or – infinity, the figure of 0x7c00 representing +¥ and 0xfc00 representing -¥.

- **Zero:** this is indicated by both a zero exponent and zero mantissa. The sign bit indicates whether this is + or – zero, the value 0x0000 representing +zero and 0x8000 representing –zero.
- **Un-normalised numbers:** numbers < 2^{-14} are indicated by a value of 0 for the exponent. The hidden bit is set to zero.

The maximum value represented by the mantissa is 0x3ff / 1024. The largest number that can be represented is therefore:

$-1^{Sign} * (1 +1023/1024) * 2^{(30 -15)} = -1.9990234 * 32768 = -65504$

Certain applications may choose to scale this value to allow representation of larger values (with a correspondingly more coarse resolution). For details, see the relevant device descriptions.

For example, a value of +2 is represented by $+2^{(16-15)} * 1.0$ = 0x4000, while a value of –2 is represented by 0xc000.

Similarly, a value of +0.625 is represented by $+2^{(17-15)} * 1.625$ = 0x4680, while –0.625 is represented by 0xc680.

2.5.2.9 Single Precision

The format of the single precision data type is based on the IEEE 754 standard for binary floating-point arithmetic [B4]. This number format should be used very sparingly, when absolutely necessary, keeping in mind the code and processing required supporting it.

The format and interpretation of values of this data type follow the same rules as given for the semi-precision data type, but with longer sub-fields, as follows.

Length of mantissa = 23 bits, length of exponent = 8 bits

For further details, see [B4].

2.5.2.10 Double Precision

The format of the double precision data type is based on the IEEE 754 standard for binary floating-point arithmetic [B4]. This number format should be used very sparingly, when absolutely necessary, keeping in mind the code and processing required supporting it.

The format and interpretation of values of this data type follow the same rules as given for the semi-precision data type, but with longer sub-fields, as follows.

Length of mantissa = 52 bits, length of exponent = 11 bits

For further details, see [B4].

2.5.2.11 Octet String

The octet string data type contains data in an application-defined format, not defined in this specification. The octet string data type shall be formatted as illustrated in Figure 2.28.

Octets: 1	Variable
Octet count	Octet data

Figure 2.28 Format of the Octet String Type

The octet count sub-field is one octet in length and specifies the number of octets contained in the octet data sub-field.

Setting this sub-field to 0x00 represents an octet string with no octet data (an "empty string"). Setting this sub-field to 0xff represents an invalid octet string value. In both cases the octet data sub-field has zero length.

The octet data sub-field is *n* octets in length, where *n* is the value of the octet count sub-field. This sub-field contains the application-defined data.

2.5.2.12 Character String

The character string data type contains data octets encoding characters according to the language and character set field of the complex descriptor. The character string data type shall be formatted as illustrated in Figure 2.29.

Octets: 1	Variable
Character count	Character data

Figure 2.29 Format of the Character String Type

The character count sub-field is one octet in length and specifies the number of characters, encoded according to the language and character set field of the complex descriptor (see [B1]), contained in the character data sub-field.

Setting this sub-field to 0x00 represents a character string with no character data (an "empty string"). Setting this sub-field to 0xff represents an invalid character string value. In both cases the character data sub-field has zero length.

The character data sub-field is *e***n* octets in length, where *e* is the size of the character, as specified by the language and character set field of the complex descriptor, and *n* is the value of the character count sub-field. This sub-field contains the encoded characters that comprise the desired character string.

A character string with no contents, i.e. with the character count sub-field equal to 0x00 and a zero length character data sub-field, shall be referred to as an 'empty string'.

2.5.2.13 Time of Day

The Time of Day data type shall be formatted as illustrated in Figure 2.30.

Octets: 1	**1**	**1**	**1**
Hours	Minutes	Seconds	Hundredths

Figure 2.30 Format of the Time of Day Type

The hours subfield represents hours according to a 24 hour clock. The range is from 0 to 23.

The minutes subfield represents minutes of the current hour. The range is from 0 to 59.

The seconds subfield represents seconds of the current minute. The range is from 0 to 59.

The hundredths subfield represents 100ths of the current second. The range is from 0 to 99.

A value of 0xff in any subfield indicates an unused subfield. If all subfields have the value 0xff, this indicates an invalid or 'don't care' value of the data type.

2.5.2.14 Date

The Time of day data type shall be formatted as illustrated in Figure 2.31.

Octets: 1	**1**	**1**	**1**
Year - 1900	Month	Day of month	Day of week

Figure 2.31 Format of the Date Type

The year - 1900 subfield has a range of 0 to 255, representing years from 1900 to 2155.

The month subfield has a range of 1 to 12, representing January to December.

The day of month subfield has a range of 1 to 31. Note that values in the range 29 to 31 may be invalid, depending on the month and year.

The day of week subfield has a range of 1 to 7, representing Monday to Sunday.

A value of 0xff in any subfield indicates an unused subfield. If all subfields have the value 0xff, this indicates an invalid or 'don't care' value of the data type.

2.5.2.15 Cluster ID

This type represents a cluster identifier as defined in 2.5.1.3.

2.5.2.16 Attribute ID

This type represents an attribute identifier as defined in 2.5.1.4.

2.5.2.17 BACnet OID (Object Identifier)

The BACnet OID data type is included to allow interworking with BACnet (see [B5]). The format is described in the referenced standard.

2.5.3 Status Enumerations

Where a ZCL command contains a status field, the actual value of the enumerated status values are listed in Table 2.15.

Table 2.15 Enumerated Status Values Used in the ZCL

Enumerated Status	Value	Description
SUCCESS	0x00	Operation was successful.
FAILURE	0x01	Operation was not successful.
-	0x02 – 0x7f	Reserved.
MALFORMED_COMMAND	0x80	The command appears to contain the wrong fields, as detected either by the presence of one or more invalid field entries or by there being missing fields. Command not carried out. Implementer has discretion as to whether to return this error or INVALID_FIELD.
UNSUP_CLUSTER_COMMAND	0x81	The specified general ZCL command is not supported on the device. Command not carried out.
UNSUP_GENERAL_COMMAND	0x82	The specified cluster command is not supported on the device.

Table 2.15 Enumerated Status Values Used in the ZCL (Continued)

Enumerated Status	Value	Description
UNSUP_MANUF_CLUSTER_COMMAND	0x83	A manufacturer specific unicast, cluster specific command was received with an unknown manufacturer code, or the manufacturer code was recognized but the command is not supported.
UNSUP_MANUF_GENERAL_COMMAND	0x84	A manufacturer specific unicast, ZCL specific command was received with an unknown manufacturer code, or the manufacturer code was recognized but the command is not supported.
INVALID_FIELD	0x85	At least one field of the command contains an incorrect value, according to the specification the device is implemented to.
UNSUPPORTED_ATTRIBUTE	0x86	The specified attribute does not exist on the device.
INVALID_VALUE	0x87	Out of range error, or set to a reserved value. Attribute keeps its old value. Note that an attribute value may be out of range if an attribute is related to another, e.g. with minimum and maximum attributes. See the individual attribute descriptions for specific details.
READ_ONLY	0x88	Attempt to write a read only attribute.
INSUFFICIENT_SPACE	0x89	An operation (e.g. an attempt to create an entry in a table) failed due to an insufficient amount of free space available.
DUPLICATE_EXISTS	0x8a	An attempt to create an entry in a table failed due to a duplicate entry already being present in the table.
NOT_FOUND	0x8b	The requested information (e.g. table entry) could not be found.
UNREPORTABLE_ATTRIBUTE	0x8c	Periodic reports cannot be issued for this attribute.
INVALID_DATA_TYPE	0x8d	The data type given for an attribute is incorrect. Command not carried out.
-	0x8e – 0xbf	Reserved
HARDWARE_FAILURE	0xc0	An operation was unsuccessful due to a hardware failure.

Table 2.15 Enumerated Status Values Used in the ZCL (Continued)

Enumerated Status	Value	Description
SOFTWARE_FAILURE	0xc1	An operation was unsuccessful due to a software failure.
CALIBRATION_ERROR	0xc2	An error occurred during calibration.
-	0xc3 – 0xff	Reserved.

2.6 Functional Description

2.6.1 Transmission

ZCL frames are transmitted via the APS sub-layer by issuing the APSDE-DATA.request primitive.

2.6.2 Reception

ZCL frames are received via the APS sub-layer by the reception of the APSDE-DATA.indication primitive.

On receipt of a command, the device shall attempt to parse and execute the command, and, if required, generate a response to it. Responses are detailed in the specification of each command. If there is no response specified for a particular set of circumstances, (e.g. if the command is not recognized, or the command has succeeded but there is no response specified to indicate success), the default response command shall be generated, taking into account the conditions in 2.4.12.2. The status code returned by the default response command shall be one of the status enumerations listed in Table 2.15.

On receipt of a frame containing a broadcast endpoint (0xff), the APS sub-layer shall direct the frame payload to each active endpoint, through its APSDE-DATA.indication primitive.

2.6.3 Manufacturer Specific Extensions

Manufacturers are free to extend a standard profile in the following ways:

- Add manufacturer specific clusters to a standard profile.
- Add manufacturer specific commands to a standard cluster.
- Add manufacturer specific attributes to a standard cluster.

All communications regarding manufacturer specific extensions shall be transmitted with the manufacturer specific sub-field of the frame control field set to 1 and the manufacturer code included in the frame.

If the manufacturer code in a command frame is not recognized, the command is not carried out.

CHAPTER 3

GENERAL SPECIFICATION

3.1 General Description

3.1.1 Introduction

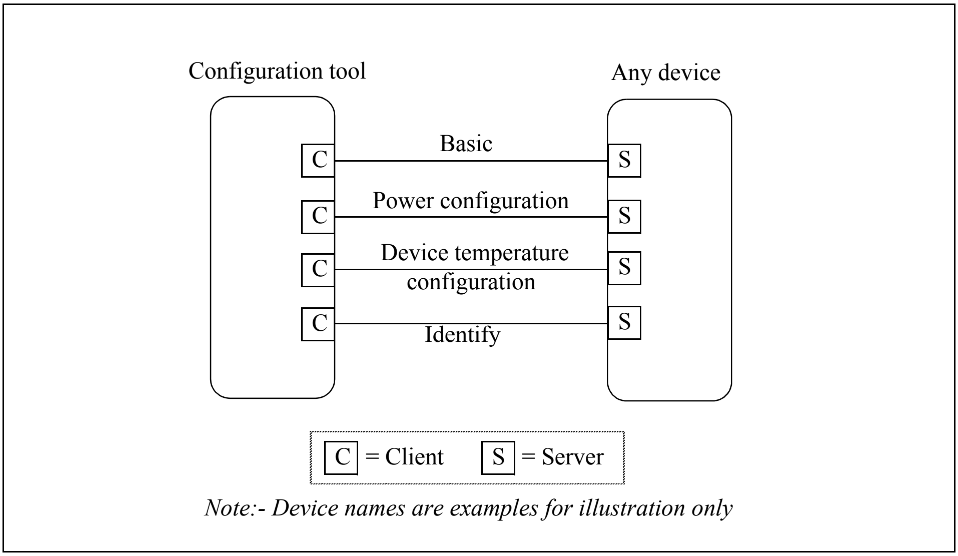
The clusters specified in this document are included here because they are sufficiently general to be of use across a wide range of application domains.

3.1.2 Cluster List

The clusters defined in this document are listed in Table 3.1 to Table 3.5.

Table 3.1 Device Configuration and Installation Clusters

Cluster Name	Description
Basic	Attributes for determining basic information about a device, setting user device information such as description of location, and enabling a device.
Power configuration	Attributes for determining more detailed information about a device's power source(s), and for configuring under/over voltage alarms.
Device temperature configuration	Attributes for determining information about a device's internal temperature, and for configuring under/over temperature alarms.
Identify	Attributes and commands for putting a device into Identification mode (e.g. flashing a light)



Note:- Device names are examples for illustration only

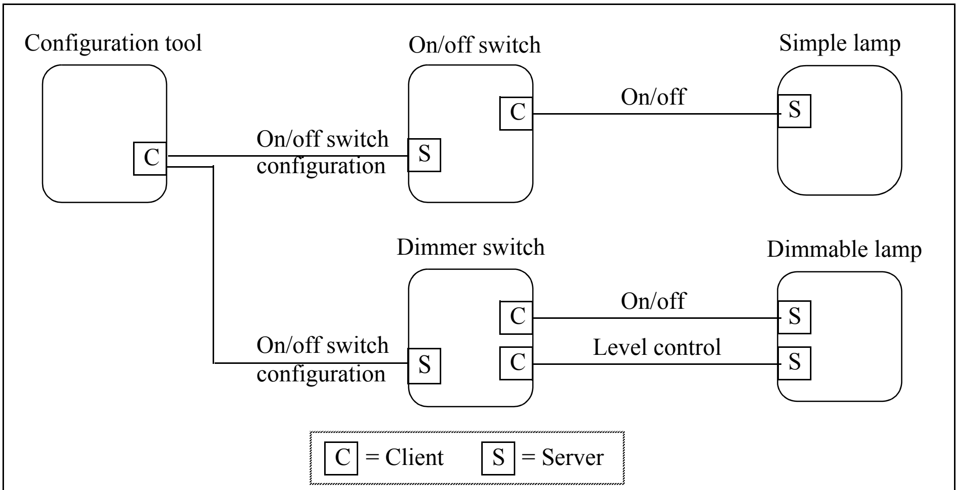
Figure 3.1 Typical Usage of Device Configuration and Installation Clusters

Table 3.2 Groups and Scenes Clusters

Cluster Name	Description
Groups	Attributes and commands for allocating a device to one or more of a number of groups of devices, where each group is addressable by a group address.
Scenes	Attributes and commands for setting up and recalling a number of scenes for a device. Each scene corresponds to a set of stored values of specified device attributes.

Table 3.3 On/Off and Level Control Clusters

Cluster Name	Description
On/off	Attributes and commands for switching devices between 'On' and 'Off' states.
On/off switch configuration	Attributes and commands for configuring on/off switching devices
Level control	Attributes and commands for controlling a characteristic of devices that can be set to a level between fully 'On' and fully 'Off'.



Note:- Device names are examples for illustration only

Figure 3.2 Typical Usage of On / Off and Level Control Clusters

Table 3.4 Alarms Cluster

Cluster Name	Description
Alarms	Attributes and commands for sending alarm notifications and configuring alarm functionality.

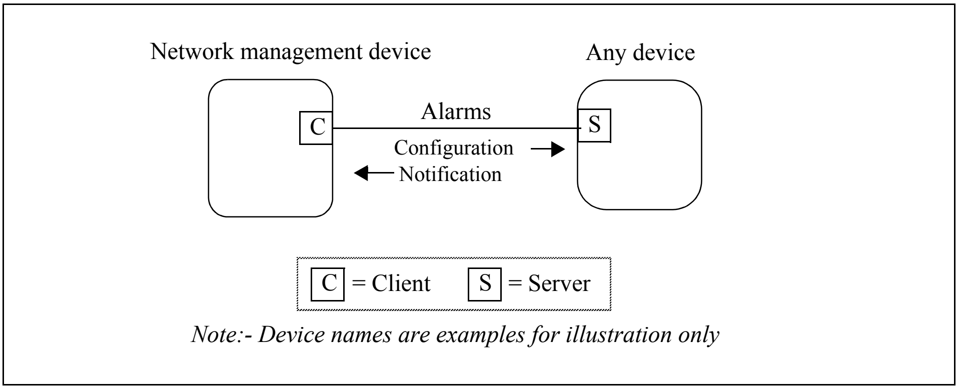

Network management device
Any device
C
Alarms
S
Configuration
Notification
C = Client
S = Server
Note:- Device names are examples for illustration only

Figure 3.3 Typical Usage of the Alarm Clusters

Table 3.5 Other Clusters

Cluster Name	Description
Time	Attributes and commands that provide a basic interface to a real-time clock.
RSSI Location	Attributes and commands that provide a means for exchanging location information and channel parameters among devices.

3.2 Basic Cluster

3.2.1 Overview

Attributes and commands for determining basic information about a device, setting user device information such as location, enabling a device and resetting it to factory defaults.

Note: Where a physical ZigBee node supports multiple endpoints it will often be the case that many of these settings will apply to the whole node, that is they are the same for every endpoint on the device. In such cases they can be implemented once for the node, and mapped to each endpoint.

3.2.2 Server

3.2.2.1 Dependencies

For the alarms functionality of this cluster to be operational, the Alarms cluster server shall be implemented on the same endpoint.

3.2.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.6.

Table 3.6 General Attribute Sets

Attribute Set IDentifier	Description
0x000	Basic Device Information
0x001	Basic Device Settings
0x002 – 0xfff	Reserved

3.2.2.2.1 Basic Device Information Attribute Set

The Basic Device Information attribute set contains the attributes summarized in Table 3.7.

Table 3.7 Attributes of the Basic Device Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	ZCLVersion	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	M
0x0001	ApplicationVersion	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	O
0x0002	StackVersion	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	O
0x0003	HWVersion	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	O
0x0004	ManufacturerName	Character string	0 – 32 bytes	Read only	Empty string	O

Table 3.7 Attributes of the Basic Device Information Attribute Set

0x0005	ModelIdentifier	Character string	0 – 32 bytes	Read only	Empty string	O
0x0006	DateCode	Character string	0 – 16 bytes	Read only	Empty string	O
0x0007	PowerSource	8-bit Enumeration	0x00 – 0xff	Read only	0x00	M

3.2.2.2.2 *ZCLVersion* Attribute

The *ZCLVersion* attribute is 8-bits in length and specifies the version number of the ZigBee Cluster Library that all clusters on this endpoint conform to. For the initial version of the ZCL, this attribute shall be set to 0x01.

Note: It is strongly recommended that new functionality is added to the ZCL either in the form of new clusters or by addition of optional attributes and optional commands to existing clusters. New functionality should be added in an 'orthogonal' way, making use of existing clusters to perform the functions they offer rather than re-implementing these functions as part of the new clusters, so that devices whose functionality has been extended via the new clusters can still interwork with devices using existing clusters. When increasing the version of the ZCL, no changes should be made to the functionality of individual clusters that prevent interworking with previous versions of the same cluster.

3.2.2.2.3 *ApplicationVersion* Attribute

The *ApplicationVersion* attribute is 8-bits in length and specifies the version number of the application software contained in the device. The usage of this attribute is manufacturer dependent.

3.2.2.2.4 *StackVersion* Attribute

The *StackVersion* attribute is 8-bits in length and specifies the version number of the implementation of the ZigBee stack contained in the device. The usage of this attribute is manufacturer dependent.

3.2.2.2.5 *HWVersion* Attribute

The *HWVersion* attribute is 8-bits in length and specifies the version number of the hardware of the device. The usage of this attribute is manufacturer dependent..

3.2.2.2.6 *ManufacturerName* Attribute

The *ManufacturerName* attribute is a maximum of 32 bytes in length and specifies the name of the manufacturer as a ZigBee character string.

3.2.2.2.7 *ModelIdentifier* Attribute

The *ModelIdentifier* attribute is a maximum of 32 bytes in length and specifies the model number (or other identifier) assigned by the manufacturer as a ZigBee character string.

3.2.2.2.8 *DateCode* Attribute

The *DateCode* attribute is a ZigBee character string with a maximum length of 16 bytes . The first 8 characters specify the date of manufacturer of the device in international date notation according to ISO 8601, i.e. YYYYMMDD, e.g. 20060814.

The final 8 characters may include country, factory, line, shift or other related information at the option of the manufacturer. The format of this information is manufacturer dependent.

3.2.2.2.9 *PowerSource* Attribute

The *PowerSource* attribute is 8-bits in length and specifies the source(s) of power available to the device. Bits b_0_b_6 of this attribute represent the primary power source of the device and bit b_7 indicates whether the device has a secondary power source in the form of a battery backup.

Bits b_0_b_6 of this attribute shall be set to one of the non-reserved values listed in Table 3.8.

Table 3.8 Values of the *PowerSource* Attribute

PowerSource **Attribute Value b_6-b_0**	**Description**
0x00	Unknown
0x01	Mains (single phase)
0x02	Mains (3 phase)
0x03	Battery
0x04	DC source
0x05	Emergency mains constantly powered
0x06	Emergency mains and transfer switch
0x07 – 0x7f	Reserved

Bit b_7 of this attribute shall be set to 1 if the device has a secondary power source in the form of a battery backup. Otherwise, bit b_7 shall be set to 0.

3.2.2.2.10 Basic Device Settings Attribute Set

The device configuration attribute set contains the attributes summarized in Table 3.9.

Table 3.9 Attributes of the Device Configuration Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	LocationDescription	Character string	0 – 16 bytes	Read/ write	Empty string	O
0x0011	PhysicalEnvironment	8-bit Enumeration	0x00 – 0xff	Read/ write	0x00	O
0x0012	DeviceEnabled	Boolean	0x00 – 0x01	Read/ write	0x01	M
0x0013	AlarmMask	8-bit Bitmap	000000xx	Read/ write	0x00	O

3.2.2.2.11 *LocationDescription* Attribute

The *LocationDescription* attribute is a maximum of 16 bytes in length and describes the physical location of the device as a ZigBee character string. This location description may be added into the device during commissioning.

3.2.2.2.12 *PhysicalEnvironment* Attribute

The *PhysicalEnvironment* attribute is 8-bits in length and specifies the type of physical environment in which the device will operate. This attribute shall be set to one of the non-reserved values listed in Table 3.10.

Table 3.10 Values of the PhysicalEnvironment Attribute

PhysicalEnvironment Attribute Value	Description
0x00	Unspecified environment
0x01 – 0x7f	Specified per Profile
0x80 – 0xfe	Reserved
0xff	Unknown environment

3.2.2.2.13 *DeviceEnable* Attribute

The *DeviceEnabled* attribute is a boolean and specifies whether the device is enabled or disabled. This attribute shall be set to one of the non-reserved values listed in Table 3.11.

Table 3.11 Values of the *DeviceEnable* Attribute

***DeviceEnable* Attribute Value**	**Description**
0x00	Disabled
0x01	Enabled

'Disabled' means that the device does not send or respond to application level commands, other than commands to read or write attributes. Values of attributes which depend on the operation of the application may be invalid, and any functionality triggered by writing to such attributes may be disabled. ZigBee networking functionality remains operational.

Note that the identify cluster cannot be disabled, and remains functional regardless of this setting.

3.2.2.2.14 *AlarmMask* Attribute

The *AlarmMask* attribute is 8-bits in length and specifies which of a number of general alarms may be generated, as listed in Table 3.12. A ‘1’ in each bit position enables the associated alarm.

Table 3.12 Values of the *AlarmMask* Attribute

***AlarmMask* Attribute Bit Number**	**Alarm Code**	**Alarm**
0	0	General hardware fault
1	1	General software fault
2 – 7	-	Reserved

These alarms are provided as basic alarms that a device may use even if no other clusters with alarms are present on the device.

3.2.2.3 Commands Received

The command IDs for the Basic cluster are listed in Table 3.13.

Table 3.13 Received Command IDs for the Basic Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Reset to Factory Defaults	0
0x01 – 0xff	Reserved	-

3.2.2.4 Commands Generated

No commands are generated by the server cluster.

3.2.2.4.1 Reset to Factory Defaults Command

This command does not have a payload.

3.2.2.4.2 Effect on Receipt

On receipt of this command, the device resets all the attributes of all its clusters to their factory defaults.

Note that ZigBee networking functionality and any bindings are not affected by this command.

3.2.3 Client

3.2.3.1 Dependencies

None

3.2.3.2 Attributes

The Client cluster has no attributes.

3.2.3.3 Commands Received

No cluster specific commands are received by the client cluster.

3.2.3.4 Commands Generated

The cluster specific commands generated by the client cluster are those received by the server, as required by the application.

3.3 Power Configuration Cluster

3.3.1 Overview

Attributes for determining detailed information about a device's power source(s), and for configuring under/over voltage alarms.

3.3.2 Server

3.3.2.1 Dependencies

Any endpoint that implements this server cluster shall also implement the Basic server cluster.

For the alarm functionality described in this cluster to be operational, any endpoint that implements the Power Configuration server cluster must also implement the Alarms server cluster (see sub-clause 3.11).

3.3.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.14.

Table 3.14 Power Configuration Attribute Sets

Attribute Set Identifier	Description
0x000	Mains Information
0x001	Mains Settings
0x002	Battery Information
0x003	Battery Settings
0x004 – 0xfff	Reserved

3.3.2.2.1 Mains Information Attribute Set

The Mains Information attribute set contains the attributes summarized in Table 3.15.

Table 3.15 Attributes of the Mains Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MainsVoltage	Unsigned 16-bit integer	0x0000 – 0xffff	Read only	-	O
0x0001	MainsFrequency	Unsigned 8-bit integer	0x00 – 0xff	Read only	-	O

3.3.2.2.1.1 *MainsVoltage* Attribute

The *MainsVoltage* attribute is 16-bits in length and specifies the actual (measured) RMS voltage (or DC voltage in the case of a DC supply) currently applied to the device, measured in units of 100mV.

3.3.2.2.1.2 *MainsFrequency* Attribute

The *MainsFrequency* attribute is 8-bits in length and represents the frequency, in Hertz, of the mains as determined by the device as follows:-

MainsFrequency = 0.5 x measured frequency

Where 2 Hz <= measured frequency <= 506 Hz, corresponding to a *MainsFrequency* in the range 1 to 0xfd.

The maximum resolution this format allows is 2 Hz.

The following special values of *MainsFrequency* apply.

0x00 indicates a frequency that is too low to be measured.

0xfe indicates a frequency that is too high to be measured.

0xff indicates that the frequency could not be measured.

In the case of a DC supply, this attribute shall also have the value zero.

3.3.2.2.2 Mains Settings Attribute Set

The Mains Settings attribute set contains the attributes summarized in Table 3.16.

Table 3.16 Attributes of the Mains Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	MainsAlarmMask	Bitmap (8-bits)	0000 00xx	Read/ write	0000 0000	O
0x0011	MainsVoltageMinThreshold	Unsigned 16-bit integer	0x0000 – 0xffff	Read/ write	0x0000	O
0x0012	MainsVoltageMaxThreshold	Unsigned 16-bit integer	0x0000 – 0xffff	Read/ write	0xffff	O
0x0013	MainsVoltageDwellTripPoint	Unsigned 16-bit integer	0x0000 – 0xffff	Read/ write	0x0000	O

The alarm settings in this table require the Alarms cluster to be implemented on the same device - see Dependencies. If the Alarms cluster is not present on the same device they may be omitted.

3.3.2.2.2.1 *MainsAlarmMask* Attribute

The *MainsAlarmMask* attribute is 8-bits in length and specifies which mains alarms may be generated, as listed in Table 3.17. A ‘1’ in each bit position enables the alarm.

Table 3.17 Values of the *MainsAlarmMask* Attribute

MainsAlarmMask Attribute Bit Number	Alarm
0	Mains Voltage too low (7.2.2.2.2)
1	Mains Voltage too high (7.2.2.2.3)
2 – 7	Reserved

3.3.2.2.2.2 *MainsVoltageMinThreshold* Attribute

The *MainsVoltageMinThreshold* attribute is 16-bits in length and specifies the lower alarm threshold, measured in units of 100mV, for the *MainsVoltage* attribute. The value of this attribute shall be less than *MainsVoltageMaxThreshold.*

If the value of *MainsVoltage* drops below the threshold specified by *MainsVoltageMinThreshold*, the device shall start a timer to expire after *MainsVoltageDwellTripPoint* seconds. If the value of this attribute increases to greater than or equal to *MainsVoltageMinThreshold* before the timer expires, the device shall stop and reset the timer. If the timer expires, an alarm shall be generated.

The Alarm Code field (see 3.11.2.3.1) included in the generated alarm shall be 0x00.

If this attribute takes the value 0xffff then this alarm shall not be generated.

3.3.2.2.2.3 MainsVoltageMaxThreshold **Attribute**

The *MainsVoltageMaxThreshold* attribute is 16-bits in length and specifies the upper alarm threshold, measured in units of 100mV, for the *MainsVoltage* attribute. The value of this attribute shall be greater than *MainsVoltageMinThreshold.*

If the value of *MainsVoltage* rises above the threshold specified by *MainsVoltageMaxThreshold*, the device shall start a timer to expire after *MainsVoltageDwellTripPoint* seconds. If the value of this attribute drops to lower than or equal to *MainsVoltageMaxThreshold* before the timer expires, the device shall stop and reset the timer. If the timer expires, an alarm shall be generated.

The Alarm Code field (see 3.11.2.3.1) included in the generated alarm shall be 0x01.

If this attribute takes the value 0xffff then this alarm shall not be generated.

3.3.2.2.2.4 ***MainsVoltageDwellTripPoint*** **Attribute**

The *MainsVoltageDwellTripPoint* attribute is 16-bits in length and specifies the length of time, in seconds that the value of *MainsVoltage* may exist beyond either of its thresholds before an alarm is generated.

If this attribute takes the value 0xffff then the associated alarms shall not be generated.

3.3.2.2.3 Battery Information Attribute Set

The Battery Information attribute set contains the attributes summarized in Table 3.18.

Table 3.18 Attributes of the Battery Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0020	BatteryVoltage	Unsigned 8-bit integer	0x00 – 0xff	Read only	-	O

3.3.2.2.3.1 *BatteryVoltage* Attribute

The *BatteryVoltage* attribute is 8-bits in length and specifies the current actual (measured) battery voltage, in units of 100mV.

The value 0xff indicates an invalid or unknown reading.

3.3.2.2.4 Battery Settings Attribute Set

Table 3.19 Attributes of the Battery Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0030	BatteryManufacturer	Character string	0 – 16 bytes	Read/write	Empty string	O
0x0031	BatterySize	8-bit Enumeration	0x00 – 0xff	Read/write	0xff	O
0x0032	BatteryAHrRating	Unsigned 16-bit integer	0x0000 – 0xffff	Read/write	-	O
0x0033	BatteryQuantity	Unsigned 8-bit integer	0x00 – 0xff	Read/write	-	O
0x0034	BatteryRatedVoltage	Unsigned 8-bit integer	0x00 – 0xff	Read/write	-	O
0x0035	BatteryAlarmMask	Bitmap (8-bits)	0000 000x	Read/write	0000 0000	O
0x0036	BatteryVoltageMin Threshold	Unsigned 8-bit integer	0x00 – 0xff	Read/write	0x0000	O

3.3.2.2.4.1 *BatteryManufacturer* Attribute

The *BatteryManufacturer* attribute is a maximum of 16 bytes in length and specifies the name of the battery manufacturer as a ZigBee character string.

3.3.2.2.4.2 *BatterySize* Attribute

The *BatterySize* attribute is an enumeration which specifies the type of battery being used by the device. This attribute shall be set to one of the non-reserved values listed in Table 3.20.

Table 3.20 Values of the *BatterySize* Attribute

BatterySize Attribute Value	Description
0x00	No battery
0x01	Built in
0x02	Other
0x03	AA

Table 3.20 Values of the *BatterySize* Attribute

BatterySize Attribute Value	Description
0x04	AAA
0x05	C
0x06	D
0x07 – 0xfe	Reserved
0xff	Unknown

3.3.2.2.4.3 *BatteryAHrRating* Attribute

The *BatteryAHrRating* attribute is 16-bits in length and specifies the Ampere-hour rating of the battery, measured in units of 10mAHr.

3.3.2.2.4.4 *BatteryQuantity* Attribute

The *BatteryQuantity* attribute is 8-bits in length and specifies the number of battery cells used to power the device.

3.3.2.2.4.5 *BatteryRatedVoltage* Attribute

The *BatteryRatedVoltage* attribute is 8-bits in length and specifies the rated voltage of the battery being used in the device, measured in units of 100mV.

3.3.2.2.4.6 *BatteryAlarmMask* Attribute

The *BatteryAlarmMask* attribute is 8-bits in length and specifies which mains alarms may be generated, as listed in Table 3.21. A ‘1’ in each bit position enables the alarm.

Table 3.21 Values of the MainsAlarmMask Attribute

MainsAlarmMask Attribute Bit Number	Alarm
0	Battery voltage too low (7.2.2.4.7)
1 – 7	Reserved

3.3.2.2.4.7 *BatteryVoltageMinThreshold* Attribute

The *BatteryVoltageMinThreshold* attribute is 8-bits in length and specifies the low voltage alarm threshold, measured in units of 100mV, for the *BatteryVoltage* attribute.

If the value of *BatteryVoltage* drops below the threshold specified by *BatteryVoltageMinThreshold* an alarm shall be generated.

The Alarm Code field (see 3.11.2.3.1) included in the generated alarm shall be 0x10.

If this attribute takes the value 0xff then this alarm shall not be generated.

3.3.2.3 Commands Received

No commands are received by the server.

3.3.2.4 Commands Generated

The server generates no commands.

3.3.3 Client

3.3.3.1 Dependencies

None

3.3.3.2 Attributes

The client has no attributes.

3.3.3.3 Commands Received

No cluster specific commands are received by the client.

3.3.3.4 Commands Generated

No cluster specific commands are generated by the client.

3.4 Device Temperature Configuration Cluster

3.4.1 Overview

Attributes for determining information about a device’s internal temperature, and for configuring under/over temperature alarms for temperatures that are outside the device's operating range.

3.4.2 Server

3.4.2.1 Dependencies

For the alarm functionality described in this cluster to be operational, any endpoint that implements the Device Temperature Configuration server cluster shall also implement the Alarms server cluster (see 3.11).

3.4.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.22.

Table 3.22 Device Temperature Configuration Attribute Sets

Attribute Set Identifier	Description
0x000	Device Temperature Information
0x001	Device Temperature Settings
0x002 – 0xfff	Reserved

3.4.2.2.1 Device Temperature Information Attribute Set

The Device Temperature Information attribute set contains the attributes summarized in Table 3.23.

Table 3.23 Attributes of the Device Temperature Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	CurrentTemperature	Signed 16-bit integer	-200 to +200	Read only	-	M
0x0001	MinTempExperienced	Signed 16-bit integer	-200 to +200	Read only	-	O
0x0002	MaxTempExperienced	Signed 16-bit integer	-200 to +200	Read only	-	O
0x0003	OverTempTotalDwell	Unsigned 16-bit integer	0x0000 – 0xffff	Read only	0	O

3.4.2.2.1.1 *CurrentTemperature* Attribute

The *CurrentTemperature* attribute is 16 bits in length and specifies the current internal temperature, in degrees Celsius, of the device. This attribute shall be specified in the range –200 to +200.

The value 0xffff indicates an invalid reading.

3.4.2.2.1.2 *MinTempExperienced* Attribute

The *MinTempExperienced* attribute is 16 bits in length and specifies the minimum internal temperature, in degrees Celsius, the device has experienced while powered. This attribute shall be specified in the range –200 to +200.

The value 0xffff indicates an invalid reading.

3.4.2.2.1.3 *MaxTempExperienced* Attribute

The *MaxTempExperienced* attribute is 16 bits in length and specifies the maximum internal temperature, in degrees Celsius, the device has experienced while powered. This attribute shall be specified in the range –200 to +200.

The value 0xffff indicates an invalid reading.

3.4.2.2.1.4 *OverTempTotalDwell* Attribute

The *OverTempTotalDwell* attribute is 16 bits in length and specifies the length of time, in hours, the device has spent above the temperature specified by the *HighTempThreshold* attribute 3.4.2.2.2.3, cumulative over the lifetime of the device.

The value 0xffff indicates an invalid time.

3.4.2.2.2 Device Temperature Settings Attribute Set

The Device Temperature Settings attribute set contains the attributes summarized in Table 3.24.

Table 3.24 Attributes of the Device Temperature Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	DeviceTempAlarmMask	Bitmap (8-bit)	0000 00xx	Read/ write	0000 0000	O
0x0011	LowTempThreshold	Signed 16-bit integer	-200 to +200	Read/ write	-	O
0x0012	HighTempThreshold	Signed 16-bit integer	-200 to +200	Read/ write	-	O
0x0013	LowTempDwellTripPoint	Unsigned 24-bit integer	0x000000 – 0xffffff	Read/ write	-	O
0x0014	HighTempDwellTripPoint	Unsigned 24-bit integer	0x000000 – 0xffffff	Read/ write	-	O

All attributes in this table require the Alarms cluster to be implemented on the same device - see Dependencies. If the Alarms cluster is not present on the same device they may be omitted.

3.4.2.2.2.1 *DeviceTempAlarmMask* Attribute

The *DeviceTempAlarmMask* attribute is 8-bits in length and specifies which alarms may be generated, as listed in Table 3.25. A ‘1’ in each bit position enables the corresponding alarm.

Table 3.25 Values of the *DeviceTempAlarmMask* Attribute

DeviceTempAlarmMask Attribute Bit Number	Alarm
0	Device Temperature too low (8.2.2.2.2)
1	Device Temperature too high (8.2.2.2.3)
2 – 7	Reserved

3.4.2.2.2.2 *LowTempThreshold* Attribute

The *LowTempThreshold* attribute is 16-bits in length and specifies the lower alarm threshold, measured in degrees Celsius (range -200°C to 200°C), for the

CurrentTemperature attribute. The value of this attribute shall be less than *HighTempThreshold.*

If the value of *CurrentTemperature* drops below the threshold specified by *LowTempThreshold*, the device shall start a timer to expire after *LowTempDwellTripPoint* seconds. If the value of this attribute increases to greater than or equal to *LowTempThreshold* before the timer expires, the device shall stop and reset the timer. If the timer expires, an alarm shall be generated.

The Alarm Code field (see 3.11.2.3.1) included in the generated alarm shall be 0x00.

If this attribute takes the value 0x8000 then this alarm shall not be generated.

3.4.2.2.2.3 *HighTempThreshold* Attribute

The *HighTempThreshold* attribute is 16-bits in length and specifies the upper alarm threshold, measured in degrees Celsius (range -200°C to 200°C), for the *CurrentTemperature* attribute. The value of this attribute shall be greater than *LowTempThreshold.*

If the value of *CurrentTemperature* rises above the threshold specified by *HighTempThreshold*, the device shall start a timer to expire after *HighTempDwellTripPoint* seconds. If the value of this attribute drops to lower than or equal to *HighTempThreshold* before the timer expires, the device shall stop and reset the timer. If the timer expires, an alarm shall be generated.

The Alarm Code field (see 3.11.2.3.1) included in the generated alarm shall be 0x01.

If this attribute takes the value 0x8000 then this alarm shall not be generated.

3.4.2.2.2.4 *LowTempDwellTripPoint* Attribute

The *LowTempDwellTripPoint* attribute is 24-bits in length and specifies the length of time, in seconds, that the value of *CurrentTemperature* may exist below *LowTempThreshold* before an alarm is generated.

If this attribute takes the value 0xffffff then this alarm shall not be generated.

3.4.2.2.2.5 *HighTempDwellTripPoint* Attribute

The *HighTempDwellTripPoint* attribute is 24-bits in length and specifies the length of time, in seconds, that the value of *CurrentTemperature* may exist above *HighTempThreshold* before an alarm is generated.

If this attribute takes the value 0xffffff then this alarm shall not be generated.

3.4.2.3 Commands Received

No commands are received by the server.

3.4.2.4 Commands Generated

The server generates no commands.

3.4.3 Client

3.4.3.1 Dependencies

None

3.4.3.2 Attributes

The client has no attributes.

3.4.3.3 Commands Received

No cluster specific commands are received by the client.

3.4.3.4 Commands Generated

No cluster specific commands are generated by the client.

3.5 Identify Cluster

3.5.1 Overview

Attributes and commands to put a device into an Identification mode (e.g. flashing a light), that indicates to an observer – e.g. an installer - which of several devices it is, also to request any device that is identifying itself to respond to the initiator.

Note that this cluster cannot be disabled, and remains functional regardless of the setting of the *DeviceEnable* attribute in the Basic cluster.

3.5.2 Server

3.5.2.1 Dependencies

None

3.5.2.2 Attributes

The server supports the attribute shown in Table 3.26.

Table 3.26 Attributes of the Identify Server Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	IdentifyTime	Unsigned 16-bit integer	0x0000 – 0xffff	Read / write	0x0000	M

3.5.2.2.1 *IdentifyTime* Attribute

The *IdentifyTime* attribute specifies the remaining length of time, in seconds, that the device will continue to identify itself.

If this attribute is set to a value other than 0x0000 then the device shall enter its identification procedure, in order to indicate to an observer which of several devices it is. It is recommended that this procedure consists of flashing a light with a period of 0.5 seconds. The *IdentifyTime* attribute shall be decremented every second.

If this attribute reaches or is set to the value 0x0000 then the device shall terminate its identification procedure.

3.5.2.3 Commands Received

The server side of the identify cluster is capable of receiving the commands listed in Table 3.27.

Table 3.27 Received Command IDs for the Identify Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Identify	M
0x01	Identify Query	M
0x02 – 0xff	Reserved	

3.5.2.3.1 Identify Command

The identify command starts or stops the receiving device identifying itself.

3.5.2.3.1.1 Payload Format

The identify query response command payload shall be formatted as illustrated in Figure 3.4.

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	Identify Time

Figure 3.4 Format of the Identify Query Response Command Payload

3.5.2.3.1.2 Effect on Receipt

On receipt of this command, the device shall set the *IdentifyTime* attribute to the value of the Identify Time field. This then starts, continues, or stops the device's identification procedure as detailed in 3.5.2.2.1.

3.5.2.3.2 Identify Query Command

The identify query command allows the sending device to request the target or targets to respond if they are currently identifying themselves.

This command has no payload.

3.5.2.3.2.1 Effect on Receipt

On receipt of this command, if the device is currently identifying itself then it shall generate an appropriate Identify Query Response command, see 3.5.2.4.1, and unicast it to the requester. If the device is not currently identifying itself it shall take no further action.

3.5.2.4 Commands Generated

The server side of the identify cluster is capable of generating the commands listed in Table 3.28.

Table 3.28 Generated Command IDs for the Identify Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Identify Query Response	M
0x01 – 0xff	Reserved	

3.5.2.4.1 Identify Query Response Command

The identify query response command is generated in response to receiving an Identify Query command, see 3.5.2.3.1, in the case that the device is currently identifying itself.

3.5.2.4.1.1 Payload Format

The identify query response command payload shall be formatted as illustrated in Figure 3.5.

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	Timeout

Figure 3.5 Format of the Identify Query Response Command Payload

3.5.2.4.1.2 Timeout Field

The Timeout field contains the current value of the *IdentifyTime* attribute, and specifies the length of time, in seconds, that the device will continue to identify itself.

3.5.2.4.1.3 Effect on Receipt

On receipt of this command, the device is informed of a device in the network which is currently identifying itself. This information may be particularly beneficial in situations where there is no commissioning tool. Note that there may be multiple responses.

3.5.3 Client

3.5.3.1 Dependencies

None.

3.5.3.2 Attributes

The client has no attributes.

3.5.3.3 Commands Received

The client receives the cluster specific response commands detailed in 3.5.2.3.

3.5.3.4 Commands Generated

The client generates the cluster specific commands detailed in 3.5.2.4, as required by the application.

3.6 Groups Cluster

3.6.1 Overview

The ZigBee specification provides the capability for group addressing. That is, any endpoint on any device may be assigned to one or more groups, each labeled with a 16-bit identifier (0x0001 – 0xfff7), which acts for all intents and purposes like a network address. Once a group is established, frames, sent using the APSDE-DATA.request primitive and having a DstAddrMode of 0x01, denoting group addressing, will be delivered to every endpoint assigned to the group address named in the DstAddr parameter of the outgoing APSDE-DATA.request primitive on every device in the network for which there are such endpoints.

Management of group membership on each device and endpoint is implemented by the APS, but the over-the-air messages that allow for remote management and commissioning of groups are defined here in the cluster library on the theory that, while the basic group addressing facilities are integral to the operation of the stack, not every device will need or want to implement this management cluster. Furthermore, the placement of the management commands here allows developers of proprietary profiles to avoid implementing the library cluster but still exploit group addressing.

Commands are defined here for discovering the group membership of a device, adding a group, removing a group and removing all groups.

Finally, the group cluster allows application entities to store a name string for each group to which they are assigned and to report that name string in response to a client request.

Note that configuration of group addresses for outgoing commands is achieved using the APS binding mechanisms, and is not part of this cluster.

3.6.2 Server

Each ZigBee device that implements this cluster may be thought of as a group management server in the sense that it responds to information requests and configuration commands regarding the contents of its group table.

Note that, since these commands are simply data frames sent using the APSDE_SAP, they must be addressed with respect to device and endpoint. In particular the destination device and endpoint of a group management command must be unambiguous at the time of the issuance of the primitive either because:

- They are explicitly spelled out in the DstAddr and DstEndpoint parameters of the primitive.

- They are not explicitly spelled out but may be derived from the binding table in the APS of the sending device.
- Broadcast addressing is being employed, either with respect to the device address or the endpoint identifier.
- Group addressing is being employed.

On receipt of a group cluster command, the APS will, at least conceptually, deliver the frame to each destination endpoint spelled out in the addressing portion of the APS header and, again conceptually speaking, the application entity resident at that endpoint will process the command and respond as necessary. From an implementation standpoint, of course, this may be done in a more economical way that does not involve duplication and separate processing, e.g by providing a hook in the APS whereby group cluster commands could be delivered to a special application entity without duplication.

3.6.2.1 Dependencies

For correct operation of the 'Add group if identifying' command, any endpoint that implements the Groups server cluster shall also implement the Identify server cluster.

3.6.2.2 Attributes

The server supports the attribute shown in Table 3.29.

Table 3.29 Attributes of the Groups Server Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	NameSupport	8-bit bitmap	x0000000	Read only	-	M

The most significant bit of the *NameSupport* attribute indicates whether or not group names are supported. A value of 1 indicates that they are supported, and a value of 0 indicates that they are not supported.

3.6.2.2.1 Group Names

Group names are between 0 and 16 characters long. Support of group names is optional, and is indicated by the *NameSupport* attribute. Group names, if supported, must be stored in a separate data structure managed by the application in which the entries correspond to group table entries.

3.6.2.2.2 Commands Received

The groups cluster is concerned with management of the group table on a device. In practice, the group table is managed by the APS and the table itself is available to the next higher layer as an AIB attribute. A command set is defined here and the implementation details of that command set in terms of the facilities provided by the APS is left up to the implementer of the cluster library itself.

The server side of the groups cluster is capable of receiving the commands listed in Table 3.30.

Table 3.30 Received Command IDs for the Groups Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Add group	M
0x01	View group	M
0x02	Get group membership	M
0x03	Remove group	M
0x04	Remove all groups	M
0x05	Add group if identifying	M
0x06 – 0xff	Reserved	-

3.6.2.2.3 Add Group Command

The add group command allows the sending device to add group membership in a particular group for one or more endpoints on the receiving device.

3.6.2.2.3.1 Payload Format

The Add Group command payload shall be formatted as illustrated in Figure 3.6

Octets		Variable
Data Type	Unsigned 16-bit Integer	Character string
Field Name	Group ID	Group Name

Figure 3.6 Format of the Add Group Command Payload

3.6.2.2.3.2 Effect on Receipt

On receipt of this command, the device shall (if possible) add the Group ID and Group Name to its Group Table. It shall then generate an appropriate Add Group Response command indicating success or failure. See 3.6.2.3.1.

3.6.2.2.4 View Group Command

The view group command allows the sending device to request that the receiving entity or entities respond with a view group response command containing the application name string for a particular group.

3.6.2.2.4.1 Payload Format

The View Group command payload shall be formatted as illustrated in Figure 3.7.

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	Group ID

Figure 3.7 Format of the View Group Command Payload

3.6.2.2.4.2 Effect on Receipt

On receipt of this command, the device shall generate an appropriate View Group Response command. 3.6.2.3.2.

3.6.2.2.5 Get Group Membership Command

The get group membership command allows the sending device to inquire about the group membership of the receiving device and endpoint in a number of ways.

3.6.2.2.5.1 Payload Format

The get group membership command payload shall be formatted as illustrated in Figure 3.8.

Octets	1	Variable
Data Type	Unsigned 8-bit Integer	List of 16-bit Integer
Field Name	Group count	Group list

Figure 3.8 Format of the Get Group Membership Command Payload

3.6.2.2.5.2 Effect on Receipt

On receipt of the get group membership command, each receiving entity shall respond, as necessary, with group membership information using the get group membership response frame shown below. An entity shall respond if and only if:

- The group count field of the command frame has a value of 0 indicating that the group list field is empty, or
- The group list field of the command frame contains at least one group of which the entity is a member. In this case the response frame will contain the identifiers of all such groups.

3.6.2.2.6 Remove Group Command

The remove group command allows the sender to request that the receiving entity or entities remove their membership, if any, in a particular group.

Note that if a group is removed the scenes associated with that group should be removed.

3.6.2.2.6.1 Payload Format

The Remove Group command payload shall be formatted as illustrated in Figure 3.9.

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	Group ID

Figure 3.9 Format of the Remove Group Command Payload

3.6.2.2.6.2 Effect on Receipt

On receipt of this command, the device shall (if possible) remove the Group ID and Group Name from its Group Table. It shall then generate an appropriate Remove Group Response command indicating success or failure. See 3.6.2.3.4.

3.6.2.2.7 Remove All Groups Command

The remove all groups command allows the sending device to direct the receiving entity or entities to remove all group associations.

Note that removing all groups necessitates the removal of all associated scenes as well. (Note: scenes not associated with a group need not be removed).

3.6.2.2.7.1 Payload Format

The Remove All Groups command has no payload.

3.6.2.2.7.2 Effect on Receipt

On receipt of this command, the device shall remove all groups on this endpoint from its Group Table.

3.6.2.2.8 Add Group If Identifying Command

The add group if identifying command allows the sending device to add group membership in a particular group for one or more endpoints on the receiving device, on condition that it is identifying itself. Identifying functionality is controlled using the identify cluster, (see 3.5).

This command might be used to assist configuring group membership in the absence of a commissioning tool.

3.6.2.2.8.1 Payload Format

The Add Group If Identifying command payload shall be formatted as illustrated in Figure 3.10.

Octets	2	Variable
Data Type	Unsigned 16-bit Integer	Character string
Field Name	Group ID	Group Name

Figure 3.10 Format of the Add Group If Identifying Command Payload

3.6.2.2.8.2 Effect on Receipt

On receipt of this command, the device shall first check whether it is currently identifying itself. If so then the device shall (if possible) add the Group ID and Group Name to its Group Table. If the device it not currently identifying itself then no action shall be taken.

No response is defined as this command is expected to be multicast or broadcast.

3.6.2.3 Commands Generated

The commands generated by the server side of the groups cluster, as listed in Table 3.31, are responses to the received commands listed above in sub-clause 3.6.2.2.2.

Table 3.31 Generated Command IDs for the Groups Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Add group response	M
0x01	View group response	M
0x02	Get group membership response	M
0x03	Remove group response	M
0x04 – 0xff	Reserved	

(Note:-There is no need for a response to the Remove all Groups command, as, at an application level, this command always succeeds)

3.6.2.3.1 Add Group Response Command

The add group response is sent by the groups cluster server in response to an add group command.

3.6.2.3.1.1 Payload Format

The Add Group Response command payload shall be formatted as illustrated in Figure 3.11.

Octets	1	2
Data Type	8-bit Enumeration	Unsigned 16-bit Integer
Field Name	Status	Group ID

Figure 3.11 Format of the Add Group Response Command Payload

3.6.2.3.1.2 When Generated

This command is generated in response to a received Add Group command 10.2.2.3. The Status field is set to SUCCESS, DUPLICATE_EXISTS, or INSUFFICIENT_SPACE as appropriate. The Group ID field is set to the Group ID field of the received Add Group command.

3.6.2.3.2 View Group Response Command

The view group response command is sent by the groups cluster server in response to a view group command.

3.6.2.3.2.1 Payload Format

The View Group Response command payload shall be formatted as illustrated in Figure 3.12.

Octets	1	2	Variable
Data Type	8-bit Enumeration	Unsigned 16-bit Integer	Character string
Field Name	Status	Group ID	Group Name

Figure 3.12 Format of the View Group Response Command Payload

3.6.2.3.2.2 When Generated

This command is generated in response to a received View Group command 10.2.2.4. The Status field is set to SUCCESS or NOT_FOUND as appropriate. The Group ID field is set to the Group ID field of the received View Group command. If the status is SUCCESS, and group names are supported, the Group Name field is set to the Group Name associated with that Group ID in the Group Table; otherwise it is set to the null (empty) string, i.e. a single octet of value 0.

3.6.2.3.3 Get Group Membership Response Command

The get group membership response command is sent by the groups cluster server in response to a get group membership command.

3.6.2.3.3.1 Payload Format

The payload of the get group membership response command is formatted as shown in Figure 3.13.

Octets	1	1	Variable
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer	List of 16-bit group ID
Field Name	Capacity	Group count	Group list

Figure 3.13 Format of the Get Group Membership Response Command Payload

The fields of the get group membership response command have the following semantics:

- The Capacity field shall contain the remaining capacity of the group table of the device. The following values apply:
  0No further groups may be added.
  0 < Capacity < 0xfeCapacity holds the number of groups that may be added
  0xfeAt least 1 further group may be added (exact number is unknown)
  0xffIt is unknown if any further groups may be added
- The Group count field shall contain the number of groups contained in the group list field.
- The Group list field shall contain the identifiers either of all the groups in the group table (in the case where the group list field of the received get group membership command was empty) or all the groups from the group list field of the received get group membership command which are in the group table.

3.6.2.3.3.2 When Generated

When an application entity receives the get group membership command and either the group list of the command payload is empty or the group list contains at least one group to which the entity belongs, the entity shall respond with a get group membership response command.

3.6.2.3.4 Remove Group Response Command

The remove group response command is generated by an application entity in response to the receipt of a remove group command.

3.6.2.3.4.1 Payload Format

The Remove Group Response command payload shall be formatted as illustrated in Figure 3.14.

Octets	1	2
Data Type	8-bit Enumeration	Unsigned 16-bit Integer
Field Name	Status	Group ID

Figure 3.14 Format of the Remove Group Response Command Payload

3.6.2.3.4.2 When Generated

This command is generated in response to a received Remove Group command 3.6.2.2.6. The Status field is set to SUCCESS or NOT_FOUND as appropriate.

The Group ID field is set to the Group ID field of the received Remove Group command.

3.6.3 Client

3.6.3.1 Dependencies

None.

3.6.3.2 Attributes

The Client cluster has no attributes.

3.6.3.3 Commands Received

The client receives the cluster specific response commands detailed in 3.6.2.3.

3.6.3.4 Commands Generated

The client generates the cluster specific commands detailed in 3.6.3.3.

3.7 Scenes Cluster

3.7.1 Overview

Attributes and commands for setting up and recalling scenes. Each scene corresponds to a set of stored values of specified attributes.

In most cases scenes are associated with a particular group ID. Scenes may also exist without a group, in which case the value 0x0000 replaces the group ID. Note that extra care is required in these cases to avoid a scene ID collision, and that commands related to scenes without a group may only be unicast, i.e.: they may not be multicast or broadcast.

3.7.2 Server

3.7.2.1 Dependencies

Any endpoint that implements the Scenes server cluster shall also implement the Groups server cluster.

3.7.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.32.

Table 3.32 Scenes Attribute Sets

Attribute Set Identifier	Description
0x000	Scene Management Information
0x001 – 0xfff	Reserved

3.7.2.2.1 Scene Management Information Attribute Set

The Scene Management Information attribute set contains the attributes summarized in Table 3.33.

Table 3.33 Attributes of the Scene Management Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	SceneCount	Unsigned 8-bit integer	0x00 – 0xff (see 3.7.2.3.1)	Read only	0x00	M
0x0001	CurrentScene	Unsigned 8-bit integer	0x00 – 0xff (see 3.7.2.3.1)	Read only	0x00	M
0x0002	CurrentGroup	Unsigned 16-bit integer	0x0000 – 0xfff7	Read only	0x00	M
0x0003	SceneValid	Boolean	0x00 – 0x01	Read only	0x00	M
0x0004	NameSupport	8-bit bitmap	x0000000	Read only	-	M
0x0005	LastConfiguredBy	IEEE Address	-	Read only	-	O

The most significant bit of the *NameSupport* attribute indicates whether or not group names are supported. A value of 1 indicates that they are supported, and a value of 0 indicates that they are not supported.

3.7.2.2.1.1 *SceneCount* Attribute

The *SceneCount* attribute specifies the number of scenes currently in the device's scene table.

3.7.2.2.1.2 *CurrentScene* Attribute

The *CurrentScene* attribute holds the Scene ID of the scene last invoked.

3.7.2.2.1.3 *CurrentGroup* Attribute

The *CurrentGroup* attribute holds the Group ID of the scene last invoked, or 0x0000 if the scene last invoked is not associated with a group.

3.7.2.2.1.4 *SceneValid* Attribute

The *SceneValid* attribute indicates whether the state of the device corresponds to the values of the *CurrentScene* and *CurrentGroup* attributes. 0x01 indicates that these attributes are valid, 0x00 indicates that they are not valid.

Before a scene has been stored or recalled, this attribute is set to 0x00. After a successful Store Scene or Recall Scene command it is set to 0x01. If, after a scene is stored or recalled, the state of the device is modified, this attribute is set to 0x00.

3.7.2.2.1.5 *NamesSupported* Attribute

The most significant bit of the *NameSupport* attribute indicates whether or not group names are supported. A value of 1 indicates that they are supported, and a value of 0 indicates that they are not supported.

3.7.2.2.1.6 *LastConfiguredBy* Attribute

The *LastConfiguredBy* attribute is 64-bits in length and specifies the IEEE address of the device that last configured the scene table.

The value 0xffffffffffffffff indicates that the device has not been configured, or that the address of the device that last configured the scenes cluster is not known.

3.7.2.3 Scene Table

The scene table is used to store information for each scene capable of being invoked on a device. Each scene is defined for a particular group.

The fields of each scene table entry consist of a number of sets. The base set consists of the first four fields of Table 3.34. A set of extension fields can be added by each additional cluster implemented on a device.

Table 3.34 Fields of a Scene Table Entry

Field	Type	Valid Range	Description
Scene group ID	Unsigned 16-bit integer	0x0000 – 0xfff7	The group ID for which this scene applies, or 0x0000 if the scene is not associated with a group.
Scene ID	Unsigned 8-bit integer	0x00 – 0xff (see 3.7.2.3.1)	The identifier, unique within this group, which is used to identify this scene.
Scene name	Character string	0 – 16 characters	The name of the scene (optional)
Scene transition time	Unsigned 16-bit integer	0x0000 – 0xffff	The amount of time, in seconds, it will take for the device to change from its current state to the requested scene.
Extension field sets	Variable	Variable	See the Scene Table Extensions subsections of individual clusters. Each extension field set holds a set of values of attributes for a cluster implemented on the device. The sum of all such sets defines a scene.

3.7.2.3.1 Scene Names

Scene names are between 0 and 16 characters long. Support of scene names is optional, and is indicated by the *NameSupport* attribute. If scene names are not supported, any commands that writes a scene name shall simply discard the name, and any command that returns a scene names shall return the null string.

3.7.2.3.2 Maximum Number of Scenes

The number of scenes capable of being stored in the table is defined by the profile in which this cluster is used. The default maximum, in the absence of specification by the profile, is 16.

3.7.2.4 Commands Received

The received command IDs for the Scenes cluster are listed in Table 3.35.

Table 3.35 Received Command IDs for the Scenes Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Add scene	M
0x01	View scene	M
0x02	Remove scene	M
0x03	Remove all scenes	M
0x04	Store scene	M
0x05	Recall scene	M
0x06	Get scene membership	M
0x07 – 0xff	Reserved	

3.7.2.4.1 Add Scene Command

The Add Scene command shall be addressed to a single device (not a group).

3.7.2.4.1.1 Payload Format

The payload shall be formatted as illustrated in Figure 3.15.

Octets	2	1	2	Variable	Variable
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 16-bit Integer	Character string	Variable (multiple types)
Field Name	Group ID	Scene ID	Transition time	Scene Name	Extension field sets, one per cluster

Figure 3.15 Format of the Add Scene Command Payload

The format of each extension field set is a 16 bit field carrying the cluster ID, followed by an 8 bit data length field and the set of scene extension fields specified in the relevant cluster.

Extension field sets =
{{ClusterID 1, length 1, {extension field set 1}}, {ClusterID 2, length 2, {extension field set 2 }}, ... }.

3.7.2.4.1.2 Effect on Receipt

On receipt of this command, the device shall (if possible) create an entry in the Scene Table with fields copied from the command payload. If there is already a scene in the table with the same Scene ID and Group ID, it shall overwrite it.

It shall then generate an appropriate Add Scene Response command indicating success or failure. See 3.7.2.5.1.

3.7.2.4.2 View Scene Command

The View Scene command shall be addressed to a single device (not a group).

3.7.2.4.2.1 Payload Format

The payload shall be formatted as illustrated in Figure 3.16.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Group ID	Scene ID

Figure 3.16 Format of the View Scene Command Payload

3.7.2.4.2.2 Effect on Receipt

On receipt of this command, the device shall generate an appropriate View Scene Response command. See 3.7.2.5.2.

3.7.2.4.3 Remove Scene Command

The Remove Scene command may be addressed to a single device or to a group.

3.7.2.4.3.1 Payload Format

The Remove Scene command payload shall be formatted as illustrated in Figure 3.17.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Group ID	Scene ID

Figure 3.17 Format of the Remove Scene Command Payload

3.7.2.4.3.2 Effect on Receipt

On receipt of this command, the device shall (if possible) remove from its Scene Table the entry with this Scene ID and group ID. If the command was addressed to a single device (not a group) then it shall generate an appropriate Remove Scene Response command indicating success or failure. See 3.7.2.5.3.

3.7.2.4.4 Remove All Scenes Command

The Remove All Scenes may be addressed to a single device or to a group.

3.7.2.4.4.1 Payload Format

The Remove All Scenes command payload shall be formatted as illustrated in Figure 3.18.

Octets	2
Data Type	Unsigned 16-bit Integer
Field Name	Group ID

Figure 3.18 Format of the Remove All Scenes Command Payload

3.7.2.4.4.2 Effect on Receipt

On receipt of this command, the device shall, if possible, remove from its Scene Table all entries with this Group ID. If the command was addressed to a single device (not to a group) it shall then generate an appropriate Remove All Scenes Response command indicating success or failure. See 3.7.2.5.4.

3.7.2.4.5 Store Scene Command

The Store Scene command may be addressed to a single device or to a group.

3.7.2.4.5.1 Payload Format

The Store Scene command payload shall be formatted as illustrated in Figure 3.19.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Group ID	Scene ID

Figure 3.19 Format of the Store Scene Command Payload

3.7.2.4.5.2 Effect on Receipt

On receipt of this command, the device shall (if possible) add an entry to the Scene Table with the Scene ID and Group ID given in the command, and all extension fields corresponding to the current state of other clusters on the device. If an entry already exists with the same Scene ID and Group ID it will be replaced.

If the command was addressed to a single device (not to a group) then it shall generate an appropriate Store Scene Response command indicating success or failure. See 3.7.2.5.5.

3.7.2.4.6 Recall Scene Command

The Recall Scene command may be addressed to a single device or to a group.

3.7.2.4.6.1 Payload Format

The Recall Scene command payload shall be formatted as illustrated in Figure 3.20.

Octets	2	1
Data Type	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Group ID	Scene ID

Figure 3.20 Format of the Recall Scene Command Payload

3.7.2.4.6.2 Effect on Receipt

On receipt of this command, the device shall (if possible) locate the entry in its Scene Table with the Group ID and Scene ID given in the command. For each other cluster on the device, it shall then retrieve any corresponding extension fields from the Scene Table and set the attributes and corresponding state of the cluster accordingly.

This command does not result in a response command.

3.7.2.4.7 Get Scene Membership Command

The Get Scene Membership command can be used to find an unused scene number within the group when no commissioning tool is in the network, or for a commissioning tool to get used scenes for a group on a single device or on all devices in the group.

3.7.2.4.7.1 Payload Format

The Get Scene Membership command may be addressed to a single device or to a group.

The Get Scene Membership command payload shall be formatted as illustrated in Figure 3.21.

Ocets	2
Data Type	Unsigned 16-bit Integer
Field Name	Group ID

Figure 3.21 Format of the Get Scene Membership Command Payload

3.7.2.4.7.2 Effect on Receipt

On receipt of this command, the device shall if addressed to a single device generate an appropriate Get Scene Membership Response command, otherwise it shall only generate an appropriate Get Scene Membership Response command if an entry within the Scene Table corresponds to the Group ID. See 3.7.2.5.6.

3.7.2.5 Commands Generated

The generated command IDs for the Scenes cluster are listed in Table 3.36.

Table 3.36 Generated Command IDs for the Scenes Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Add scene response	M
0x01	View scene response	M
0x02	Remove scene response	M
0x03	Remove all scenes response	M
0x04	Store scene response	M

Table 3.36 Generated Command IDs for the Scenes Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x05	Reserved	-
0x06	Get scene membership response	M
0x07 – 0xff	Reserved	-

3.7.2.5.1 Add Scene Response Command

3.7.2.5.1.1 Payload Format

The Add Scene Response command payload shall be formatted as illustrated in Figure 3.22.

Octets	1	2	1
Data Type	8-bit Enumeration	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Status	Group ID	Scene ID

Figure 3.22 Format of the Add Scene Response Command Payload

3.7.2.5.1.2 When Generated

This command is generated in response to a received Add Scene command 11.2.4.1. The Status field is set to SUCCESS, INSUFFICIENT_SPACE or INVALID_FIELD (the group is not present in the Group Table) as appropriate. The Group ID and Scene ID fields are set to the corresponding fields of the received Add Scene command.

3.7.2.5.2 View Scene Response Command

3.7.2.5.2.1 Payload Format

The View Scene Response command payload shall be formatted as illustrated in Figure 3.23.

Octets	1	2	1	0 / 2	0 / Variable	0 / Variable
Data Type	8-bit Enumeration	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 16-bit Integer	Character string	Variable (multiple types)
Field Name	Status	Group ID	Scene ID	Transition time	Scene Name	Extension field sets, one per cluster

Figure 3.23 Format of the View Scene Response Command Payload

The format of each extension field set is a 16 bit field carrying the cluster ID, followed by an 8 bit data length field and the set of scene extension fields specified in the relevant cluster. These fields are concatenated together in the order given in the cluster.

Extension field sets =
{{ClusterID 1, length 1, {extension field set 1}}, {ClusterID 2, length 2, {extension field set 2 }}, ... }.

3.7.2.5.2.2 When Generated

This command is generated in response to a received View Scene command 11.2.4.2.

The entry in the Scene Table with Scene ID and Group ID given in the received View Scene command is located (if possible). The Status field is set to SUCCESS, NOT_FOUND (the scene is not present in the Scene Table) or INVALID_FIELD (the group is not present in the Group Table) as appropriate. The Group ID and Scene ID fields are set to the corresponding fields in the received View Scene command.

If the status is SUCCESS, the Transition time, Scene Name and Extension field fields are copied from the corresponding fields in the table entry, otherwise they are omitted.

3.7.2.5.3 Remove Scene Response Command

3.7.2.5.3.1 Payload Format

The Remove Scene Response command payload shall be formatted as illustrated in Figure 3.24.

Octets	1	2	1
Data Type	8-bit Enumeration	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Status	Group ID	Scene ID

Figure 3.24 Format of the Remove Scene Response Command Payload

3.7.2.5.3.2 When Generated

This command is generated in response to a received Remove Scene command 10.2.2.4. The Status field is set to SUCCESS, NOT_FOUND (the scene is not present in the Scene Table) or INVALID_FIELD (the group is not present in the Group Table) as appropriate. The Group ID and Scene ID fields are set to the corresponding fields of the received Remove Scene command.

3.7.2.5.4 Remove All Scenes Response Command

3.7.2.5.4.1 Payload Format

The Remove All Scenes Response command payload shall be formatted as illustrated in Figure 3.25.

Octets	1	2
Data Type	8-bit Enumeration	Unsigned 16-bit Integer
Field Name	Status	Group ID

Figure 3.25 Format of the Remove All Scenes Response Command Payload

3.7.2.5.4.2 When Generated

This command is generated in response to a received Remove All Scenes command, see 3.7.2.4.4. The Status field is set to SUCCESS or INVALID_FIELD (the group is not present in the Group Table) as appropriate. The Group ID field is set to the corresponding field of the received Remove All Scenes command.

3.7.2.5.5 Store Scene Response Command

3.7.2.5.5.1 Payload Format

The Store Scene Response command payload shall be formatted as illustrated in Figure 3.26.

Octets	1	2	1
Data Type	8-bit Enumeration	Unsigned 16-bit Integer	Unsigned 8-bit Integer
Field Name	Status	Group ID	Scene ID

Figure 3.26 Format of the Store Scene Response Command Payload

3.7.2.5.5.2 When Generated

This command is generated in response to a received Store Scene command 10.2.2.4. The Status field is set to SUCCESS, INSUFFICIENT_SPACE or INVALID_FIELD (the group is not present in the Group Table) as appropriate. The Group ID and Scene ID fields are set to the corresponding fields of the received Store Scene command.

3.7.2.5.6 Get Scene Membership Response Command

3.7.2.5.6.1 Payload Format

The Get Scene Membership Response command payload shall be formatted as illustrated in Figure 3.27.

Octets	1	1	2	0/1	Variable
Data Type	8-bit Enumeration	Unsigned 8-bit Integer	Unsigned 16-bit Integer	Unsigned 8-bit Integer	Unsigned 8-bit Integer x N
Field Name	Status	Capacity	Group ID	Scene count	Scene list

Figure 3.27 Format of the Get Scene Membership Response Command Payload

The fields of the get scene membership response command have the following semantics:

- The Capacity field shall contain the remaining capacity of the scene table of the device. (for all groups). The following values apply:

0No further scenes may be added.
0 < Capacity < 0xfeCapacity holds the number of scenes that may be added
0xfeAt least 1 further scene may be added (exact number is unknown)
0xffIt is unknown if any further scenes may be added

- The Status field shall contain SUCCESS or INVALID_FIELD (the group is not present in the Group Table) as appropriate.
- The Group ID field shall be set to the corresponding field of the received Get Scene Membership command.
- If the status is not SUCCESS, then the Scene count and Scene list field are omitted, else
  - The Scene count field shall contain the number of scenes contained in the Scene list field.
  - The Scene list field shall contain the identifiers of all the scenes in the scene table with the corresponding Group ID.

3.7.2.5.6.2 When Generated

This command is generated in response to a received Get Scene Membership command, 3.7.2.4.7.

3.7.3 Client

3.7.3.1 Dependencies

None.

3.7.3.2 Attributes

The Client cluster has no attributes.

3.7.3.3 Commands Received

The client receives the cluster specific response commands detailed in 3.7.2.5.

3.7.3.4 Commands Generated

The client generates the cluster specific commands detailed in 3.7.2.4, as required by the application.

3.8 On/Off Cluster

3.8.1 Overview

Attributes and commands for switching devices between ‘On’ and ‘Off’ states.

3.8.2 Server

3.8.2.1 Dependencies

None

3.8.2.2 Attributes

The server supports the attributes shown in Table 3.37.

Table 3.37 Attributes of the On/Off Server Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	OnOff	Boolean	0x00 – 0x01	Read only	0x00	M

The *OnOff* attribute has the following values: 0 = Off, 1 = On

3.8.2.3 Commands Received

The command IDs for the *On/Off* cluster are listed in Table 3.38.

Table 3.38 Command IDs for the On/Off Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Off	M
0x01	On	M
0x02	Toggle	M
0x03 – 0xff	Reserved	

3.8.2.3.1 Off Command

This command does not have a payload.

3.8.2.3.1.1 Effect on Receipt

On receipt of this command, a device shall enter its 'Off' state. This state is device dependent, but it is recommended that it is used for power off or similar functions.

3.8.2.3.2 On Command

This command does not have a payload.

3.8.2.3.2.1 Effect on Receipt

On receipt of this command, a device shall enter its 'On' state. This state is device dependent, but it is recommended that it is used for power on or similar functions.

3.8.2.3.3 Toggle Command

This command does not have a payload.

3.8.2.3.3.1 Effect on Receipt

On receipt of this command, if a device is in its ‘Off’ state it shall enter its 'On' state. Otherwise, if it is in its ‘On’ state it shall enter its 'Off' state.

3.8.2.4 Commands Generated

The server generates no commands.

3.8.2.5 Scene Table Extensions

If the Scenes server cluster (11) is implemented, the following extension field is added to the Scenes table:
OnOff

3.8.2.6 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval settings described in the ZCL Foundation specification (see 2.4.7). The following attribute shall be reported:
OnOff

3.8.3 Client

3.8.3.1 Dependencies

None.

3.8.3.2 Attributes

The client has no attributes.

3.8.3.3 Commands Received

No cluster specific commands are received by the client.

3.8.3.4 Commands Generated

The client generates the cluster specific commands received by the server, as required by the application. See 3.8.2.3

3.9 On/Off Switch Configuration Cluster

3.9.1 Overview

Attributes and commands for configuring On/Off switching devices

3.9.2 Server

3.9.2.1 Dependencies

Any endpoint that implements this server cluster shall also implement the On/Off client cluster.

3.9.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.39.

Table 3.39 On/Off Switch Configuration Attribute Sets

Attribute Set Identifier	Description
0x000	Switch Information
0x001	Switch Settings
0x002 – 0xfff	Reserved

3.9.2.2.1 Switch Information Attribute Set

The switch information attribute set contains the attributes summarized in Table 3.40.

Table 3.40 Attributes of the Switch Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	SwitchType	8-bit Enumeration	0x00 – 0x01	Read only	-	M

3.9.2.2.2 *SwitchType* Attribute

The *SwitchType* attribute specifies the basic functionality of the On/Off switching device. This attribute shall be set to one of the non-reserved values listed in Table 3.41.

Table 3.41 Values of the *SwitchType* Attribute

SwitchType Attribute Value	Description	Details
0x00	Toggle	A switch with two physical states. An action by the user (e.g. toggling a rocker switch) moves the switch from state 1 to state 2. The switch then remains in that state until another action from the user returns it to state 1.
0x01	Momentary	A switch with two physical states. An action by the user (e.g. pressing a button) moves the switch from state 1 to state 2. When the user ends his action (e.g. releases the button) the switch returns to state 1.
0x02 – 0xff	Reserved	-

3.9.2.2.3 Switch Settings Attribute Set

The switch settings attribute set contains the attributes summarized in Table 3.42.

Table 3.42 Attributes of the Switch Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	SwitchActions	8-bit Enumeration	0x00 – 0x02	Read/ write	0x00	M

3.9.2.2.3.1 *SwitchActions* Attribute

The *SwitchActions* attribute is 8-bits in length and specifies the commands of the On/Off cluster (12) to be generated when the switch moves between its two states, as detailed in Table 3.43.

Table 3.43 Values of the *SwitchActions* Attribute

SwitchActions Attribute Value	Command Generated When Arriving at State 2 From State 1	Command Generated When Arriving at State 1 From State 2
0x00	On	Off
0x01	Off	On
0x02	Toggle	Toggle
0x03 – 0xff	Reserved	

3.9.2.3 Commands Received

No commands are received by the server.

3.9.2.4 Commands Generated

The server generates no commands.

3.9.3 Client

3.9.3.1 Dependencies

None

3.9.3.2 Attributes

The client has no attributes.

3.9.3.3 Commands Received

No cluster specific commands are received by the client.

3.9.3.4 Commands Generated

No cluster specific commands are generated by the client.

3.10 Level Control Cluster

3.10.1 Overview

This cluster provides an interface for controlling a characteristic of a device that can be set to a level, for example the brightness of a light, the degree of closure of a door, or the power output of a heater.

3.10.2 Server

3.10.2.1 Dependencies

For many applications, a close relationship between this cluster and the OnOff cluster is needed. This section describes the dependencies that are required when an endpoint that implements the Level Control server cluster also implements the On/Off server cluster (12.2).

The *OnOff* attribute of the On/Off cluster and the *CurrentLevel* attribute of the Level Control cluster are intrinsically independent variables, as they are on different clusters. However, when both clusters are implemented on the same endpoint, dependencies may be introduced between them. Facilities are provided to introduce dependencies if required.

3.10.2.1.1 Effect of On/Off Commands on the *CurrentLevel* Attribute

The attribute *OnLevel* (see 3.10.2.2.4) determines whether commands of the On/Off cluster have a permanent effect on the *CurrentLevel* attribute or not. If this attribute is defined (i.e. implemented and not 0xff) they do have a permanent effect, otherwise they do not. There is always a temporary effect, due to fading up / down.

The effect on the Level Control cluster on receipt of the various commands of the On/Off cluster are as detailed in Table 3.44. In this table, and throughout this cluster specification, 'level' means the value of the *CurrentLevel* attribute (see 3.10.2.2.1).

Table 3.44 Actions on Receipt for On/Off Commands, When Associated With Level Control

Command	Action On Receipt
On	Temporarily store *CurrentLevel* Set *CurrentLevel* to the minimum level allowed for the device. Move *CurrentLevel* to *OnLevel,* or to the stored level if *OnLevel* is not defined, over the time period *OnOffTransitionTime*.
Off	Temporarily store *CurrentLevel* Move *CurrentLevel* to the minimum level allowed for the device over the time period *OnOffTransitionTime*. If *OnLevel* is not defined, set the *CurrentLevel* to the stored level.
Toggle	If the *OnOff* attribute has the value Off, proceed as for the On command. Otherwise proceed as for the Off command.

3.10.2.1.2 Effect of Level Control Commands on the *OnOff* Attribute

There are two sets of commands provided in the Level Control cluster. These are identical, except that the first set (Move to Level, Move and Step) shall not effect the *OnOff* attribute, whereas the second set ('with On/Off' variants) shall.

The first set is used to maintain independence between the *CurrentLevel* and *OnOff* attributes, so changing *CurrentLevel* has no effect on the *OnOff* attribute. As examples, this represents the behavior of a volume control with a mute button, or a 'turn to set level and press to turn on/off' light dimmer.

The second set is used to link the *CurrentLevel* and *OnOff* attributes. When the level is reduced to its minimum the *OnOff* attribute is automatically turned to Off, and when the level is increased above its minimum the *OnOff* attribute is automatically turned to On. As an example, this represents the behavior of a light dimmer with no independent on/off switch.

3.10.2.2 Attributes

The attributes of the Level Control server cluster are summarized in Table 3.45.

Table 3.45 Attributes of the Level Control Server Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	CurrentLevel	Unsigned 8-bit integer	0x00 – 0xff	Read only	0x00	M
0x0001	RemainingTime	Unsigned 16-bit integer	0x0000 – 0xffff	Read only	0x0000	O
0x0010	OnOffTransitionTime	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	0x0000	O
0x0011	OnLevel	Unsigned 8-bit Integer	0x00 – 0xfe	Read / Write	0xfe	O

3.10.2.2.1 *CurrentLevel* Attribute

The *CurrentLevel* attribute represents the current level of this device. The meaning of 'level' is device dependent.

3.10.2.2.2 *RemainingTime* Attribute

The *RemainingTime* attribute represents the time remaining until the current command is complete - it is specified in 1/10ths of a second.

3.10.2.2.3 *OnOffTransitionTime* Attribute

The *OnOffTransitionTime* attribute represents the time taken to move to or from the target level when On of Off commands are received by an On/Off cluster on the same endpoint. It is specified in 1/10ths of a second.

The actual time taken should be as close to *OnOffTransitionTime* as the device is able. N.B. If the device is not able to move at a variable rate, the *OnOffTransitionTime* attribute should not be implemented.

3.10.2.2.4 *OnLevel* Attribute

The *OnLevel* attribute determines the value that the *CurrentLevel* attribute is set to when the *OnOff* attribute of an On/Off cluster on the same endpoint is set to On. If the *OnLevel* attribute is not implemented, or is set to 0xff, it has no effect. For more details see 3.10.2.1.1.

3.10.2.3 Commands Received

The command IDs for the Level Control cluster are listed in Table 3.46.

Table 3.46 Command IDs for the Level Control Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Move to Level	M
0x01	Move	M
0x02	Step	M
0x03	Stop	M
0x04	Move to Level (with On/Off)	M
0x05	Move (with On/Off)	M
0x06	Step (with On/Off)	M
0x07	Stop	M
0x08 – 0xff	Reserved	-

3.10.2.3.1 Move to Level Command

3.10.2.3.1.1 Payload Format

The Move to Level command payload shall be formatted as illustrated in Figure 3.28.

Octets	1	2
Data Type	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	Level	Transition time

Figure 3.28 Format of the Move to Level Command Payload

3.10.2.3.1.2 Effect on Receipt

On receipt of this command, a device shall move from its current level to the value given in the Level field. The meaning of ‘level’ is device dependent – e.g. for a light it may mean brightness level.

The movement shall be as continuous as technically practical, i.e. not a step function, and the time taken to move to the new level shall be equal to the value of the Transition time field, in tenths of a second, or as close to this as the device is able.

If the Transition time field takes the value 0xffff then the time taken to move to the new level shall instead be determined by the *OnOffTransitionTime* attribute. If *OnOffTransitionTime* , which is an optional attribute, is not present, the device shall move to its new level as fast as it is able.

If the device is not able to move at a variable rate, the Transition time field may be disregarded.

3.10.2.3.2 Move Command

3.10.2.3.2.1 Payload Format

The Move command payload shall be formatted as illustrated in Figure 3.29.

Octets	1	2
Data Type	8-bit Enumeration	Unsigned 8-bit Integer
Field Name	Move mode	Rate

Figure 3.29 Format of the Move Command Payload

3.10.2.3.2.2 Move Mode Field

The Move mode field shall be one of the non-reserved values in Table 3.47.

Table 3.47 Values of the Move Mode Field

Fade Mode Value	Description
0x00	Up
0x01	Down
0x02 – 0xff	Reserved

3.10.2.3.2.3 Rate Field

The Rate field specifies the rate of movement in units per second. The actual rate of movement should be as close to this rate as the device is able. If the Rate field is 0xff the device should move as fast as it is able.

If the device is not able to move at a variable rate, this field may be disregarded.

3.10.2.3.2.4 Effect on Receipt

On receipt of this command, a device shall move from its current level in an up or down direction in a continuous fashion, as detailed in Table 3.48.

Table 3.48 Actions on Receipt for Move Command

Fade Mode	Action on Receipt
Up	Increase the device's level at the rate given in the Rate field. If the level reaches the maximum allowed for the device, stop.
Down	Decrease the device's level at the rate given in the Rate field. If the level reaches the minimum allowed for the device, stop.

3.10.2.3.3 Step Command

3.10.2.3.3.1 Payload Format

The Step command payload shall be formatted as illustrated in Figure 3.30.

Octets	1	1	2
Data Type	8-bit Enumeration	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	Step mode	Step size	Transition time

Figure 3.30 Format of the Step Command Payload

The Step mode field shall be one of the non-reserved values in Table 3.49.

Table 3.49 Values of the Step Mode Field

Fade Mode Value	Description
0x00	Up
0x01	Down
0x02 – 0xff	Reserved

The Transition time field specifies the time that shall be taken to perform the step, in tenths of a second. A step is a change in the *CurrentLevel* of 'Step size' units. The actual time taken should be as close to this as the device is able. If the Transition time field is 0xffff the device should move as fast as it is able.

If the device is not able to move at a variable rate, the Transition time field may be disregarded.

3.10.2.3.3.2 Effect on Receipt

On receipt of this command, a device shall move from its current level in an up or down direction as detailed in Table 3.50.

Table 3.50 Actions on Receipt for Step Command

Fade Mode	Action on Receipt
Up	Increase *CurrentLevel* by 'Step size' units, or until it reaches the maximum level allowed for the device if this reached in the process. In the latter case, the transition time shall be proportionally reduced.
Down	Decrease *CurrentLevel* by 'Step size' units, or until it reaches the minimum level allowed for the device if this reached in the process. In the latter case, the transition time shall be proportionally reduced.

3.10.2.3.4 Stop Command

This command has no payload. Upon receipt of this command, any Move to Level, Move or Step command (and their 'with On/Off' variants) currently in process shall be terminated. The value of *CurrentLevel* shall be left at its value upon receipt of the Stop command, and *RemainingTime* shall be set to zero.

This command has two entries in Table 3.1, one for the Move to Level, Move and Set commands, and one for their 'with On/Off' counterparts. This is solely for symmetry, to allow easy choice of one or other set of commands – the Stop commands are identical.

3.10.2.3.5 'With On/Off' Commands

The Move to Level (with On/Off), Move (with On/Off) and Step (with On/Off) commands have identical payloads to the Move to Level, Move and Step commands respectively. The also have the same effects, except for the following additions.

- Before commencing any command that has the effect of increasing *CurrentLevel*, the *OnOff* attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to On.
- If any command that decreases *CurrentLevel* reduces it to the minimum level allowed by the device, the *OnOff* attribute of the On/Off cluster on the same endpoint, if implemented, shall be set to Off.

3.10.2.4 Commands Generated

The server generates no commands.

3.10.2.5 Scene Table Extensions

If the Scenes server cluster (11) is implemented, the following extension field is added to the Scenes table:

CurrentLevel

3.10.2.6 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation Specification (see 2.4.7). The following attribute shall be reported:

CurrentLevel

3.10.3 Client

3.10.3.1 Dependencies

None.

3.10.3.2 Attributes

The client has no attributes.

3.10.3.3 Commands Received

No cluster specific commands are received by the client.

3.10.3.4 Commands Generated

The client generates the cluster specific commands received by the server (see 3.10.2.3), as required by the application.

3.11 Alarms Cluster

3.11.1 Overview

Attributes and commands for sending alarm notifications and configuring alarm functionality.

Alarm conditions and their respective alarm codes are described in individual clusters, along with an alarm mask field. Where not masked, alarm notifications are reported to subscribed targets using binding.

Where an alarm table is implemented, all alarms, masked or otherwise, are recorded and may be retrieved on demand.

Alarms may either reset automatically when the conditions that cause are no longer active, or may need to be explicitly reset.

3.11.2 Server

3.11.2.1 Dependencies

Any endpoint which implements time stamping shall also implement the Time server cluster.

3.11.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.51.

Table 3.51 Alarms Cluster Attribute Sets

Attribute Set Identifier	Description
0x000	Alarm Information
0x001 – 0xfff	Reserved

3.11.2.2.1 Alarm Information Attribute Set

The Alarm Information attribute set contains the attributes summarized in Table 3.52.

Table 3.52 Attributes of the Alarm Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	AlarmCount	Unsigned 16-bit integer	0x00 – maximum defined in profile	Read only	0x00	O

3.11.2.2.1.1 *AlarmCount* Attribute

The *AlarmCount* attribute is 16-bits in length and specifies the number of entries currently in the alarm table. This attribute shall be specified in the range 0x00 to the maximum defined in the profile using this cluster.

If alarm logging is not implemented this attribute shall always take the value 0x00.

3.11.2.3 Alarm Table

The alarm table is used to store details of alarms generated within the devices. Alarms are requested by clusters which have alarm functionality, e.g. when attributes take on values that are outside 'safe' ranges.

The maximum number of entries in the table is device dependent.

When an alarm is generated, a corresponding entry is placed in the table. If the table is full, the entry with the earliest time stamp field is replaced by the new entry.

3.11.2.3.1 Alarm Table Format

The format of an alarm table entry is illustrated in Table 3.53.

Table 3.53 Format of the Alarm Table

Field	Type	Valid range	Description
Alarm code	8-bit Enumeration	0x00 – 0xff	Identifying code for the cause of the alarm, as given in the specification of the cluster whose attribute generated this alarm.
Cluster identifier	Cluster ID	0x0000 – 0xffff	The identifier of the cluster whose attribute generated this alarm.
Time stamp	Unsigned 32-bit integer	0x00000000 – 0xffffffff	The time at which the alarm occurred, or 0xffffffff if no time information is available. This time is taken from a Time server cluster, which must be present on the same endpoint.

3.11.2.4 Commands Received

The received command IDs for the Alarms cluster are listed in Table 3.54.

Table 3.54 Received Command IDs for the Alarms Cluster

Command identifier field value	Description	Mandatory / Optional
0x00	Reset Alarm	M
0x01	Reset all alarms	M
0x02	Get Alarm	O
0x03	Reset alarm log	O
0x04 – 0xff	Reserved	

3.11.2.4.1 Reset Alarm Command

This command resets a specific alarm. This is needed for some alarms that do not reset automatically. If the alarm condition being reset was in fact still active then a new notification will be generated and, where implemented, a new record added to the alarm log.

3.11.2.4.1.1 Payload Format

The Reset Alarm command payload shall be formatted as illustrated in Figure 3.31.

Octets	1	2
Data Type	8-bit Enumeration	Cluster ID
Field Name	Alarm code	Cluster identifier

Figure 3.31 Format of the Reset Alarm Command Payload

3.11.2.4.2 Reset All Alarms Command

This command resets all alarms. Any alarm conditions that were in fact still active will cause a new notification to be generated and, where implemented, a new record added to the alarm log.

3.11.2.4.3 Get Alarm Command

This command causes the alarm with the earliest timestamp in the alarm table to be reported in a get alarm response command 3.11.2.5.2. This command enables the reading of logged alarm conditions from the alarm table. Once an alarm condition has been reported the corresponding entry in the table is removed.

This command does not have a payload.

3.11.2.4.4 Reset Alarm Log Command

This command causes the alarm table to be cleared, and does not have a payload.

3.11.2.5 Commands Generated

The generated command IDs for the Alarms cluster are listed in Table 3.55.

Table 3.55 Generated Command IDs for the Alarms Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Alarm	M
0x01	Get alarm response	O
0x02 – 0xff	Reserved	-

3.11.2.5.1 Alarm Command

The alarm command signals an alarm situation on the sending device.

An alarm command is generated when a cluster which has alarm functionality detects an alarm condition, e.g. an attribute has taken on a value that is outside a ‘safe’ range. The details are given by individual cluster specifications.

3.11.2.5.1.1 Payload Format

The alarm command payload shall be formatted as illustrated in Figure 3.32.

Octets	1	2
Data Type	8-bit Enumeration	Cluster ID
Field Name	Alarm code	Cluster identifier

Figure 3.32 Format of the Alarm Command Payload

3.11.2.5.2 Get Alarm Response Command

The get alarm response command returns the results of a request to retrieve information from the alarm log, along with a time stamp indicating when the alarm situation was detected.

3.11.2.5.2.1 Payload Format

The get alarm response command payload shall be formatted as illustrated in Figure 3.33.

Octets	1	0/1	0/2	0/4
Data Type	8-bit Enumeration	8-bit Enumeration	Cluster ID	Unsigned 32-bit integer
Field Name	Status	Alarm code	Cluster identifier	Time stamp

Figure 3.33 Format of the Get Alarm Response Command Payload

If there is at least one alarm record in the alarm table then the status field is set to SUCCESS. The alarm code, cluster identifier and time stamp fields shall all be present and shall take their values from the item in the alarm table that they are reporting.

If there are no more alarms logged in the alarm table then the status field is set to NOT_FOUND and the alarm code, cluster identifier and time stamp fields shall be omitted.

3.11.3 Client

3.11.3.1 Dependencies

None

3.11.3.2 Attributes

The client has no attributes.

3.11.3.3 Commands Received

The client receives the cluster specific commands generated by the server (see 3.11.2.5).

3.11.3.4 Commands Generated

The client generates the cluster specific commands received by the server (see 3.11.2.4), as required by the application.

3.12 Time Cluster

3.12.1 Overview

This cluster provides a basic interface to a real-time clock. The clock time may be read and also written, in order to synchronize the clock (as close as practical) to a time standard. This time standard is the number of seconds since 0 hrs 0 mins 0 sec on 1st January 2000 UTC.

The functionality does not include clock functions such as time zone or daylight saving time.

3.12.2 Server

3.12.2.1 Dependencies

None

3.12.2.2 Attributes

The server supports the attributes shown in Table 3.56.

Table 3.56 Attributes of the On/Off Server Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	Time	Unsigned 32-bit integer	0x00000000 – 0xffffffff	Read / Write	-	M
0x0001	TimeStatus	8-bit bitmap	0000 00xx	Read / Write	00000000	M

3.12.2.2.1 *Time* Attribute

The *Time* attribute is 32-bits in length and holds the time value of a real time clock. If the Master bit of the *TimeStatus* attribute has a value of 0, writing to this attribute shall set the real time clock to the written value, otherwise it cannot be written. The value 0xffffffff indicates an invalid time.

3.12.2.2.2 *TimeStatus* Attribute

The *TimeStatus* attribute holds a number of bit fields, as detailed in Table 3.57.

Table 3.57 Bit Values of the *TimeStatus* Attribute

TimeStatus Attribute Bit Number	Meaning	Values
0	Master	1 – master clock 0 – not master clock
1	Synchronized	1 – synchronized 0 – not synchronized
2-7	Reserved	-

The Master and Synchronized bits together provide information on how closely the *Time* attribute conforms to the time standard.

The Master bit specifies whether the real time clock corresponding to the *Time* attribute is internally set to the time standard (see 3.12.1). This bit is not writeable – if a value is written to the *TimeStatus* attribute, this bit does not change.

The Synchronized bit specifies whether *Time* has been set over the network to synchronize it (as close as may be practical) to the time standard (see 3.12.1). This bit must be explicitly written to indicate this – i.e. it is not set automatically on writing to the *Time* attribute. If the Master bit is 1, the value of this bit is 0.

If both the Master and Synchronized bits are 0, the real time clock has no defined relationship to the time standard (e.g. it may record the number of seconds since the device was initialized).

3.12.2.3 Commands Received

The server receives no commands except those to read and write attributes.

3.12.2.4 Commands Generated

The server generates no commands.

3.12.3 Client

3.12.3.1 Dependencies

None.

3.12.3.2 Attributes

The client has no attributes.

3.12.3.3 Commands Received

No cluster specific commands are received by the client.

3.12.3.4 Commands Generated

The client generates no cluster specific commands.

3.13 Rssi Location Cluster

3.13.1 Overview

This cluster provides a means for exchanging Received Signal Strength Indication (RSSI) based location information and channel parameters among devices.

3.13.2 Server

3.13.2.1 Dependencies

None

3.13.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 3.58.

Table 3.58 Location Attribute Sets

Attribute Set Identifier	Description
0x000	Location Information
0x001	Location Settings
0x002 – 0xfff	Reserved

3.13.2.2.1 Location Information Attribute Set

The Location Information attribute set contains the attributes summarized in Table 3.59.

Table 3.59 Attributes of the Location Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	LocationType	8-bit Data	0000xxxx	Read only	-	M
0x0001	LocationMethod	8-bit Enumeration	0x00 – 0xff	Read only	-	M
0x0002	LocationAge	Unsigned 16-bit Integer	0x0000 – 0xffff	Read only	-	O
0x0003	QualityMeasure	Unsigned 8-bit Integer	0x00 – 0x64	Read only	-	O
0x0004	NumberOfDevices	Unsigned 8-bit Integer	0x00 – 0xff	Read only	-	O

3.13.2.2.1.1 *LocationType* Attribute

The *LocationType* attribute is 8 bits long and is divided into bit fields. The meanings of the individual bit fields are detailed in Table 3.60.

Table 3.60 Bit Values of the *LocationType* Attribute

Bit Field (Bit Numbers)	Meaning	Values
0	Absolute	1 – Absolute location 0 – Measured location
1	2-D	1 – Two dimensional 0 – Three dimensional
2-3	Coordinate System	0 – Rectangular (installation-specific origin and orientation) 1-3 – Reserved
4-7	Reserved	-

The Absolute bit field indicates whether the location is a known absolute location or is calculated.

The 2-D bit field indicates whether the location information is two- or three-dimensional. If the location information is two-dimensional, Coordinate 3 is unknown and shall be set to 0x8000.

The Coordinate System bit field indicates the geometry of the system used to express the location coordinates. If the field is set to zero, the location coordinates are expressed using the rectangular coordinate system. All other values are reserved.

3.13.2.2.1.2 *LocationMethod* Attribute

The Location Method attribute shall be set to one of the non-reserved values in Table 3.61.

Table 3.61 Values of the *LocationMethod* Attribute

Value	Method	Description
0x00	Lateration	A method based on RSSI measurements from three or more sources.
0x01	Signposting	The location reported is the location of the neighboring device with the strongest received signal.
0x02	RF fingerprinting	RSSI signatures are collected into a database at commissioning time. The location reported is the location taken from the RSSI signature database that most closely matches the device's own RSSI signature.
0x03	Out of band	The location is obtained by accessing an out-of-band device (that is, the device providing the location is not part of the ZigBee network).
0x04 – 0x3f	-	Reserved
0x40 – 0xff	-	Reserved for manufacturer specific location methods.

3.13.2.2.1.3 *LocationAge* Attribute

The *LocationAge* attribute indicates the amount of time, measured in seconds, that has transpired since the location information was last calculated. This attribute is not valid if the Absolute bit of the *LocationType* attribute is set to one.

3.13.2.2.1.4 *QualityMeasure* Attribute

The *QualityMeasure* attribute is a measure of confidence in the corresponding location information. The higher the value, the more confident the transmitting device is in the location information. A value of 0x64 indicates complete (100%) confidence and a value of 0x00 indicates zero confidence. (Note: no fixed confidence metric is mandated – the metric may be application and manufacturer dependent.)

This field is not valid if the Absolute bit of the *LocationType* attribute is set to one.

3.13.2.2.1.5 *NumberOfDevices* Attribute

The *NumberOfDevices* attribute is the number of devices whose location data were used to calculate the last location value. This attribute is related to the *QualityMeasure* attribute.

3.13.2.2.2 Location Settings Attribute Set

The Location Settings attribute set contains the attributes summarized in Table 3.62.

Table 3.62 Attributes of the Location Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	Coordinate1	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	M
0x0011	Coordinate2	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	M
0x0012	Coordinate3	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	O
0x0013	Power	Signed 16-bit integer	0x8000 – 0x7fff	Read / Write	-	M
0x0014	PathLossExponent	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	-	M
0x0015	ReportingPeriod	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	-	O
0x0016	CalculationPeriod	Unsigned 16-bit integer	0x0000 – 0xffff	Read / Write	-	O
0x0017	NumberRSSIMeasurements	Unsigned 8-bit integer	0x01 – 0xff	Read / Write	-	M

3.13.2.2.2.1 *Coordinate* 1,2,3 Attributes

The *Coordinate1, Coordinate2 and Coordinate3* attributes are signed 16-bit integers, and represent orthogonal linear coordinates x, y, z in meters as follows.

$x = Coordinate1 / 10, \quad y = Coordinate2 / 10, \quad z = Coordinate3 / 10$

The range of x is -3276.7 to 3276.7 meters, corresponding to *Coordinate1* between 0x8001 and 0x7fff. The same range applies to y and z. A value of 0x8000 for any of the coordinates indicates that the coordinate is unknown.

3.13.2.2.2.2 *Power* Attribute

The *Power* attribute specifies the value of the average power P_0, measured in dBm, received at a reference distance of one meter from the transmitter.

$P_0 = Power / 100$

A value of 0x8000 indicates that *Power* is unknown.

3.13.2.2.2.3 *PathLossExponent* Attribute

The *PathLossExponent* attribute specifies the value of the Path Loss Exponent n, an exponent that describes the rate at which the signal power decays with increasing distance from the transmitter.

$n = PathLossExponent / 100$

A value of 0xffff indicates that *PathLossExponent* is unknown.

The signal strength in dBm at a distance d meters from the transmitter is given by

$P = P_0 – 10n \times \log_{10}(d)$

where

P is the power in dBm at the receiving device.

P_0 is the average power in dBm received at a reference distance of 1meter from the transmitter.

n is the path loss exponent.

d is the distance in meters between the transmitting device and the receiving device.

3.13.2.2.2.4 *ReportingPeriod* Attribute

The *ReportingPeriod* attribute specifies the time in seconds between successive reports of the device's location by means of the Location Data Notification command. The minimum value this attribute can take is specified by the profile in use. If *ReportingPeriod* is zero, the device does not automatically report its location. Note that location information can always be polled at any time.

3.13.2.2.2.5 *CalculationPeriod* Attribute

The *CalculationPeriod* attribute specifies the time in seconds between successive calculations of the device's location. If *CalculationPeriod* is less than the physically possible minimum period that the calculation can be performed, the calculation will be repeated as frequently as possible.

3.13.2.2.2.6 *NumberRSSIMeasurements* Attribute

The *NumberRSSIMeasurements* attribute specifies the number of RSSI measurements to be used to generate one location estimate. The measurements are averaged to improve accuracy. *NumberRSSIMeasurements* must be greater than or equal to 1.

3.13.2.3 Commands Received

The received command IDs for the Location cluster are listed in Table 3.63

Table 3.63 Received Command IDs for the Location Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Set absolute location	M
0x01	Set device configuration	M
0x02	Get device configuration	M
0x03	Get location data	M
0x04– 0xff	Reserved	-

3.13.2.3.1 Set Absolute Location Command

This command is used to set a device’s absolute (known, not calculated) location and the channel parameters corresponding to that location.

3.13.2.3.1.1 Payload Format

The Set Absolute Location command payload shall be formatted as illustrated in Figure 3.34.

Octets	2	2	2	2	2
Data Type	Signed Integer	Signed Integer	Signed Integer	Signed Integer	Unsigned Integer
Field Name	Coordinate 1	Coordinate 2	Coordinate 3	Power	Path Loss Exponent

Figure 3.34 Format of the Set Absolute Location Command Payload

The fields of the payload correspond directly to the attributes with the same names. For details of their meaning and ranges see the descriptions of the individual attributes.

The three coordinate fields shall contain the absolute location (known, not calculated) of the destination device. If any coordinate field(s) is not known, the value(s) shall be set to 0x8000.

3.13.2.3.1.2 Effect on Receipt

On receipt of this command, the device shall update the attributes corresponding to (i.e. with the same names as) the payload fields.

3.13.2.3.2 Set Device Configuration Command

This command is used to set a device's location parameters, which will be used for calculating and reporting measured location. This command is invalid unless the Absolute bit of the *LocationType* attribute has a value of 0.

3.13.2.3.2.1 Payload Format

The Set Device Configuration command payload shall be formatted as illustrated in Figure 3.35.

Octets	2	2	2	1	2
Data Type	Signed Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer
Field Name	Power	Path Loss Exponent	Calculation Period	Number RSSI Measurements	Reporting Period

Figure 3.35 Format of the Set Device Configuration Payload

The fields of the payload correspond directly to the attributes with the same names. For details of their meaning and ranges see the descriptions of the individual attributes.

3.13.2.3.2.2 Effect on Receipt

On receipt of this command, the device shall update the attributes corresponding to (i,.e. with the same names as) the payload fields.

3.13.2.3.3 Get Device Configuration Command

This command is used to request the location parameters of a device. The location parameters are used for calculating and reporting measured location.

3.13.2.3.3.1 Payload Format

The Get Device Configuration command payload shall be formatted as illustrated in Figure 3.36.

Octets	1
Data Type	IEEE Address
Field Name	Target Address

Figure 3.36 Format of the Get Device Configuration Payload

The Target Address field contains the 64-bit IEEE address of the device for which the location parameters are being requested. This field may contain the address of the sending device, the address of the receiving device or the address of a third device.

Note:- one reason a device may request its own configuration is that there may be a designated device which holds the configurations of other devices for distribution at commissioning time. It is also possible that the device may lose its configuration settings for some other reason (loss of power, reset). In the case of a third device, that device may sleep a lot and not be easily accessible.

3.13.2.3.3.2 Effect on Receipt

On receipt of this command, the device shall generate a Device Configuration Response command (3.13.2.4.1).

3.13.2.3.4 Get Location Data Command

This command is used to request a device’s location information and channel parameters. It may be sent as a unicast, multicast or broadcast frame. When sent as a broadcast frame, care should be taken to minimize the risk of a broadcast 'storm' - in particular, it is recommended that the broadcast radius is set to 1.

(Note: devices may or may not acquire and store information on other devices' locations such that this information may be requested by another device. This is application dependent.)

3.13.2.3.4.1 Payload Format

The Get Location Data command payload shall be formatted as illustrated in Figure 3.37.

Bits	3	1	1	1	1	1	8	0 / 64
Data Type	8-bit Bitmap						Unsigned Integer	IEEE address
Field Name	Reserved	Compact Response	Broadcast Response	Broadcast Indicator	Re-calculate	Absolute Only	Number Responses	Target Address

Figure 3.37 Format of the Get Location Data Payload

The highest 3 bits of the first octet are reserved and shall be set to zero.

The Absolute Only field (bit 0 of the first octet) specifies the type of location information being requested. If the Absolute Only field is set to one, the device is only requesting absolute location information (a device may want to gather absolute node locations for use in its own location calculations, and may not be interested in neighbors with calculated values). Otherwise, if the field is set to zero, the device is requesting all location information (absolute and calculated).

The Recalculate field (bit 1 of the first octet) indicates whether the device is requesting that a new location calculation be performed. If the field is set to zero, the device is requesting the currently stored location information. Otherwise, if the field is set to one, the device is requesting that a new calculation be performed. This field is only valid if the Absolute Only field is set to zero.

The Broadcast Indicator field (bit 2 of the first octet) indicates whether the command is being sent as a unicast, multicast or broadcast frame. If the field is set to one, the command is sent as a broadcast or multicast, else it is sent as a unicast.

The Broadcast Response field (bit 3 of the first octet)) indicates whether subsequent responses after the first (where the Number Responses field is greater than one) shall be unicast or broadcast. Broadcast responses can be used as a 'location beacon'.

The Compact Response field (bit 3 of the first octet)) indicates whether subsequent responses after the first (where the Number Responses field is greater than one) shall be sent using the Location Data Notification or the Compact Location Data Notification command.

The Number Responses field indicates the number of location responses to be returned. The information to be returned is evaluated this number of times, with a period equal to the value of the *ReportingPeriod* attribute, and a separate response is sent for each evaluation. This field shall have a minimum value of one. Values greater than one are typically used for situations where locations are changing.

The Target Address field contains the 64-bit IEEE address of the device for which the location information and channel parameters are being requested. If the Broadcast Indicator field is set to zero (i.e. the command is sent as a unicast) this field may contain the address of the receiving device, the address of the sending device or the address of any other device. If the Broadcast Indicator field is set to one (i.e. the command is sent as a broadcast or multicast) the target address is implicitly that of the receiving device, so this field shall be omitted.

3.13.2.3.4.2 Effect on Receipt

On receipt of this command, if the Location Type field is set to zero, only a receiving device(s) that knows its absolute location shall respond by generating a Location Data Response command. If the Location Type field is set to one, all devices receiving this command shall respond by generating a Location Data Response command.

If the command is sent as a unicast, information for the device specified in the Target Address field shall be returned, if the receiving device has or can obtain the information for that device.. If the information is not available, the Status field of the Location Data Response command shall be set to NOT_FOUND.

If the command is sent as a broadcast or multicast, receiving devices shall send back their own information (there is no IEEE target address in this case).

If the Number Responses field is greater than one, the subsequent location readings/calculations shall be sent using the Location Data Notification or the Compact Location Data Notification command, depending on the value of the Reduced Response field .

3.13.2.4 Commands Generated

Table 3.64 Generated Command IDs for the Location Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Device configuration response	M
0x01	Location data response	M
0x02	Location data notification	M
0x03	Compact location data notification	M
0x04	RSSI Ping	M
0x05– 0xff	Reserved	M

3.13.2.4.1 Device Configuration Response Command

This command is sent by a device in response to a Get Device Configuration command (3.13.2.3.3).

3.13.2.4.1.1 Payload Format

The Device Configuration Response command payload shall be formatted as illustrated in Figure 3.38. All payload fields are relevant to the device for which the location parameters have been requested.

Octets	1	0 / 2	0 / 2	0 / 2	0 / 1	0 / 2
Data Type	Enumeration	Signed Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer	Unsigned Integer
Field Name	Status	Power	Path Loss Exponent	Calculation Period	Number RSSI Measurements	Reporting Period

Figure 3.38 Format of the Device Configuration Response Payload

The fields of the payload (other than Status) correspond directly to the attributes with the same names. For details of their meaning and ranges see the descriptions of the individual attributes.

The Status field indicates whether the response to the request was successful or not. If the field is set to SUCCESS, the response was successful. If the field is set to NOT_FOUND, the receiving device was unable to provide the location parameters of the device for which the location parameters were requested. If the field is set to NOT_FOUND, all other payload fields shall not be sent. See 2.5.3 for status codes.

3.13.2.4.2 Location Data Response Command

This command is sent by a device in response to a request for location information and channel parameters.

3.13.2.4.2.1 Payload Format

The Location Data Response command payload shall be formatted as illustrated in Figure 3.39. All payload fields are relevant to the device for which the location parameters have been requested.

Octets	1	0 / 1	0 / 2	0 / 2	0 / 2	0 / 2	0 / 2	0 / 1	0 / 1	0 / 2
Data Type	Enumeration	Data	Signed Integer	Signed Integer	Signed Integer	Signed Integer	Unsigned Integer	Enumeration	Unsigned Integer	Unsigned Integer
Field Name	Status	Location Type	Coordinate 1	Coordinate 2	Coordinate 3	Power	Path Loss Exponent	Location Method	Quality Measure	Location Age

Figure 3.39 Format of the Location Data Response Payload

The fields of the payload correspond directly to the attributes with the same names. For details of their meaning and ranges see the descriptions of the individual attributes.

If the Absolute bit of the Location Type field is set to 1, the Location Method, Quality Measure and Location Age fields are not applicable and shall not be sent.

If the 2-D bit of the Location Type field is set to 1, the Coordinate 3 field shall not be sent.

The Status field indicates whether the response to the request was successful or not. If the field is set to SUCCESS, the response was successful. If the field is set to NOT_FOUND, the receiving device was unable to provide the location parameters of the device for which the location parameters were requested. If the field is set to NOT_FOUND, all other payload fields shall not be sent. See 2.5.3 for status codes.

3.13.2.4.3 Location Data Notification Command

This command is sent periodically by a device to announce its location information and channel parameters. The period is equal to the value of the *ReportingPeriod* attribute.

The location data notification command may be sent as a unicast or as a broadcast frame. When sent as a broadcast frame, it is recommended that the broadcast radius is set to 1.

3.13.2.4.3.1 Payload Format

The Location Data Notification command payload shall be formatted as illustrated in Figure 3.40.

Octets	**1**	**2**	**2**	**0 / 2**	**2**	**2**	**0 / 1**	**0 / 1**	**0 / 2**
Data Type	Data	Signed Integer	Signed Integer	Signed Integer	Signed Integer	Unsigned Integer	Enumeration	Unsigned Integer	Unsigned Integer
Field Name	Location Type	Coordinate 1	Coordinate 2	Coordinate 3	Power	Path Loss Exponent	Location Method	Quality Measure	Location Age

Figure 3.40 Format of the Location Data Notification Payload

The fields of the payload correspond directly to the attributes with the same names. For details of their meaning and ranges see the descriptions of the individual attributes.

If the 2-D bit of the Location Type field is set to 1, the Coordinate 3 field shall not be sent.

If the Absolute bit of the Location Type field is set to 1, the Location Method, Quality Measure and Location Age fields are not applicable and shall not be sent.

3.13.2.4.4 Compact Location Data Notification Command

This command is identical in format and use to the Location Data Notification command, except that the Power, Path Loss Exponent and Location Method fields are not included.

3.13.2.4.5 RSSI Ping Command

This command is sent periodically by a device to enable listening devices to measure the received signal strength in the absence of other transmissions from that device. The period is given by the *ReportingPeriod* attribute.

The RSSI Ping command may be sent as a unicast or as a broadcast frame. When sent as a broadcast frame, it is recommended that the broadcast radius is set to 1.

3.13.2.4.5.1 Payload Format

The RSSI Ping command payload shall be formatted as illustrated in Figure 3.41.

Octets	1
Data Type	Integer
Field Name	Location Type

Figure 3.41 Format of the RSSI Ping Payload

The Location Type field holds the value of the *LocationType* attribute.

3.13.2.5 Client

3.13.2.6 Dependencies

None

3.13.2.7 Attributes

None

3.13.2.8 Commands Received

The client receives the cluster specific commands generated by the server (see 3.13.2.4).

3.13.2.8.1 Commands Generated

The client generates the cluster specific commands received by the server (see 3.13.2.3), as required by the application.

This page left intentionally blank

CHAPTER 4

MEASUREMENT AND SENSING SPECIFICATION

4.1 General Description

4.1.1 Introduction

The clusters specified in this document are generic measurement and sensing interfaces that are sufficiently general to be of use across a wide range of application domains.

4.1.2 Cluster List

This section lists the clusters specified in this document, and gives examples of typical usage for the purpose of clarification.

The clusters specified in the Measurement and sensing functional domain are listed in Table 4.1 to Table 4.3.

4.1.2.1 Illuminance Measurement and Level Sensing

Table 4.1 Illuminance Measurement and Level Sensing Clusters

Cluster Name	Description
Illuminance measurement	Attributes and commands for configuring the measurement of illuminance, and reporting illuminance measurements
Illuminance level sensing	Attributes and commands for configuring the sensing of illuminance levels, and reporting whether illuminance is above, below, or on target.

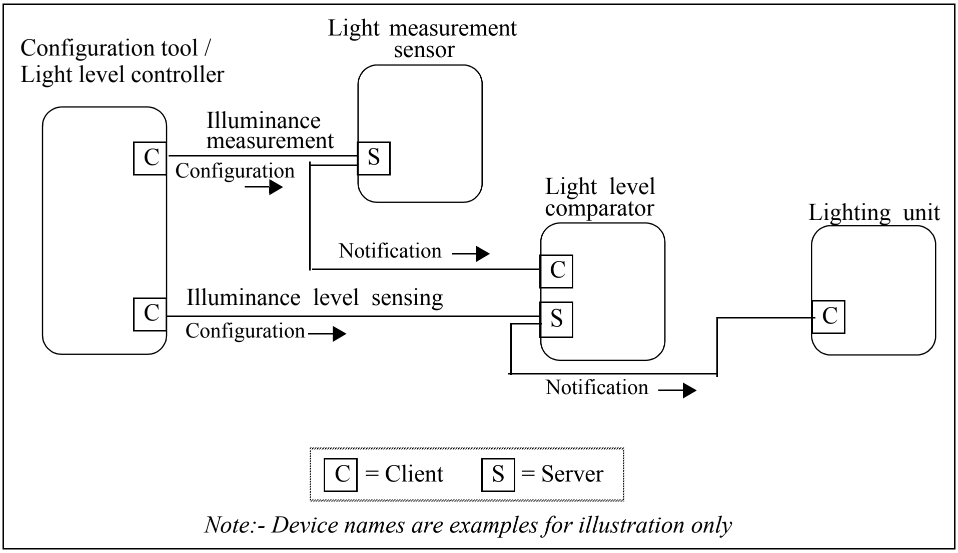


Figure 4.1 Typical Usage of Illuminance Measurement and Level Sensing Clusters

4.1.2.2 Temperature, Pressure and Flow Measurement

Table 4.2 Pressure and Flow Measurement Clusters

Cluster Name	Description
Temperature measurement	Attributes and commands for configuring the measurement of temperature, and reporting temperature measurements
Pressure measurement	Attributes and commands for configuring the measurement of pressure, and reporting pressure measurements
Flow measurement	Attributes and commands for configuring the measurement of flow, and reporting flow rates
Relative Humidity measurement	Attributes and commands for configuring the measurement of relative humidity, and reporting relative humidity measurements

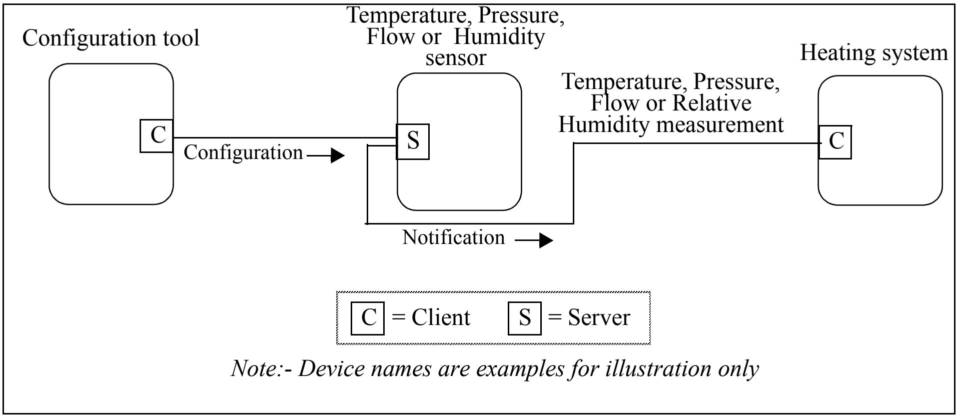
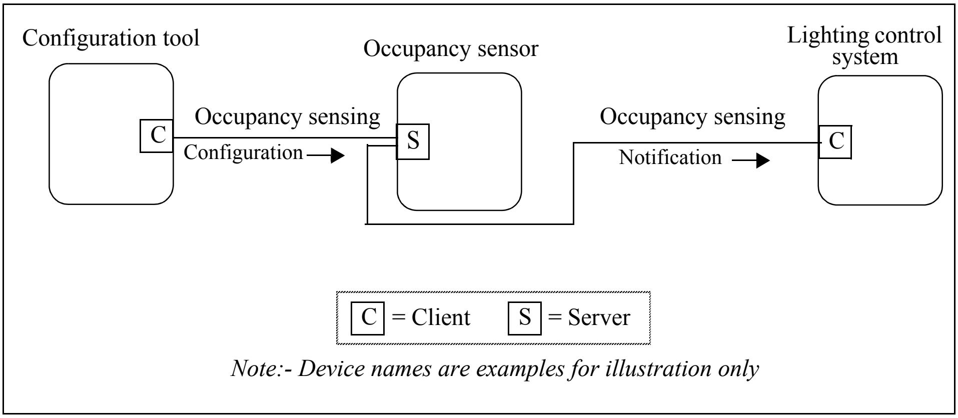

Configuration tool
Temperature, Pressure, Flow or Humidity sensor
Heating system
Temperature, Pressure, Flow or Relative Humidity measurement
C
Configuration
S
C
Notification
C = Client
S = Server
Note:- Device names are examples for illustration only

Figure 4.2 Typical Usage of Temperature, Pressure and Flow Measurement Clusters

4.1.2.3 Occupancy Sensing

Table 4.3 Occupancy Sensing Clusters

Cluster Name	Description
Occupancy sensing	Attributes and commands for configuring occupancy sensing, and reporting occupancy status



Note:- Device names are examples for illustration only

Figure 4.3 Typical Usage of Occupancy Sensing Cluster

4.2 Illuminance Measurement Cluster

4.2.1 Overview

The server cluster provides an interface to illuminance measurement functionality, including configuration and provision of notifications of illuminance measurements.

4.2.2 Server

4.2.2.1 Dependencies

None

4.2.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 4.4.

Table 4.4 Illuminance Measurement Attribute Sets

Attribute Set Identifier	Description
0x000	Illuminance Measurement Information
0x001 – 0xfff	Reserved

4.2.2.2.1 Illuminance Measurement Information Attribute Set

The Illuminance Measurement Information attribute set contains the attributes summarized in Table 4.5.

Table 4.5 Attributes of the Illuminance Measurement Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MeasuredValue	16-bit Unsigned Integer	*MinMeasuredValue* to *MaxMeasuredValue*	Read only	0	M
0x0001	MinMeasuredValue	16-bit Unsigned Integer	0x0002 – 0xfffd	Read only	-	M
0x0002	MaxMeasuredValue	16-bit Unsigned Integer	0x0001 – 0xfffe	Read only	-	M
0x0003	Tolerance	16-bit Unsigned Integer	0x0000 – 0x0800	Read only	-	O
0x0004	LightSensorType	8-bit Enumeration	0x00 – 0xff	Read only	-	O

4.2.2.2.1.1 *MeasuredValue* Attribute

MeasuredValue represents the Illuminance in Lux (symbol lx) as follows:-

MeasuredValue = 10,000 x $\log_{10}$ Illuminance + 1

Where 1 lx <= Illuminance <=3.576 Mlx, corresponding to a *MeasuredValue* in the range 1 to 0xfffe.

The following special values of *MeasuredValue* apply.

0x0000 indicates a value of Illuminance that is too low to be measured.

0xffff indicates that the Illuminance measurement is invalid.

MeasuredValue is updated continuously as new measurements are made.

4.2.2.2.1.2 *MinMeasuredValue* Attribute

The *MinMeasuredValue* attribute indicates the minimum value of *MeasuredValue* that can be measured. A value of 0xffff indicates that this attribute is not defined.

4.2.2.2.1.3 *MaxMeasuredValue* Attribute

The *MaxMeasuredValue* attribute indicates the maximum value of *MeasuredValue* that can be measured. A value of 0xffff indicates that this attribute is not defined.

MaxMeasuredValue shall be greater than *MinMeasuredValue.*

MinMeasuredValue and *MaxMeasuredValue* define the range of the sensor.

4.2.2.2.1.4 *Tolerance* Attribute

The *Tolerance* attribute indicates the magnitude of the possible error that is associated with *MeasuredValue* . The true value is located in the range (*MeasuredValue – Tolerance*) to (*MeasuredValue* + *Tolerance*).

4.2.2.2.1.5 *LightSensorType* Attribute

The *LightSensorType* attribute specifies the electronic type of the light sensor. This attribute shall be set to one of the non-reserved values listed in Table 4.6.

Table 4.6 Values of the *LightSensorType* Attribute

LightSensorType **Attribute Value**	**Description**
0x00	Photodiode
0x01	CMOS
0x02 – 0x3f	Reserved
0x40 – 0xfe	Reserved for manufacturer specific light sensor types
0xff	Unknown

4.2.2.3 Commands Received

No cluster specific commands are received by the server cluster.

4.2.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

4.2.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting intervals and reportable change settings described in the ZCL Foundation specification (see 2.4.7) and, where relevant, according to the *MinPercentChange* and *MinAbsoluteChange* settings in this cluster. The following attributes shall be reported:

MeasuredValue

Tolerance

4.2.3 Client

4.2.3.1 Dependencies

None.

4.2.3.2 Attributes

The Client cluster has no attributes.

4.2.3.3 Commands Received

No cluster specific commands are received by the client cluster.

4.2.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

4.3 Illuminance Level Sensing Cluster

4.3.1 Overview

The server cluster provides an interface to illuminance level sensing functionality, including configuration and provision of notifications of whether the illuminance is within, above or below a target band.

4.3.2 Server

4.3.2.1 Dependencies

None.

4.3.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 4.7.

Table 4.7 Illuminance Level Sensing Attribute Sets

Attribute Set Identifier	Description
0x000	Illuminance Level Sensing Information
0x001	Illuminance Level Sensing Settings
0x002 – 0xfff	Reserved

4.3.2.3 Illuminance Level Sensing Information Attribute Set

The light sensor configuration attribute set contains the attributes summarized in Table 4.8.

Table 4.8 Attributes of the Illuminance Level Sensing Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	LevelStatus	8-bit Enumeration	0x00 – 0xfe	Read only	-	M
0x0001	LightSensorType	8-bit Enumeration	0x00 – 0xfe	Read only	-	O

4.3.2.3.1 *LevelStatus* Attribute

The *LevelStatus* attribute indicates whether the measured illuminance is above, below, or within a band around *IlluminanceTargetLevel* (see 4.3.2.4.1). It may have any non-reserved value shown in Table 4.9.

Table 4.9 Values of the *LevelStatus* Attribute

LevelStatus Attribute Value	Description
0x00	Illuminance on target

Table 4.9 Values of the *LevelStatus* Attribute

LevelStatus Attribute Value	Description
0x01	Illuminance below target
0x02	Illuminance above target
0x03 – 0xff	Reserved

4.3.2.3.2 *LightSensorType* Attribute

The *LightSensorType* attribute specifies the electronic type of the light sensor. This attribute shall be set to one of the non-reserved values listed in Table 4.10.

Table 4.10 Values of the *LightSensorType* Attribute

LightSensorType Attribute Value	Description
0x00	Photodiode
0x01	CMOS
0x02 – 0x3f	Reserved
0x40 – 0xfe	Reserved for manufacturer specific light sensor types
0xff	Unknown

4.3.2.4 Illuminance Level Sensing Settings Attribute Set

The light sensor configuration attribute set contains the attributes summarized in Table 4.11.

Table 4.11 Attributes of the Illuminance Level Sensing Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	IlluminanceTargetLevel	Unsigned 16-bit integer	0x0000 – 0xfffe	Read/ Write	-	M

4.3.2.4.1 *IlluminanceTargetLevel* Attribute

The *IlluminanceTargetLevel* attribute specifies the target illuminance level. This target level is taken as the centre of a 'dead band', which must be sufficient in width, with hysteresis bands at both top and bottom, to provide reliable notifications without 'chatter'. Such a dead band and hysteresis bands must be provided by any implementation of this cluster. (N.B. Manufacturer specific attributes may be provided to configure these).

IlluminanceTargetLevel represents illuminance in Lux (symbol lx) as follows:

IlluminanceTargetLevel = 10,000 x $\log_{10}$ Illuminance

Where 1 lx <= Illuminance <=3.576 Mlx, corresponding to a *MeasuredValue* in the range 0 to 0xfffe.

A value of 0xffff indicates that this attribute is not valid.

4.3.2.5 Commands Received

No cluster specific commands are received by the server.

4.3.2.6 Commands Generated

No cluster specific commands are generated by the server cluster.

4.3.2.7 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval settings described in the ZCL Foundation Specification (see 2.4.7). The following attribute shall be reported:
LevelStatus

4.3.3 Client

4.3.3.1 Dependencies

None.

4.3.3.2 Attributes

The client cluster has no attributes.

4.3.3.3 Commands Received

No cluster specific commands are received by the client cluster.

4.3.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

4.4 Temperature Measurement Cluster

4.4.1 Overview

The server cluster provides an interface to temperature measurement functionality, including configuration and provision of notifications of temperature measurements.

4.4.2 Server

4.4.2.1 Dependencies

None.

4.4.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 4.12.

Table 4.12 Temperature Measurement Attribute Sets

Attribute Set Identifier	Description
0x000	Temperature Measurement Information
0x001 – 0xfff	Reserved

4.4.2.2.1 Temperature Measurement Information Attribute Set

The Temperature Measurement Information attribute set contains the attributes summarized in Table 4.13

Table 4.13 Attributes of the Temperature Measurement Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MeasuredValue	Signed 16-bit Integer	*MinMeasuredValue* to *MaxMeasuredValue*	Read only	0	M
0x0001	MinMeasuredValue	Signed 16-bit Integer	0x954d – 0x7ffe	Read only	-	M
0x0002	MaxMeasuredValue	Signed 16-bit Integer	0x954e – 0x7fff	Read only	-	M
0x0003	Tolerance	Unsigned 16-bit Integer	0x0000 – 0x0800	Read only	-	O

4.4.2.2.1.1 *MeasuredValue* Attribute

MeasuredValue represents the temperature in degrees Celsius as follows:-

MeasuredValue = 100 x temperature in degrees Celsius.

Where -273.15°C <= temperature <= 327.67 ºC, corresponding to a *MeasuredValue* in the range 0x954d to 0x7fff. The maximum resolution this format allows is 0.01 ºC.

A *MeasuredValue* of 0x8000 indicates that the temperature measurement is invalid.

MeasuredValue is updated continuously as new measurements are made.

4.4.2.2.1.2 *MinMeasuredValue* Attribute

The *MinMeasuredValue* attribute indicates the minimum value of *MeasuredValue* that is capable of being measured. A *MinMeasuredValue* of 0x8000 indicates that the minimum value is unknown.

4.4.2.2.1.3 *MaxMeasuredValue* Attribute

The *MaxMeasuredValue* attribute indicates the maximum value of *MeasuredValue* that is capable of being measured.

MaxMeasuredValue shall be greater than *MinMeasuredValue.*

MinMeasuredValue and *MaxMeasuredValue* define the range of the sensor.

A *MaxMeasuredValue* of 0x8000 indicates that the maximum value is unknown.

4.4.2.2.1.4 *Tolerance* Attribute

The *Tolerance* attribute indicates the magnitude of the possible error that is associated with *MeasuredValue* . The true value is located in the range (*MeasuredValue* – *Tolerance*) to (*MeasuredValue* + *Tolerance*).

4.4.2.3 Commands Received

No cluster specific commands are received by the server cluster.

4.4.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

4.4.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation specification (see 2.4.7) and, where relevant, according to the *MinPercentChange* and *MinAbsoluteChange* settings in this cluster. The following attribute shall be reported:

M*easuredValue*

*Tole*rance

4.4.3 Client

4.4.3.1 Dependencies

None.

4.4.3.2 Attributes

The Client cluster has no attributes.

4.4.3.3 Commands Received

No cluster specific commands are received by the client cluster.

4.4.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

4.5 Pressure Measurement Cluster

4.5.1 Overview

The server cluster provides an interface to pressure measurement functionality, including configuration and provision of notifications of pressure measurements.

4.5.2 Server

4.5.2.1 Dependencies

None

4.5.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 4.14.

Table 4.14 Pressure Measurement Attribute Sets

Attribute Set Identifier	Description
0x000	Pressure Measurement Information
0x001 – 0xfff	Reserved

4.5.2.2.1 Pressure Measurement Information Attribute Set

The Pressure Measurement Information attribute set contains the attributes summarized in Table 4.15.

Table 4.15 Attributes of the Pressure Measurement Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MeasuredValue	Signed 16-bit Integer	*MinMeasuredValue* to *MaxMeasuredValue*	Read only	0	M
0x0001	MinMeasuredValue	Signed 16-bit Integer	0x8001-0x7ffe	Read only	-	M
0x0002	MaxMeasuredValue	Signed 16-bit Integer	0x8002-0x7fff	Read only	-	M
0x0003	Tolerance	Unsigned 16-bit Integer	0x0000 – 0x0800	Read only	-	O

4.5.2.2.1.1 *MeasuredValue* Attribute

MeasuredValue represents the pressure in kPa as follows:-

MeasuredValue = 10 x Pressure

Where -3276.7 kPa <= Pressure <= 3276.7 kPa, corresponding to a *MeasuredValue* in the range 0x8001 to 0x7fff.

Note:- The maximum resolution this format allows is 0.1 kPa,.

A *MeasuredValue* of 0x8000 indicates that the pressure measurement is invalid.

MeasuredValue is updated continuously as new measurements are made.

4.5.2.2.1.2 *MinMeasuredValue* Attribute

The *MinMeasuredValue* attribute indicates the minimum value of *MeasuredValue* that can be measured. A value of 0x8000 means this attribute is not defined

4.5.2.2.1.3 *MaxMeasuredValue* Attribute

The *MaxMeasuredValue* attribute indicates the maximum value of *MeasuredValue* that can be measured. A value of 0x8000 means this attribute is not defined.

MaxMeasuredValue shall be greater than *MinMeasuredValue.*

MinMeasuredValue and *MaxMeasuredValue* define the range of the sensor.

4.5.2.2.1.4 *Tolerance* Attribute

The *Tolerance* attribute indicates the magnitude of the possible error that is associated with *MeasuredValue* . The true value is located in the range (*MeasuredValue* – *Tolerance*) to (*MeasuredValue* + *Tolerance*).

4.5.2.3 Commands Received

No cluster specific commands are received by the server cluster.

4.5.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

4.5.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation specification (see 2.4.7) and, where relevant, according to the *MinPercentChange* and *MinAbsoluteChange* settings in this cluster. The following attribute shall be reported:

MeasuredValue

Tolerance

4.5.3 Client

4.5.3.1 Dependencies

None.

4.5.3.2 Attributes

The Client cluster has no attributes.

4.5.3.3 Commands Received

No commands are received by the client cluster.

4.5.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

4.6 Flow Measurement Cluster

4.6.1 Overview

The server cluster provides an interface to flow measurement functionality, including configuration and provision of notifications of flow measurements.

4.6.2 Server

4.6.2.1 Dependencies

None

4.6.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets for are listed in Table 4.16.

Table 4.16 Flow Measurement Attribute Sets

Attribute Set Identifier	Description
0x000	Flow Measurement Information
0x001 – 0xfff	Reserved

4.6.2.2.1 Flow Measurement Information Attribute Set

The Flow Measurement Information attribute set contains the attributes summarized in Table 4.17.

Table 4.17 Attributes of the Flow Measurement Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MeasuredValue	Unsigned 16-bit Integer	*MinMeasuredValue* to *MaxMeasuredValue*	Read only	0	M
0x0001	MinMeasuredValue	Unsigned 16-bit Integer	0x0000 – 0xfffd	Read only	-	M
0x0002	MaxMeasuredValue	Unsigned 16-bit Integer	0x0001 – 0xfffe	Read only	-	M
0x0003	Tolerance	Unsigned 16-bit Integer	0x0000 – 0x0800	Read only	-	O

4.6.2.2.1.1 *MeasuredValue* Attribute

MeasuredValue represents the flow in m^3/h as follows:-

MeasuredValue = 10 x Flow

Where 0 m^3/h <= Flow <= 6,553.4 m^3/h, corresponding to a *MeasuredValue* in the range 0 to 0xfffe.

The maximum resolution this format allows is 0.1 m^3/h.

A *MeasuredValue* of 0xffff indicates that the pressure measurement is invalid.

MeasuredValue is updated continuously as new measurements are made.

4.6.2.2.1.2 *MinMeasuredValue* Attribute

The *MinMeasuredValue* attribute indicates the minimum value of *MeasuredValue* that can be measured. A value of 0xffff means this attribute is not defined

4.6.2.2.1.3 *MaxMeasuredValue* Attribute

The *MaxMeasuredValue* attribute indicates the maximum value of *MeasuredValue* that can be measured. A value of 0xffff means this attribute is not defined

MaxMeasuredValue shall be greater than *MinMeasuredValue.*

MinMeasuredValue and *MaxMeasuredValue* define the range of the sensor

4.6.2.2.1.4 *Tolerance* Attribute

The *Tolerance* attribute indicates the magnitude of the possible error that is associated with *MeasuredValue* . The true value is located in the range (*MeasuredValue – Tolerance*) to (*MeasuredValue* + *Tolerance*).

4.6.2.3 Commands Received

No cluster specific commands are received by the server cluster.

4.6.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

4.6.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation specification (see 2.4.7) and, where relevant, according to the *MinPercentChange* and *MinAbsoluteChange* settings in this cluster. The following attribute shall be reported:

MeasuredValue

Tolerance

4.6.3 Client

4.6.3.1 Dependencies

None.

4.6.3.2 Attributes

The Client cluster has no attributes.

4.6.3.3 Commands Received

No cluster specific commands are received by the client cluster.

4.6.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

4.7 Relative Humidity Measurement Cluster

4.7.1 Overview

The server cluster provides an interface to relative humidity measurement functionality, including configuration and provision of notifications of relative humidity measurements.

4.7.2 Server

4.7.2.1 Dependencies

None

4.7.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 4.18.

Table 4.18 Relative Humidity Measurement Attribute Sets

Attribute Set Identifier	Description
0x000	Relative Humidity Measurement Information
0x001 – 0xfff	Reserved

4.7.2.2.1 Relative Humidity Measurement Information Attribute Set

The Relative Humidity Measurement Information attribute set contains the attributes summarized in Table 4.19.

Table 4.19 Attributes of the Relative Humidity Measurement Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MeasuredValue	Unsigned 16-bit Integer	*MinMeasuredValue* to *MaxMeasuredValue*	Read only	-	M
0x0001	MinMeasuredValue	Unsigned 16-bit Integer	0x0000 – 0x270f	Read only	-	M
0x0002	MaxMeasuredValue	Unsigned 16-bit Integer	0x0001 – 0x2710	Read only	-	M
0x0003	Tolerance	Unsigned 16-bit Integer	0x0000 – 0x0800	Read only	-	O

4.7.2.2.1.1 *MeasuredValue* Attribute

MeasuredValue represents the relative humidity in % as follows:-

MeasuredValue = 100 x Relative humidity

Where 0% <= Relative humidity <= 100%, corresponding to a *MeasuredValue* in the range 0 to 0x2710.

The maximum resolution this format allows is 0.01%.

A *MeasuredValue* of 0xffff indicates that the measurement is invalid.

MeasuredValue is updated continuously as new measurements are made.

4.7.2.2.1.2 *MinMeasuredValue* Attribute

The *MinMeasuredValue* attribute indicates the minimum value of *MeasuredValue* that can be measured. A value of 0xffff means this attribute is not defined

4.7.2.2.1.3 *MaxMeasuredValue* Attribute

The *MaxMeasuredValue* attribute indicates the maximum value of *MeasuredValue* that can be measured. A value of 0xffff means this attribute is not defined

MaxMeasuredValue shall be greater than *MinMeasuredValue*.

MinMeasuredValue and *MaxMeasuredValue* define the range of the sensor

4.7.2.2.1.4 *Tolerance* Attribute

The *Tolerance* attribute indicates the magnitude of the possible error that is associated with *MeasuredValue* . The true value is located in the range (*MeasuredValue* – *Tolerance*) to (*MeasuredValue* + *Tolerance*).

4.7.2.3 Commands Received

No cluster specific commands are received by the server cluster.

4.7.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

4.7.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation specification (see 2.4.7) and, where relevant, according to the *MinPercentChange* and *MinAbsoluteChange* settings in this cluster. The following attribute shall be reported:

MeasuredValue
Tolerance

4.7.3 Client

4.7.3.1 Dependencies

None.

4.7.3.2 Attributes

The Client cluster has no attributes.

4.7.3.3 Commands Received

No cluster specific commands are received by the client cluster.

4.7.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

4.8 Occupancy Sensing Cluster

4.8.1 Overview

The server cluster provides an interface to occupancy sensing functionality, including configuration and provision of notifications of occupancy status.

4.8.2 Server

4.8.2.1 Dependencies

None.

4.8.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 4.20.

Table 4.20 Occupancy Sensor Attribute Sets

Attribute Set Identifier	Description
0x000	Occupancy sensor information
0x001	PIR configuration
0x002	Ultrasonic configuration
0x003 – 0xfff	Reserved

4.8.2.2.1 Occupancy Sensor Information Set

The occupancy sensor information attribute set contains the attributes summarized in Table 4.21.

Table 4.21 Attributes of the Occupancy Sensor Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	Occupancy	8-bit Bitmap	0000000x	Read only	-	M
0x0001	OccupancySensor Type	8-bit Enumeration	0x00 – 0xfe	Read only	-	M

4.8.2.2.1.1 *Occupancy* Attribute

The *Occupancy* attribute is a bitmap.

Bit 0 specifies the sensed occupancy as follows: 1 = occupied, 0 = unoccupied.

All other bits are reserved.

4.8.2.2.1.2 *OccupancySensorType* Attribute

The *OccupancySensorType* attribute specifies the type of the occupancy sensor. This attribute shall be set to one of the non-reserved values listed in Table 4.22.

Table 4.22 Values of the *OccupancySensorType* Attribute

OccupancySensorType Attribute Value	Description
0x00	PIR
0x01	Ultrasonic
0x02	PIR and ultrasonic
0x03 – 0xff	Reserved

4.8.2.2.2 PIR Configuration Set

The PIR sensor configuration attribute set contains the attributes summarized in Table 4.23.

Table 4.23 Attributes of the PIR Configuration Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	PIROccupiedTo UnoccupiedDelay	Unsigned 8-bit integer	0x00 – 0xfe	Read/ Write	0x00	O
0x0011	PIRUnoccupiedTo OccupiedDelay	Unsigned 8-bit integer	0x00 – 0xfe	Read/ write	0x00	O

4.8.2.2.2.1 *PIROccupiedToUnoccupiedTime* Attribute

The *PIROccupiedToUnoccupiedDelay* attribute is 8-bits in length and specifies the time delay, in seconds, before the PIR sensor changes to its occupied state when the sensed area becomes unoccupied. This attribute, along with *PIRUnoccupiedToOccupiedTime*, may be used to reduce sensor 'chatter' when used in an area where occupation changes frequently.

4.8.2.2.2.2 *PIRUnoccupiedToOccupiedTime* Attribute

The *PIRUnoccupiedToOccupiedDelay* attribute is 8-bits in length and specifies the time delay, in seconds, before the PIR sensor changes to its unoccupied state when the sensed area becomes occupied.

4.8.2.2.3 Ultrasonic Configuration Set

The ultrasonic sensor configuration attribute set contains the attributes summarized in Table 4.24.

Table 4.24 Attributes of the Ultrasonic Configuration Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0020	UltraSonicOccupiedTo UnoccupiedDelay	Unsigned 8-bit integer	0x00 – 0xfe	Read/ Write	0x00	O
0x0021	UltraSonicUnoccupiedTo OccupiedDelay	Unsigned 8-bit integer	0x00 – 0xfe	Read/ write	0x00	O

4.8.2.2.3.1 *UltraSonicOccupiedToUnoccupiedDelay* Attribute

The *UltraSonicOccupiedToUnoccupiedTime* attribute specifies the time delay, in seconds, before the ultrasonic sensor changes to its occupied state when the sensed area becomes unoccupied. This attribute, along with *UltraSonicUnoccupiedToOccupiedTime*, may be used to reduce sensor 'chatter' when used in an area where occupation changes frequently.

4.8.2.2.3.2 *UltraSonicUnoccupiedToOccupiedDelay* Attribute

The *UltraSonicUnoccupiedToOccupiedTime* attribute specifies the time delay, in seconds, before the ultrasonic sensor changes to its unoccupied state when the sensed area becomes occupied.

4.8.2.3 Commands Received

No cluster specific commands are received by the server cluster.

4.8.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

4.8.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval settings described in the ZCL Foundation specification (see 2.4.7). The following attribute shall be reported:

Occupancy

4.8.3 Client

4.8.3.1 Dependencies

None.

4.8.3.2 Attributes

The client cluster has no attributes.

4.8.3.3 Commands Received

No cluster specific commands are received by the client cluster.

4.8.3.4 Commands Generated

No cluster specific commands are generated by the client cluster.

CHAPTER 5

LIGHTING SPECIFICATION

5.1 General Description

5.1.1 Introduction

The clusters specified in this document are for use typically in ZigBee lighting applications, but may be used in any application domain.

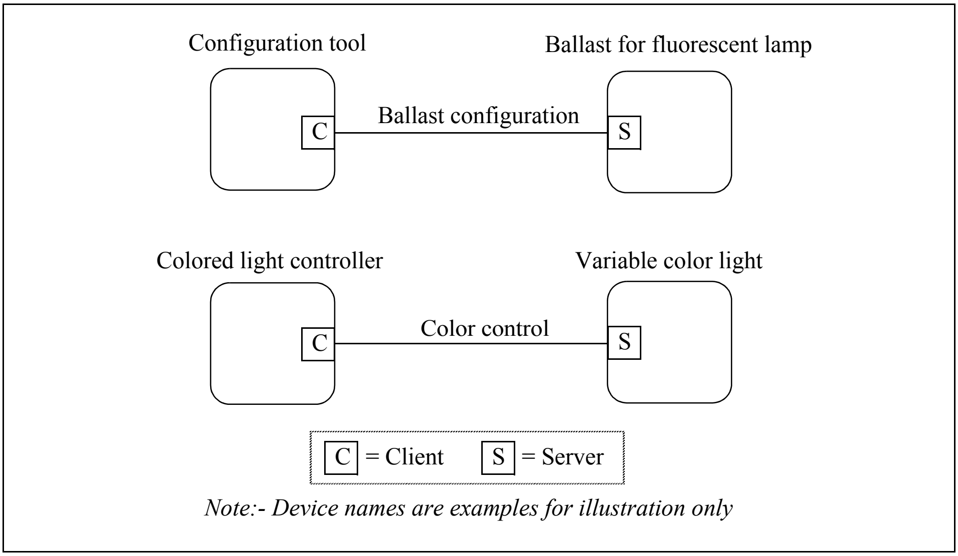
5.1.2 Cluster List

This section lists the clusters specified in this document, and gives examples of typical usage for the purpose of clarification.

The clusters specified in this document are listed in Table 5.1.

Table 5.1 Clusters Specified for the Lighting Functional Domain

Cluster Name	Description
Color Control	Attributes and commands for controlling the hue and saturation of a color-capable light.
Ballast Configuration	Attributes and commands for configuring a lighting ballast



Note:- Device names are examples for illustration only

Figure 5.1 Typical Usage of Ballast Configuration and Color Control Clusters

5.2 Color Control Cluster

5.2.1 Overview

This cluster provides an interface for changing the color of a light by controlling its hue and saturation. Control over luminance is not included, as this is provided by means of the Level Control cluster of the General library (see 3.10).

It is recommended that the hue and saturation are interpreted according to the HSV (aka HSB) color model.

5.2.2 Server

5.2.2.1 Dependencies

None

5.2.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set

and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 5.2.

Table 5.2 Hue Control Attribute Sets

Attribute Set Identifier	Description
0x000	Color Information
0x001 – 0xfff	Reserved

5.2.2.2.1 Color Information Attribute Set

The Color Information attribute set contains the attributes summarized in Table 5.3.

Table 5.3 Attributes of the Color Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	CurrentHue	Unsigned 8-bit integer	0x00 – 0xfe	Read only	0x00	M
0x0001	CurrentSaturation	Unsigned 8-bit integer	0x00 – 0xfe	Read only	0x00	M
0x0002	RemainingTime	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	0x00	O

5.2.2.2.1.1 *CurrentHue* Attribute

The *CurrentHue* attribute contains the current hue value of the light. It is updated as fast as practical during commands that change the hue.

The hue in degrees shall be related to the CurrentHue attribute by the relationship
Hue = CurrentHue x 360 / 254 (CurrentHue in the range 0 to 254 inclusive)

5.2.2.2.1.2 *CurrentSaturation* Attribute

The *CurrentSaturation* attribute holds the current saturation value of the light. It is updated as fast as practical during commands that change the hue.

The saturation shall be related to the CurrentSaturation attribute by the relationship
Saturation = CurrentSaturation / 254 (CurrentSaturation in the range 0 to 254 inclusive)

5.2.2.2.1.3 *RemainingTime* Attribute

The *RemainingTime* attribute holds the time remaining, in 1/10ths of a second, until the currently active command will be complete.

5.2.2.3 Commands Received

The command IDs for the Hue Control cluster are listed in Table 5.4.

Table 5.4 Command IDs for the Hue Light Control Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Move to Hue	M
0x01	Move Hue	M
0x02	Step Hue	M
0x03	Move to Saturation	M
0x04	Move Saturation	M
0x05	Step Saturation	M
0x06	Move to Hue and Saturation	M
0x07 – 0xff	Reserved	

5.2.2.3.1 Move to Hue Command

5.2.2.3.1.1 Payload Format

The Move to Hue command payload shall be formatted as illustrated in Figure 5.2.

Bits	8	8	16
Data Type	Unsigned 8-bit Integer	8-bit Enumeration	Unsigned 16-bit Integer
Field Name	Hue	Direction	Transition time

Figure 5.2 Format of the Move to Hue Command Payload

5.2.2.3.1.2 Hue Field

The Hue field specifies the hue to be moved to.

5.2.2.3.1.3 Direction Field

The Direction field shall be one of the non-reserved values in Table 5.5.

Table 5.5 Values of the Direction Field

Fade Mode Value	Description
0x00	Shortest distance
0x01	Longest distance
0x02	Up
0x03	Down
0x04 – 0xff	Reserved

5.2.2.3.1.4 Transition Time Field

The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to move to the new hue

5.2.2.3.1.5 Effect on Receipt

On receipt of this command, a device shall move from its current hue to the value given in the Hue field. The movement shall be continuous, i.e. not a step function, and the time taken to move to the new hue shall be equal to the Transition time field.

As hue is effectively measured on a circle, the new hue may be moved to in either direction. The direction of hue change is given by the Direction field. If Direction is 'Shortest distance', the direction is taken that involves the shortest path round the circle. This case corresponds to expected normal usage. If Direction is 'Longest distance', the direction is taken that involves the longest path round the circle. This case can be used for 'rainbow effects'. In both cases, if both distances are the same, the Up direction shall be taken.

5.2.2.3.2 Move Hue Command

5.2.2.3.2.1 Payload Format

The Move Hue command payload shall be formatted as illustrated in Figure 5.3.

Bits	8	8
Data Type	8-bit Enumeration	Unsigned 8-bit Integer
Field Name	Move mode	Rate

Figure 5.3 Format of the Move Hue Command Payload

5.2.2.3.2.2 Move Mode Field

The Move mode field shall be one of the non-reserved values in Table 5.6.

Table 5.6 Values of the Move Mode Field

Fade Mode Value	Description
0x00	Stop
0x01	Up
0x02	Reserved
0x03	Down
0x04 – 0xff	Reserved

5.2.2.3.2.3 Rate Field

The Rate field specifies the rate of movement in steps per second. A step is a change in the device’s hue of one unit. If the Rate field has a value of zero, the command has no effect and a default response command (see 2.4.12) is sent in response, with the status code set to INVALID_FIELD.

5.2.2.3.2.4 Effect on Receipt

On receipt of this command, a device shall move from its current hue in an up or down direction in a continuous fashion, as detailed in Table 5.7.

Table 5.7 Actions on Receipt for Move Command

Fade Mode	Action on Receipt
Stop	If moving, stop, else ignore the command (i.e. the command is accepted but has no effect).
Up	Increase the device's hue at the rate given in the Rate field. If the hue reaches the maximum allowed for the device, then proceed to its minimum allowed value.
Down	Decrease the device's hue at the rate given in the Rate field. If the hue reaches the minimum allowed for the device, then proceed to its maximum allowed value.

5.2.2.3.3 Step Hue Command

5.2.2.3.3.1 Payload Format

The Step Hue command payload shall be formatted as illustrated in Figure 5.4.

Bits	8	8
Data Type	8-bit Enumeration	Unsigned 8-bit Integer
Field Name	Step mode	Transition time

Figure 5.4 Format of the Step Hue Command Payload

5.2.2.3.3.2 Step Mode Field

The Move mode field shall be one of the non-reserved values in Table 5.8.

Table 5.8 Values of the Step Mode Field

Fade Mode Value	Description
0x00	Reserved
0x01	Up

Table 5.8 Values of the Step Mode Field (Continued)

Fade Mode Value	Description
0x02	Reserved
0x03	Down
0x04 – 0xff	Reserved

5.2.2.3.3.3 Transition Time Field

The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform a single step. A step is a change in the device's hue of one unit.

5.2.2.3.3.4 Effect on Receipt

On receipt of this command, a device shall move from its current hue in an up or down direction by one step, as detailed in Table 5.9.

Table 5.9 Actions on Receipt for Step Command

Fade Mode	Action on Receipt
Up	Increase the device's hue by one step. If the hue value is already the maximum value then proceed to the minimum allowed value.
Down	Decrease the device's hue by one step. If the hue value is already the minimum value then proceed to the maximum allowed value.

5.2.2.3.4 Move to Saturation Command

5.2.2.3.4.1 Payload Format

The Move to Saturation command payload shall be formatted as illustrated in Figure 5.5.

Bits	8	16
Data Type	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	Saturation	Transition time

Figure 5.5 Format of the Move to Saturation Command Payload

5.2.2.3.4.2 Effect on Receipt

On receipt of this command, a device shall move from its current saturation to the value given in the Saturation field.

The movement shall be continuous, i.e. not a step function, and the time taken to move to the new saturation shall be equal to the Transition time field, in 1/10ths of a second.

5.2.2.3.5 Move Saturation Command

5.2.2.3.5.1 Payload Format

The Move Saturation command payload shall be formatted as illustrated in Figure 5.6.

Bits	8	8
Data Type	8-bit Enumeration	Unsigned 8-bit Integer
Field Name	Move mode	Rate

Figure 5.6 Format of the Move Saturation Command Payload

5.2.2.3.5.2 Move Mode Field

The Move mode field shall be one of the non-reserved values in Table 5.10.

Table 5.10 Values of the Move Mode Field

Fade Mode Value	**Description**
0x00	Stop
0x01	Up
0x02	Reserved
0x03	Down
0x04 – 0xff	Reserved

5.2.2.3.5.3 Rate Field

The Rate field specifies the rate of movement in steps per second. A step is a change in the device’s saturation of one unit. If the Rate field has a value of zero, the command has no effect and a default response command (see 2.4.12) is sent in response, with the status code set to INVALID_FIELD.

5.2.2.3.5.4 Effect on Receipt

On receipt of this command, a device shall move from its current saturation in an up or down direction in a continuous fashion, as detailed in Table 5.11

Table 5.11 Actions on Receipt for Move Command

Fade Mode	Action on Receipt
Stop	If moving, stop, else ignore the command (i.e. the command is accepted but has no affect).
Up	Increase the device's saturation at the rate given in the Rate field. If the saturation reaches the maximum allowed for the device, stop.
Down	Decrease the device's saturation at the rate given in the Rate field. If the saturation reaches the minimum allowed for the device, stop.

5.2.2.3.6 Step Saturation Command

5.2.2.3.6.1 Payload Format

The Step Saturation command payload shall be formatted as illustrated in Figure 5.7.

Bits	8	8
Data Type	8-bit Enumeration	Unsigned 8-bit Integer
Field Name	Step mode	Transition time

Figure 5.7 Format of the Step Saturation Command Payload

5.2.2.3.6.2 Step Mode Field

The Step mode field shall be one of the non-reserved values in Table 5.12.

Table 5.12 Values of the Step Mode Field

Step Mode Value	Description
0x00	Reserved
0x01	Up

Table 5.12 Values of the Step Mode Field (Continued)

Step Mode Value	Description
0x02	Reserved
0x03	Down
0x04 – 0xff	Reserved

5.2.2.3.6.3 Transition Time Field

The Transition time field specifies, in 1/10ths of a second, the time that shall be taken to perform a single step. A step is a change in the device's saturation of one unit.

5.2.2.3.6.4 Effect on Receipt

On receipt of this command, a device shall move from its current saturation in an up or down direction by one step, as detailed in Table 5.13.

Table 5.13 Actions on Receipt for Step Command

Step Mode	Action on Receipt
Up	Increase the device's saturation by one step. However, if the saturation value is already the maximum value then do nothing.
Down	Decrease the device's saturation by one step. However, if the saturation value is already the minimum value then do nothing.

5.2.2.3.7 Move to Hue and Saturation Command

5.2.2.3.7.1 Payload Format

The Move to Hue and Saturation command payload shall be formatted as illustrated in Figure 5.8.

Bits	8	8	16
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer	Unsigned 16-bit Integer
Field Name	Hue	Saturation	Transition time

Figure 5.8 Format of the Move to Hue and Saturation Command Payload

5.2.2.3.7.2 Effect on Receipt

On receipt of this command, a device shall move from its current hue and saturation to the values given in the Hue and Saturation fields.

The movement shall be continuous, i.e. not a step function, and the time taken to move to the new saturation shall be equal to the Transition time field, in 1/10ths of a second.

The path through color space taken during the transition is not specified, but it is recommended that the shortest path is taken though hue/saturation space, i.e. movement is 'in a straight line' across the hue/saturation disk.

5.2.2.4 Commands Generated

The server generates no cluster specific commands

5.2.2.5 Scene Table Extensions

If the Scenes server cluster (see 3.7) is implemented, the following extension fields are added to the Scenes table:

- CurrentHue
- CurrentSaturation

5.2.2.6 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation specification (see 2.4.7). The following attributes shall be reported:

CurrentHue

CurrentSaturation

5.2.3 Client

5.2.3.1 Dependencies

None

5.2.3.2 Attributes

None

5.2.3.3 Commands Received

No cluster specific commands are received by the server.

5.2.3.4 Commands Generated

The client generates the commands detailed in 5.2.2.3, as required by the application.

5.3 Ballast Configuration Cluster

5.3.1 Overview

Attributes and commands for configuring a lighting ballast.

5.3.2 Server

5.3.2.1 Dependencies

For the alarm functionality specified by this cluster to be operational, the Alarms server cluster shall be implemented on the same endpoint.

5.3.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 5.14.

Table 5.14 Ballast Configuration Attribute Sets

Attribute Set Identifier	Description
0x000	Ballast information
0x001	Ballast settings
0x002	Lamp information
0x003	Lamp settings
0x004 – 0xfff	Reserved

5.3.2.2.1 Ballast Information Attribute Set

The Ballast Information attribute set contains the attributes summarized in Table 5.15.

Table 5.15 Attributes of the Ballast Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	PhysicalMinLevel	Unsigned 8-bit integer	0x01 – 0xfe	Read only	0x01	O
0x0001	PhysicalMaxLevel	Unsigned 8-bit integer	0x01 – 0xfe	Read only	0xfe	O
0x0002	BallastStatus	8-bit Bitmap	0000 00xx	Read only	0000 0000	M

5.3.2.2.1.1 *PhysicalMinLevel* Attribute

The *PhysicalMinLevel* attribute is 8-bits in length and specifies the minimum light level the ballast can achieve. This attribute shall be specified in the range 0x01 to 0xfe, and specifies the light output of the ballast according to the dimming light curve (see 5.3.4).

5.3.2.2.1.2 *PhysicalMaxLevel* Attribute

The *PhysicalMaxLevel* attribute is 8-bits in length and specifies the maximum light level the ballast can achieve. This attribute shall be specified in the range 0x01 to 0xfe, and specifies the light output of the ballast according to the dimming light curve (see 5.3.4).

5.3.2.2.1.3 *BallastStatus* Attribute

The *BallastStatus* attribute is 8-bits in length and specifies the activity status of the ballast functions. The usage of the bits is specified in Table 5.16. Where a function is active, the corresponding bit shall be set to 1. Where a function is not active, the corresponding bit shall be set to 0.

Table 5.16 Bit Usage of the *BallastStatus* Attribute

BallastStatus **Attribute Bit Number**	**Ballast Function**	**Details**
0	Non-operational	0 = The ballast is fully operational 1 = The ballast is not fully operational
1	Lamp not in socket	0 = All lamps are in their sockets 1 = One or more lamp is not in its socket
2 – 7	Reserved	-

5.3.2.2.2 Ballast Settings Attribute Set

The Ballast Settings attribute set contains the attributes summarized in Table 5.17.

Table 5.17 Attributes of the Ballast Settings Attribute Set

Identifier	**Name**	**Type**	**Range**	**Access**	**Default**	**Mandatory / Optional**
0x0010	MinLevel	Unsigned 8-bit integer	0x01 – 0xfe	Read/ write	Physical MinLevel	O
0x0011	MaxLevel	Unsigned 8-bit integer	0x01 – 0xfe	Read/ write	Physical MaxLevel	O
0x0012	PowerOnLevel	Unsigned 8-bit integer	0x00 – 0xfe	Read/ write	Physical MaxLevel	O
0x0013	PowerOn FadeTime	Unsigned 16-bit integer	0x0000 – 0xfffe	Read/ write	0x0000	O
0x0014	Intrinsic BallastFactor	Unsigned 8-bit integer	0x00 – 0xfe	Read/ write	-	O
0x0015	BallastFactor Adjustment	Unsigned 8-bit integer	0x64 – manufacturer dependent	Read/ write	0xff	O

5.3.2.2.2.1 *MinLevel* Attribute

The *MinLevel* attribute is 8-bits in length and specifies the minimum light level the ballast is permitted to use. This attribute shall be specified in the range 0x01 to 0xfe, and specifies the light output of the ballast according to the dimming light curve (see 7.4).

The value of this attribute shall be both greater than or equal to *PhysicalMinLevel* and less than or equal to *MaxLevel*. If an attempt is made to set this attribute to a level where these conditions are not met, a default response command shall be returned with status code set to INVALID_VALUE, and the level shall not be set.

5.3.2.2.2.2 *MaxLevel* Attribute

The *MaxLevel* attribute is 8-bits in length and specifies the maximum light level the ballast is permitted to use. This attribute shall be specified in the range 0x01 to 0xfe, and specifies the light output of the ballast according to the dimming light curve (see 5.3.4).

The value of this attribute shall be both less than or equal to *PhysicalMaxLevel* and greater than or equal to *MinLevel*. If an attempt is made to set this attribute to a level where these conditions are not met, a default response command shall be returned with status code set to INVALID_VALUE, and the level shall not be set.

5.3.2.2.2.3 *PowerOnLevel* Attribute

The *PowerOnLevel* attribute is 8-bits in length and specifies the light level to which the ballast will go when power is applied (e.g. when mains power is re-established after a power failure). This attribute shall be set to one of the values listed in Table 5.18.

Table 5.18 Values of the *PowerOnLevel* Attribute

***PowerOnLevel* Attribute Value**	**Description**
0x00 – 0xfe	Set to this specific light level, according to the dimming light curve (see 5.3.4).
0xff	Restore the light level being used prior to power failure.

The value of this attribute shall be both less than or equal to *PhysicalMaxLevel* and greater than or equal to *MinLevel*. If an attempt is made to set this attribute to a level where these conditions are not met, a default response command shall be returned with status code set to INVALID_VALUE, and the level shall not be set.

5.3.2.2.2.4 *PowerOnFadeTime* Attribute

The *PowerOnFadeTime* attribute is 16-bits in length and specifies the length of time, in tenths of a second, that the ballast takes to move to the light level specified in *PowerOnLevel* when power is applied (e.g. when mains power is re-established after a power failure).

5.3.2.2.2.5 *IntrinsicBallastFactor* Attribute

The *IntrinsicBallastFactor* attribute is 8-bits in length and specifies as a percentage the ballast factor of the ballast/lamp combination (see also clause 5.3), prior to any adjustment.

A value of 0xff indicates in invalid value.

5.3.2.2.2.6 *BallastFactorAdjustment* Attribute

The *BallastFactorAdjustment* attribute is 8-bits in length and specifies the multiplication factor, as a percentage, to be applied to the configured light output of the lamps (see also clause 5.3). A typical usage of this mechanism is to compensate for reduction in efficiency over the lifetime of a lamp.

The light output is given by

Actual light output = configured light output x *BallastFactorAdjustment* / 100%

The range for this attribute is manufacturer dependent. If an attempt is made to set this attribute to a level that cannot be supported, a default response command shall be returned with status code set to INVALID_VALUE, and the level shall not be set. The value 0xff indicates that ballast factor scaling is not in use.

5.3.2.2.3 Lamp Information Attribute Set

The lamp information attribute set contains the attributes summarized in Table 5.19.

Table 5.19 Attributes of the Lamp Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0020	LampQuantity	Unsigned 8-bit integer	0x00 – 0xfe	Read only	-	O

5.3.2.2.3.1 *LampQuantity* Attribute

The *LampQuantity* attribute is 8 bits in length and specifies the number of lamps connected to this ballast. (Note 1: this number does not take into account whether lamps are actually in their sockets or not).

5.3.2.2.4 Lamp Settings Attribute Set

The Lamp Settings attribute set contains the attributes summarized in Table 5.20. If *LampQuantity* is greater than one, each of these attributes is taken to apply to the lamps as a set. For example, all lamps are taken to be of the same *LampType* with the same *LampBurnHours.*

Table 5.20 Attributes of the Lamp Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0030	LampType	Character string	-	Read/write	Empty string	O
0x0031	LampManufacturer	Character string	-	Read/write	Empty string	O
0x0032	LampRatedHours	Unsigned 24-bit integer	0x000000 – 0xfffffe	Read/write	0xffffff	O
0x0033	LampBurnHours	Unsigned 24-bit integer	0x000000 – 0xfffffe	Read/write	0x000000	O
0x0034	LampAlarmMode	Bitmap (8-bit)	0000 000x	Read/write	0000 0000	O
0x0035	LampBurnHours TripPoint	Unsigned 24-bit integer	0x000000 – 0xfffffe	Read/write	0xffffff	O

5.3.2.2.4.1 *LampType* Attribute

The *LampType* attribute is an character string of up to 16 bytes in length. It specifies the type of lamps (including their wattage) connected to the ballast.

5.3.2.2.4.2 *LampManufacturer* Attribute

The *LampManufacturer* attribute is an character string of up to 16 bytes in length. It specifies the name of the manufacturer of the currently connected lamps.

5.3.2.2.4.3 *LampRatedHours* Attribute

The *LampRatedHours* attribute is 24-bits in length and specifies the number of hours of use the lamps are rated for by the manufacturer.

A value of 0xffffff indicates an invalid or unknown time.

5.3.2.2.4.4 *LampBurnHours* Attribute

The *LampBurnHours* attribute is 24-bits in length and specifies the length of time, in hours, the currently connected lamps have been operated, cumulative since the last re-lamping. Burn hours shall not be accumulated if the lamps are off.

This attribute should be reset to zero (e.g. remotely) when the lamp(s) are changed. If partially used lamps are connected, *LampBurnHours* should be updated to reflect the burn hours of the lamps.

A value of 0xffffff indicates an invalid or unknown time.

5.3.2.2.4.5 *LampAlarmMode* Attribute

The *LampsAlarmMode* attribute is 8-bits in length and specifies which attributes may cause an alarm notification to be generated, as listed in Table 5.21. A '1' in each bit position causes its associated attribute to be able to generate an alarm. (Note:-. All alarms are also logged in the alarm table – see Alarms cluster 3.11).

Table 5.21 Values of the *MainsAlarmMode* Attribute

MainsAlarmMode **Attribute Bit Number**	**Attribute**
0	LampBurnHours
1 – 7	Reserved

5.3.2.2.4.6 *LampBurnHoursTripPoint* Attribute

The *LampBurnHoursTripPoint* attribute is 24-bits in length and specifies the number of hours the *LampBurnHours* attribute may reach before an alarm is generated.

If the Alarms cluster is not present on the same device this attribute is not used and thus may be omitted (see 5.3.2.1).

The Alarm Code field included in the generated alarm shall be 0x01.

If this attribute takes the value 0xffffff then this alarm shall not be generated.

5.3.2.3 Commands Received

No commands are received by the server.

5.3.2.4 Commands Generated

The server generates no commands.

5.3.3 Client

5.3.3.1 Dependencies

None

5.3.3.2 Attributes

The client has no attributes.

5.3.3.3 Commands Received

No cluster specific commands are received by the server.

5.3.3.4 Commands Generated

No cluster specific commands are generated by the server.

5.3.4 The Dimming Light Curve

The dimming curve is recommended to be logarithmic, as defined by the following equation:

This page is intentionally blank

$$\%Light = 10^{\left(\frac{Level - 1}{\left(\frac{253}{3}\right)} - 1\right)}$$

Where:%Light is the percent light output of the ballast
Level is an 8-bit integer between 1 (0.1% light output) and 254 (100% output).

255 is reserved - the exact meaning of this value depends on the specific attribute or command.

Note:-. The light output is determined by this curve together with the *IntrinsicBallastFactor* and *BallastFactorAdjustment*. Attributes.

CHAPTER 6

HVAC SPECIFICATION

6.1 General Description

6.1.1 Introduction

The clusters specified in this document are for use typically in ZigBee HVAC applications, but may be used in any application domain.

6.1.2 Cluster List

This section lists the clusters specified in this document, and gives examples of typical usage for the purpose of clarification.

The clusters defined in this document are listed in Table 6.1.

Table 6.1 Clusters Specified in the HVAC Functional Domain

Cluster Name	Description
Pump Configuration and Control	An interface for configuring and controlling pumps.
Thermostat	An interface for configuring and controlling the functionality of a thermostat
Fan Control	An interface for controlling a fan in a heating / cooling system
Dehumidification Control	An interface for controlling dehumidification
Thermostat User Interface Configuration	An interface for configuring the user interface of a thermostat (which may be remote from the thermostat)

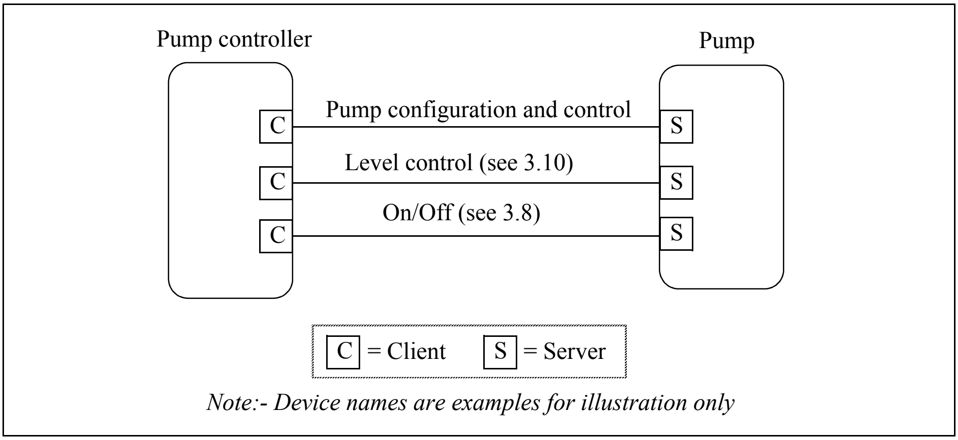


Figure 6.1 Typical Usage of the Pump Configuration and Control Cluster

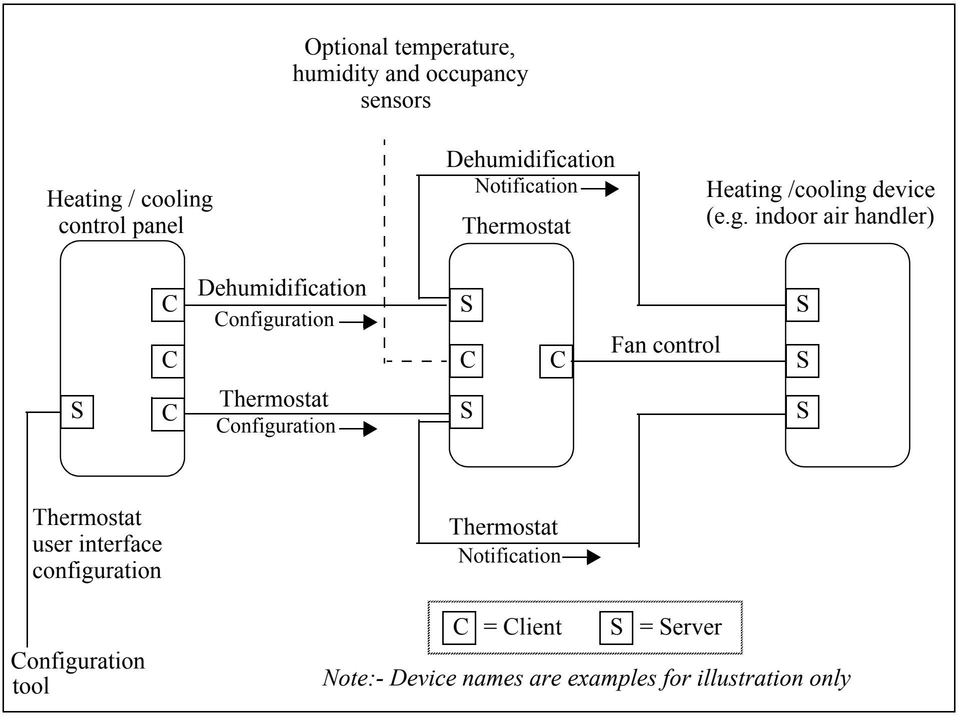

Optional temperature, humidity and occupancy sensors
Dehumidification
Notification
Heating / cooling control panel
Thermostat
Heating /cooling device (e.g. indoor air handler)
C
C
C
S
Dehumidification
Configuration
Thermostat
Configuration
S
C
S
C
Fan control
S
S
S
Thermostat
Notification
Thermostat user interface configuration
Configuration tool
C = Client
S = Server
Note:- Device names are examples for illustration only

Figure 6.2 Example Usage of the Thermostat and Related Clusters

6.2 Pump Configuration and Control Cluster

6.2.1 Overview

The Pump Configuration and Control cluster provides an interface for the setup and control of pump devices, and the automatic reporting of pump status information. Note that control of pump speed is not included – speed is controlled by the On/Off and Level Control clusters (see Figure 6.1).

6.2.2 Server

6.2.2.1 Dependencies

Where external pressure, flow and temperature measurements are processed by this cluster (see Table 6.8), these are provided by a Pressure Measurement cluster (4.5), a Flow Measurement cluster (4.6) and a Temperature Measurement client cluster (4.4) respectively. These 3 client clusters are used for connection to a remote sensor device. The pump is able to use the sensor measurement provided by a remote sensor for regulation of the pump speed.

For the alarms described in Table 6.9 to be operational, the Alarms server cluster (3.11) shall be implemented on the same endpoint.

Note that control of the pump setpoint is not included in this cluster – the On/Off and Level Control clusters (see Figure 6.1) may be used by a pump device to turn it on and off and control its setpoint. Note that the Pump Configuration and Control Cluster may override on/off/setpoint settings for specific operation modes (See section 6.2.2.2.3.1 for detailed description of the operation and control of the pump.).

6.2.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set

and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 6.2.

Table 6.2 Pump Configuration Attribute Sets

Attribute Set Identifier	Description
0x000	Pump Information
0x001	Pump Dynamic Information
0x002	Pump Settings
0x003 – 0xfff	Reserved

6.2.2.2.1 Pump Information Attribute Set

The pump information attribute set contains the attributes summarized in Table 6.3.

Table 6.3 Attributes of the Pump Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MaxPressure	Signed 16-bit integer	0x8001-0x7fff	Read only	-	M
0x0001	MaxSpeed	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	-	M
0x0002	MaxFlow	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	-	M
0x0003	MinConstPressure	Signed 16-bit integer	0x8001-0x7fff	Read only	-	O
0x0004	MaxConstPressure	Signed 16-bit integer	0x8001-0x7fff	Read only	-	O
0x0005	MinCompPressure	Signed 16-bit integer	0x8001-0x7fff	Read only	-	O
0x0006	MaxCompPressure	Signed 16-bit integer	0x8001-0x7fff	Read only	-	O
0x0007	MinConstSpeed	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	-	O
0x0008	MaxConstSpeed	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	-	O

Table 6.3 Attributes of the Pump Information Attribute Set (Continued)

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0009	MinConstFlow	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	-	O
0x000a	MaxConstFlow	Unsigned 16-bit integer	0x0000 – 0xfffe	Read only	-	O
0x000b	MinConstTemp	Signed 16-bit integer	0x954d – 0x7fff	Read only	-	O
0x000c	MaxConstTemp	Signed 16-bit integer	0x954d – 0x7fff	Read only	-	O

6.2.2.2.1.1 *MaxPressure* Attribute

The *MaxPressure* attribute specifies the maximum pressure the pump can achieve. It is a physical limit, and does not apply to any specific control mode or operation mode.

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is -3,276.7 kPa to 3,276.7 kPa (steps of 0.1 kPa)
The value -3,276.8 kPa (0x8000) indicates that this value is invalid.

6.2.2.2.1.2 *MaxSpeed* Attribute

The *MaxSpeed* attribute specifies the maximum speed the pump can achieve. It is a physical limit, and does not apply to any specific control mode or operation mode.

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is 0 to 65,534 RPM (steps of 1 RPM)
The value 65,535 RPM (0xffff) indicates that this value is invalid.

6.2.2.2.1.3 *MaxFlow* Attribute

The *MaxFlow* attribute specifies the maximum flow the pump can achieve. It is a physical limit, and does not apply to any specific control mode or operation mode.

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is 0 m^3/h to 6,553.4 m^3/h (steps of 0.1 m^3/h)
The value 6,553.5 m^3/h (0xffff) indicates that this value is invalid.

6.2.2.2.1.4 *MinConstPressure* Attribute

The *MinConstPressure* attribute specifies the minimum pressure the pump can achieve when it is running and working in control mode constant pressure (*ControlMode* attribute of the Pump settings attribute set is set to Constant pressure).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is –3,276.7 kPa to 3,276.7 kPa (steps of 0.1 kPa)
The value -3,276.8 kPa (0x8000) indicates that this value is invalid.

6.2.2.2.1.5 *MaxConstPressure* Attribute

The *MaxConstPressure* attribute specifies the maximum pressure the pump can achieve when it is working in control mode constant pressure (*ControlMode* attribute of the Pump settings attribute set is set to Constant pressure).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is –3,276.7 kPa to 3,276.7 kPa (steps of 0.1 kPa)
The value -3,276.8 kPa (0x8000) indicates that this value is invalid.

6.2.2.2.1.6 *MinCompPressure* Attribute

The *MinCompPressure* attribute specifies the minimum compensated pressure the pump can achieve when it is running and working in control mode Proportional pressure (*ControlMode* attribute of the Pump settings attribute set is set to Proportional pressure).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is –3,276.7 kPa to 3,276.7 kPa (steps of 0.1 kPa)
The value -3,276.8 kPa (0x8000) indicates that this value is invalid.

6.2.2.2.1.7 *MaxCompPressure* Attribute

The *MaxCompPressure* attribute specifies the maximum compensated pressure the pump can achieve when it is working in control mode Proportional pressure (*ControlMode* attribute of the Pump settings attribute set is set to Proportional pressure).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is –3,276.7 kPa to 3,276.7 kPa (steps of 0.1 kPa)
The value -3,276.8 kPa (0x8000) indicates that this value is invalid..

6.2.2.2.1.8 *MinConstSpeed* Attribute

The *MinConstSpeed* attribute specifies the minimum speed the pump can achieve when it is running and working in control mode Constant speed (*ControlMode* attribute of the Pump settings attribute set is set to Constant speed).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is 0 to 65,534 RPM (steps of 1 RPM)
The value 65,535 RPM (0xffff) indicates that this value is invalid.

6.2.2.2.1.9 *MaxConstSpeed* Attribute

The *MaxConstSpeed* attribute specifies the maximum speed the pump can achieve when it is working in control mode Constant speed (*ControlMode* attribute of the Pump settings attribute set is set to Constant speed).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is 0 to 65,534 RPM (steps of 1 RPM)
The value 65,535 RPM (0xffff) indicates that this value is invalid.

6.2.2.2.1.10 *MinConstFlow* Attribute

The *MinConstFlow* attribute specifies the minimum flow the pump can achieve when it is running and working in control mode Constant flow (*ControlMode* attribute of the Pump settings attribute set is set to Constant flow).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is 0 m^3/h to 6,553.4 m^3/h (steps of 0.1 m^3/h)
The value 6,553.5 m^3/h (0xffff) indicates that this value is invalid.

6.2.2.2.1.11 *MaxConstFlow* Attribute

The *MaxConstFlow* attribute specifies the maximum flow the pump can achieve when it is running and working in control mode Constant flow (*ControlMode* attribute of the Pump settings attribute set is set to Constant flow).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is 0 m^3/h to 6,553.4 m^3/h (steps of 0.1 m^3/h).
The value 6,553.5 m^3/h (0xffff) indicates that this value is invalid.

6.2.2.2.1.12 ***MinConstTemp*** Attribute

The *MinConstTemp* attribute specifies the minimum temperature the pump can maintain in the system when it is running and working in control mode Constant temperature (*ControlMode* attribute of the Pump settings attribute set is set to Constant temperature).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value.

Valid range is –273.15 C to 327.67 C (steps of 0.01 C).
The value -327.68 C (0x8000) indicates that this value is invalid.

6.2.2.2.1.13 ***MaxConstTemp*** Attribute

The *MaxConstTemp* attribute specifies the maximum temperature the pump can maintain in the system when it is running and working in control mode Constant temperature (*ControlMode* attribute of the Pump settings attribute set is set to Constant temperature).

This attribute is read only, and can only be set by the manufacturer. If the value is not available, this attribute will display the invalid value. *MaxConstTemp* shall be greater than or equal to *MinConstTemp*

Valid range is –273.15 C to 327.67 C (steps of 0.01 C).
The value -327.68 C (0x8000) indicates that this value is invalid.

6.2.2.2.2 Pump Dynamic Information Attribute Set

The pump dynamic information attribute set contains the attributes summarized in Table 6.4.

Table 6.4 Attributes of the Pump Dynamic Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	PumpStatus	16-bit Bitmap	-	Read only	-	O
0x0011	EffectiveOperationMode	8-bit Enumeration	0x00 – 0xfe	Read only	-	M
0x0012	EffectiveControlMode	8-bit Enumeration	0x00 – 0xfe	Read only	-	M

Table 6.4 Attributes of the Pump Dynamic Information Attribute Set (Continued)

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0013	Capacity	Signed 16-bit integer	0x0000-0x7fff	Read only	-	M
0x0014	Speed	Unsigned 16-bit integer	0x0000 - 0xfffe	Read only	-	O
0x0015	LifetimeRunningHours	Unsigned 24-bit integer	0x000000 - 0xfffffe	Read / Write	0	O
0x0016	Power	Unsigned 24-bit integer	0x000000 - 0xfffffe	Read / Write	-	O
0x0017	LifetimeEnergyConsumed	Unsigned 32-bit integer	0x00000000 - 0xfffffffe	Read only	0	O

6.2.2.2.2.1 *PumpStatus* Attribute

The *PumpStatus* attribute specifies the activity status of the pump functions listed in Table 6.5. Where a pump controller function is active, the corresponding bit shall be set to 1. Where a pump controller function is not active, the corresponding bit shall be set to 0.

Table 6.5 Values of the *PumpStatus* Attribute

PumpStatus Attribute Bit Number	Pump Function	Remarks
0	Device fault	A fault related to the pump device is detected (Corresponds to a Alarm code in the range 6-13, see Table 6.9)
1	Supply fault	A fault related to the supply to the pump is detected (Corresponds to a Alarm code in the range 0-5 or 13, see Table 6.9)
2	Speed low	Setpoint is too low to achieve
3	Speed high	Setpoint is too high to achieve
4	Local override	The pump is overridden by local control
5	Running	Pump is currently running

Table 6.5 Values of the *PumpStatus* Attribute (Continued)

6	Remote Pressure	A remote pressure sensor is used as the sensor for the regulation of the pump. *EffectiveControlMode* is Constant pressure, and the setpoint for the pump is interpreted as a percentage of the range of the remote sensor ([*MinMeasuredValue – MaxMeasuredValue*])
7	Remote Flow	A remote flow sensor is used as the sensor for the regulation of the pump. *EffectiveControlModeI* is Constant flow, and the setpoint for the pump is interpreted as a percentage of the range of the remote sensor ([*MinMeasuredValue – MaxMeasuredValue*])
8	Remote Temperature	A remote temperature sensor is used as the sensor for the regulation of the pump. *EffectiveControlModeI* is Constant temperature, and setpoint is interpreted as a percentage of the range of the remote sensor ([*MinMeasuredValue – MaxMeasuredValue*])
9 – 15	Reserved	-

6.2.2.2.2.2 *EffectiveOperationMode* Attribute

The *EffectiveOperationMode* attribute specifies current effective operation mode of the pump. The value of the *EffectiveOperationMode* attribute is the same as the *OperationMode* attribute of the Pump settings attribute set, except when it is overridden locally. See section 6.2.2.2.3.1 for a detailed description of the operation and control of the pump.

This attribute is read only.

Valid range is defined by the operation modes listed in Table 6.1.

6.2.2.2.2.3 *EffectiveControlMode* Attribute

The *EffectiveControlMode* attribute specifies the current effective control mode of the pump.

The *EffectiveControlMode* attribute contains the control mode that currently applies to the pump. It will have the value of the *ControlMode* attribute, unless a remote sensor is used as the sensor for regulation of the pump. In this case, *EffectiveControlMode* will display Constant pressure, Constant flow or Constant temperature if the remote sensor is a pressure sensor, a flow sensor or a temperature sensor respectively, regardless of the value of the *ControlMode* attribute.

See section 6.2.2.2.3.1 for detailed description of the operation and control of the pump. This attribute is read only.

Valid range is defined by the control modes listed in Table 6.8.

6.2.2.2.2.4 *Capacity* Attribute

The *Capacity* attribute specifies the actual capacity of the pump as a percentage of the effective maximum setpoint value. It is updated dynamically as the speed of the pump changes.

This attribute is read only. If the value is not available (the measurement or estimation of the speed is done in the pump), this attribute will contain the invalid value.

Valid range is 0 % to 163.835% (0.005 % granularity). Although the *Capacity* attribute is a signed value, values of capacity less than zero have no physical meaning.
The value -163.840 % (0x8000) indicates that this value is invalid.

6.2.2.2.2.5 *Speed* Attribute

The *Speed* attribute specifies the actual speed of the pump measured in RPM. It is updated dynamically as the speed of the pump changes.

This attribute is read only. If the value is not available (the measurement or estimation of the speed is done in the pump), this attribute will contain the invalid value.

Valid range is 0 to 65.534 RPM
The value 65.535 RPM (0xffff) indicates that this value is invalid.

6.2.2.2.2.6 *LifetimeRunningHours* Attribute

The *LifetimeRunningHours* attribute specifies the accumulated number of hours, that the pump has been powered and the motor has been running. It is updated dynamically as it increases. It is preserved over power cycles of the pump. if *LifeTimeRunningHours* rises above maximum value it “rolls over” and starts at 0 (zero).

This attribute is writeable, in order to allow setting to an appropriate value after maintenance. If the value is not available, this attribute will contain the invalid value.

Valid range is 0 to 16,777,214 hrs.
The value 16,777,215 (0xffffff) indicates that this value is unknown.

6.2.2.2.2.7 *Power* Attribute

The *Power* attribute specifies the actual power consumption of the pump in Watts. The value of the *Power* attribute is updated dynamically as the power consumption of the pump changes.

This attribute is read only. If the value is not available (the measurement of power consumption is not done in the pump), this attribute will display the invalid value.

Valid range is 0 to 16,777,214 Watts.
The value 16,777,215 (0xffffff) indicates that this value is unknown.

6.2.2.2.2.8 *LifetimeEnergyConsumed* Attribute

The *LifetimeEnergyConsumed* attribute specifies the accumulated energy consumption of the pump through the entire lifetime of the pump in kWh. The value of the *LifetimeEnergyConsumed* attribute is updated dynamically as the energy consumption of the pump increases. If *LifetimeEnergyConsumed* rises above maximum value it “rolls over” and starts at 0 (zero).

This attribute is writeable, in order to allow setting to an appropriate value after maintenance.

Valid range is 0 kWh to 4,294,967,294 kWh.
The value 4,294,967,295 (0xffffffff) indicates that this value is unknown.

6.2.2.2.3 Pump Settings Attribute Set

The pump settings attribute set contains the attributes summarized in Table 6.6.

Table 6.6 Attributes of the Pump Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0020	OperationMode	8-bit Enumeration	0x00 – 0xfe	Read / Write	0x0	M
0x0021	ControlMode	8-bit Enumeration	0x00 – 0xfe	Read / Write	0x0	O
0x0022	AlarmMask	16-bit bitmap	-	Read only	-	O

6.2.2.2.3.1 *OperationMode* Attribute

The *OperationMode* attribute specifies the operation mode of the pump. This attribute shall have one of the values listed in Table 6.7.

The actual operating mode of the pump is a result of the setting of the attributes *OperationMode*, *ControlMode* and the optional connection of a remote sensor. The operation and control is prioritized as shown in the scheme in the figure below:

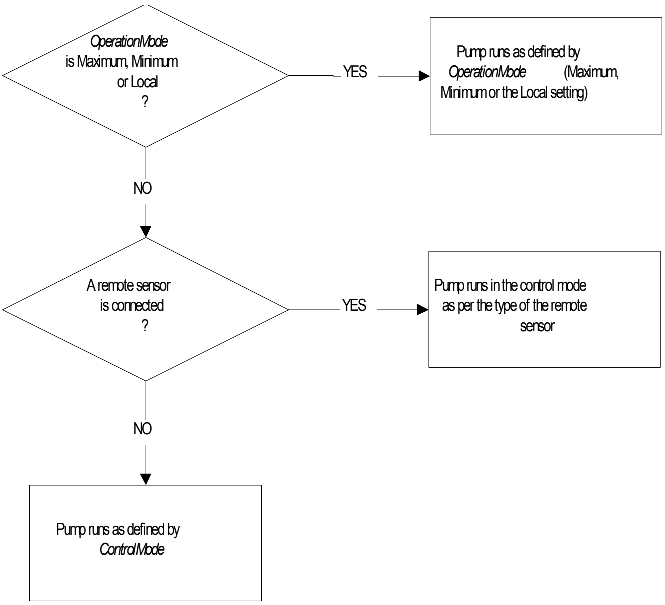

OperationMode is Maximum, Minimum or Local ?
YES
Pump runs as defined by OperationMode (Maximum, Minimum or the Local setting)
NO
A remote sensor is connected ?
YES
Pump runs in the control mode as per the type of the remote sensor
NO
Pump runs as defined by ControlMode

Figure 6.3 Priority Scheme of Pump Operation and Control

If the *OperationMode* attribute is Maximum, Minimum or Local, the *OperationMode* attribute decides how the pump is operated.

If the *OperationMode* attribute is Normal and a remote sensor is connected to the pump, the type of the remote sensor decides the control mode of the pump. A connected remote pressure sensor will make the pump run in control mode Constant pressure and vice versa for flow and temperature type sensors. This is regardless of the setting of the *ControlMode* attribute.

If the *OperationMode* attribute is Normal and no remote sensor is connected, the control mode of the pump is decided by the *ControlMode* attribute.

OperationMode may be changed at any time, even when the pump is running. The behavior of the pump at the point of changing the value of the *OperationMode* attribute is vendor specific.

Table 6.7 Values of the *OperationMode* Attribute

***OperationMode* Attribute Value**	**Name**	**Explanation**
0	Normal	The pump is controlled by a setpoint, as defined by a connected remote sensor or by the *ControlMode* attribute. (N.B. The setpoint is an internal variable which may be controlled between 0% and 100%, e.g. by means of the Level Control cluster 3.10)
1	Minimum	This value sets the pump to run at the minimum possible speed it can without being stopped
2	Maximum	This value sets the pump to run at its maximum possible speed
3	Local	This value sets the pump to run with the local settings of the pump, regardless of what these are
4-254	Reserved	Reserved for future use

6.2.2.2.3.2 *ControlMode* Attribute

The *ControlMode* attribute specifies the control mode of the pump. This attribute shall have one of the values listed in Table 6.8.

See section 6.2.2.2.3.1 for detailed description of the operation and control of the pump.

ControlMode may be changed at any time, even when the pump is running. The behavior of the pump at the point of changing is vendor specific.

Table 6.8 Values of the *ControlMode* Attribute

ControlMode Attribute Value	Name	Explanation
0	Constant speed	The pump is running at a constant speed. The setpoint is interpreted as a percentage of the *MaxSpeed* attribute
1	Constant pressure	The pump will regulate its speed to maintain a constant differential pressure over its flanges. The setpoint is interpreted as a percentage of the range of the sensor used for this control mode. In case of the internal pressure sensor, this will be the range derived from the [*MinConstPressure - MaxConstPressure*] attributes. In case of a remote pressure sensor, this will be the range derived from the [*MinMeasuredValue – MaxMeasuredValue*] attributes of the remote pressure sensor.
2	Proportional pressure	The pump will regulate its speed to maintain a constant differential pressure over its flanges. The setpoint is interpreted as a percentage of the range derived of the [*MinCompPressure - MaxCompPressure*] attributes. The internal setpoint will be lowered (compensated) dependant on the flow in the pump (lower flow => lower internal setpoint)
3	Constant flow	The pump will regulate its speed to maintain a constant flow through the pump. The setpoint is interpreted as a percentage of the range of the sensor used for this control mode. In case of the internal flow sensor, this will be the range derived from the [*MinConstFlow - MaxConstFlow*] attributes. In case of a remote flow sensor, this will be the range derived from the [*MinMeasuredValue – MaxMeasuredValue*] attributes of the remote flow sensor.
4	Reserved	-
5	Constant temperature	The pump will regulate its speed to maintain a constant temperature. The setpoint is interpreted as a percentage of the range of the sensor used for this control mode. In case of the internal temperature sensor, this will be the range derived from the [*MinConstTemp - MaxConstTemp*] attributes. In case of a remote temperature sensor, this will be the range derived from the [*MinMeasuredValue – MaxMeasuredValue*] attributes of the remote temperature sensor.

Table 6.8 Values of the *ControlMode* Attribute (Continued)

ControlMode Attribute Value	Name	Explanation
6	Reserved	-
7	Automatic	The operation of the pump is automatically optimized to provide the most suitable performance with respect to comfort and energy savings. This behavior is manufacturer defined. The pump can be stopped by setting the setpoint of the level control cluster to 0 of by using the On/Off cluster. If the pump is started (at any setpoint), the speed of the pump is entirely determined by the pump.
8-254	Reserved	-

6.2.2.2.3.3 *AlarmMask* Attribute

The *AlarmMask* attribute specifies whether each of the alarms listed in Table 6.9 is enabled. When the bit number corresponding to the alarm code is set to 1, the alarm is enabled, else it is disabled. Bits not corresponding to a code in the table (bits 14, 15) are reserved.

When the Alarms cluster is implemented on a device, and one of the alarm conditions included in this table occurs, an alarm notification is generated, with the alarm code field set as listed in the table.

Table 6.9 Alarm Codes

Alarm Code	Alarm Condition
0	Supply voltage too low
1	Supply voltage too high
2	Power missing phase
3	System pressure too low
4	System pressure too high
5	Dry running
6	Motor temperature too high
7	Pump motor has fatal failure
8	Electronic temperature too high
9	Pump blocked

Table 6.9 Alarm Codes

Alarm Code	Alarm Condition
10	Sensor failure
11	Electronic non fatal failure
12	Electronic fatal failure
13	General fault

6.2.2.3 Commands Received

None.

6.2.2.4 Commands Generated

None.

6.2.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command, according to the minimum and maximum reporting interval, reportable change, and timeout period settings described in the ZCL Foundation Specification (see 2.4.7).

The following attributes shall be reported:

PumpStatus

Capacity

6.2.3 Client

6.2.3.1 Dependencies

None

6.2.3.2 Attributes

The client supports no attributes.

6.2.3.3 Commands Received

The client receives no cluster specific commands.

6.2.3.4 Commands Generated

The client generates no cluster specific commands.

6.3 Thermostat Cluster

6.3.1 Overview

This cluster provides an interface to the functionality of a thermostat.

6.3.2 Server

6.3.2.1 Dependencies

For alarms to be generated by this cluster, the Alarms server cluster (see 3.11) shall be included on the same endpoint. For remote temperature sensing, the Temperature Measurement client cluster (see 4.4) may be included on the same endpoint. For occupancy sensing, the Occupancy Sensing client cluster (see 4.8) may be included on the same endpoint.

6.3.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets for Thermostat are listed in Table 6.10.

Table 6.10 Currently Defined Thermostat Attribute Sets

Attribute Set Identifier	Description
0x000	Thermostat Information
0x001	Thermostat Settings
0x002 – 0x3ff	Reserved
0x400 – 0xfff	Reserved for vendor specific attributes

6.3.2.2.1 Thermostat Information Attribute Set

The Thermostat Information attribute set contains the attributes summarized in Table 6.11.

Table 6.11 Attributes of the Thermostat Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	LocalTemperature	Signed 16-bit Integer	0x954d – 0x7fff	Read	-	M
0x0001	OutdoorTemperature	Signed 16-bit Integer	0x954d – 0x7fff	Read	-	O
0x0002	Ocupancy	8-bit bitmap	0000000x	Read	00000000	O
0x0003	AbsMinHeatSetpointLimit	Signed 16-bit Integer	0x954d – 0x7fff	Read	0x02bc (7°C)	O
0x0004	AbsMaxHeatSetpointLimit	Signed 16-bit Integer	0x954d – 0x7fff	Read	0x0bb8 (30°C)	O
0x0005	AbsMinCoolSetpointLimit	Signed 16-bit Integer	0x954d – 0x7fff	Read	0x0640 (16°C)	O
0x0006	AbsMaxCoolSetpointLimit	Signed 16-bit Integer	0x954d – 0x7fff	Read	0x0c80 (32°C)	O
0x0007	PICoolingDemand	Unsigned 8-bit Integer	0x00 – 0x64	Read	-	O
0x0008	PIHeatingDemand	Unsigned 8-bit Integer	0x00 – 0x64	Read	-	O

6.3.2.2.1.1 *LocalTemperature* Attribute

LocalTemperature represents the temperature in degrees Celsius, as measured locally or remotely (over the network) as follows:-

LocalTemperature = 100 x temperature in degrees Celsius.

Where -273.15°C <= temperature <= 327.67 ºC, corresponding to a *LocalTemperature* in the range 0x954d to 0x7fff.

The maximum resolution this format allows is 0.01 ºC.

A *LocalTemperature* of 0x8000 indicates that the temperature measurement is invalid.

6.3.2.2.1.2 *OutdoorTemperature* Attribute

OutdoorTemperature represents the outdoor temperature in degrees Celsius, as measured locally or remotely (over the network). It is measured as described for *LocalTemperature.*

6.3.2.2.1.3 *Occupancy* Attribute

Occupancy specifies whether the heated/cooled space is occupied or not, as measured locally or remotely (over the network). If bit 0 = 1, the space is occupied, else it is unoccupied. All other bits are reserved.

6.3.2.2.1.4 *AbsMinHeatSetpointLimit* Attribute

The *MinHeatSetpointLimit* attribute specifies the absolute minimum level that the heating setpoint may be set to. This is a limitation imposed by the manufacturer. The value is calculated as described in the LocalTemperature attribute.

6.3.2.2.1.5 *AbsMaxHeatSetpointLimit* Attribute

The *MaxHeatSetpointLimit* attribute specifies the absolute maximum level that the heating setpoint may be set to. This is a limitation imposed by the manufacturer. The value is calculated as described in the LocalTemperature attribute.

6.3.2.2.1.6 *AbsMinCoolSetpointLimit* Attribute

The *MinCoolSetpointLimit* attribute specifies the absolute minimum level that the cooling setpoint may be set to. This is a limitation imposed by the manufacturer. The value is calculated as described in the LocalTemperature attribute.

6.3.2.2.1.7 *AbsMaxCoolSetpointLimit* Attribute

The *MaxCoolSetpointLimit* attribute specifies the absolute maximum level that the cooling setpoint may be set to. This is a limitation imposed by the manufacturer. The value is calculated as described in the LocalTemperature attribute.

6.3.2.2.1.8 *PICoolingDemand* Attribute

The *PICoolingDemand* attribute is 8-bits in length and specifies the level of cooling demanded by the PI (proportional integral) control loop in use by the thermostat (if any), in percent. This value is 0 when the thermostat is in “off” or “heating” mode.

This attribute is reported regularly and may be used to control a heating device.

6.3.2.2.1.9 *PIHeatingDemand* Attribute

The *PIHeatingDemand* attribute is 8-bits in length and specifies the level of heating demanded by the PI loop in percent. This value is 0 when the thermostat is in “off” or “cooling” mode.

This attribute is reported regularly and may be used to control a cooling device.

6.3.2.2.2 Thermostat Settings Attribute Set

The Thermostat settings attribute set contains the attributes summarized in Table 6.12.

Table 6.12 Attributes of the Thermostat Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	LocalTemperature Calibration	Signed 8-bit Integer	0xE7 – 0x19	Read / Write	0x00 (0°C)	O
0x0011	OccupiedCooling Setpoint	Signed 16-bit Integer	*MinCoolSetpoint Limit – MaxCoolSetpoint Limit*	Read / Write	0x0a28 (26°C)	M
0x0012	OccupiedHeating Setpoint	Signed 16-bit Integer	*MinHeatSetpoint Limit – MaxHeatSetpoint Limit*	Read / Write	0x07d0 (20°C)	M
0x0013	UnoccupiedCooling Setpoint	Signed 16-bit Integer	*MinCoolSetpoint Limit – MaxCoolSetpoint Limit*	Read / Write	0x0a28 (26°C)	O
0x0014	UnoccupiedHeating Setpoint	Signed 16-bit Integer	*MinHeatSetpoint Limit – MaxHeatSetpoint Limit*	Read / Write	0x07d0 (20°C)	O
0x0015	MinHeatSetpoint Limit	Signed 16-bit Integer	0x954d – 0x7fff	Read / Write	0x02bc (7°C)	O
0x0016	MaxHeatSetpoint Limit	Signed 16-bit Integer	0x954d – 0x7fff	Read / Write	0x0bb8 (30°C)	O
0x0017	MinCoolSetpoint Limit	Signed 16-bit Integer	0x954d – 0x7fff	Read / Write	0x02bc (7°C)	O
0x0018	MaxCoolSetpoint Limit	Signed 16-bit Integer	0x954d – 0x7fff	Read / Write	0x0bb8 (30°C)	O

Table 6.12 Attributes of the Thermostat Settings Attribute Set (Continued)

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0019	MinSetpointDeadBand	Signed 8-bit Integer	0x0a – 0x19	Read / Write	0x19 (2.5°C)	O
0x001a	RemoteSensing	8-bit Bitmap	00000xxx	Read / Write	0	O
0x001b	ControlSequenceOfOperation	8-bit Enumeration	0x00 – 0x05	Read / Write	0x04	M
0x001c	SystemMode	8-bit Enumeration	0x00 – 0x02	Read / Write	0x02	M
0x001d	AlarmMask	8-bit bitmap	00000xxx	Read only	0	O

6.3.2.2.2.1 *LocalTemperatureCalibration* Attribute

The *LocalTemperatureCalibration* attribute specifies the offset that can be added/ subtracted to the actual displayed room temperature, in steps of 0.1°C. The range of this offset is –2.5 °C to +2.5 °C).

6.3.2.2.2.2 *OccupiedCoolingSetpoint* Attribute

The *OccupiedCoolingSetpoint* attribute is 16-bits in length and specifies the cooling mode setpoint when the room is occupied. It shall be set to a value in the range defined by the MinCoolSetpointLimit and MaxCoolSetpointLimit attributes. The value is calculated as described in the LocalTemperature attribute.

The OccupiedHeatingSetpoint attribute shall always be below the value specified in the OccupiedCoolingSetpoint by at least SetpointDeadband. If an attempt is made to set it such that this condition is violated, a default response command with the status code INVALID_VALUE (see 2.5.3) shall be returned. This shall apply to all attempts to set values of attributes which violate similar conditions.

If it is unknown if the room is occupied or not, this attribute shall be used as the cooling mode setpoint.

6.3.2.2.2.3 *OccupiedHeatingSetpoint* Attribute

The *OccupiedHeatingSetpoint* attribute is 16-bits in length and specifies the heating mode setpoint when the room is occupied. It shall be set to a value in the range defined by the *MinHeatSetpointLimit* and *MaxHeatSetpointLimit* attributes. The value is calculated as described in the *LocalTemperature* attribute. The

OccupiedCoolingSetpoint attribute shall always be above the value specified in the *OccupiedHeatingSetpoint* by at least *SetpointDeadband*.

If it is unknown if the room is occupied or not, this attribute shall be used as the cooling mode setpoint.

6.3.2.2.2.4 *UnoccupiedCoolingSetpoint* Attribute

The *UnoccupiedCoolingSetpoint* attribute is 16-bits in length and specifies the cooling mode setpoint when the room is unoccupied. It shall be set to a value in the range defined by the *MinCoolSetpointLimit* and *MaxCoolSetpointLimit* attributes. The value is calculated as described in the *LocalTemperature* attribute. The *UnoccupiedHeatingSetpoint* attribute shall always be below the value specified in the *UnoccupiedCoolingSetpoint* by at least *SetpointDeadband*.

If it is unknown if the room is occupied or not, this attribute shall not be used.

6.3.2.2.2.5 *UnoccupiedHeatingSetpoint* Attribute

The *UnoccupiedHeatingSetpoint* attribute is 16-bits in length and specifies the heating mode setpoint when the room is unoccupied. It shall be set to a value in the range defined by the *MinHeatSetpointLimit* and *MaxHeatSetpointLimit* attributes. The value is calculated as described in the *LocalTemperature* attribute. The *UnoccupiedCoolingSetpoint* attribute shall always be below the value specified in the *UnoccupiedHeatingSetpoint* by at least *SetpointDeadband*.

If it is unknown if the room is occupied or not, this attribute shall not be used.

6.3.2.2.2.6 *MinHeatSetpointLimit* Attribute

The *MinHeatSetpointLimit* attribute specifies the minimum level that the heating setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be greater than or equal to *AbsMinHeatSetpointLimit.* If this attribute is not present, it shall be taken as equal to *AbsMinHeatSetpointLimit.*

This attribute, and the following three attributes, allow the user to define setpoint limits more constrictive than the manufacturer imposed ones. Limiting users (e.g.in a commercial building) to such setpoint limits can help conserve power.

6.3.2.2.2.7 *MaxHeatSetpointLimit* Attribute

The *MaxHeatSetpointLimit* attribute specifies the maximum level that the heating setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be less than or equal to *AbsMaxHeatSetpointLimit.* If this attribute is not present, it shall be taken as equal to *AbsMaxHeatSetpointLimit.*

6.3.2.2.2.8 *MinCoolSetpointLimit* Attribute

The *MinCoolSetpointLimit* attribute specifies the minimum level that the cooling setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be greater than or equal to *AbsMinCoolSetpointLimit.* If this attribute is not present, it shall be taken as equal to *AbsMinCoolSetpointLimit.*

6.3.2.2.2.9 *MaxCoolSetpointLimit* Attribute

The *MaxCoolSetpointLimit* attribute specifies the maximum level that the cooling setpoint may be set to. The value is calculated as described in the LocalTemperature attribute. It must be less than or equal to *AbsMaxCoolSetpointLimit.* If this attribute is not present, it shall be taken as equal to *AbsMaxCoolSetpointLimit.*

6.3.2.2.2.10 *MinSetpointDeadBand* Attribute

The *MinSetpointDeadBand* attribute specifies the minimum difference between the Heat Setpoint and the Cool SetPoint, in steps of 0.1°C. Its range is 0x0a to 0x19 (1°C to 2.5°C).

6.3.2.2.2.11 *RemoteSensing* Attribute

The RemoteSensing attribute is an 8-bit bitmap that specifies whether the local temperature, outdoor temperature and occupancy are being sensed by internal sensors or remote networked sensors. The meanings of individual bits are detailed in Table 6.13.

Table 6.13 *RemoteSensing* Attribute Bit Values

Bit Number	Description
0	0 – local temperature sensed internally 1 – local temperature sensed remotely
1	0 – outdoor temperature sensed internally 1 – outdoor temperature sensed remotely
2	0 – occupancy sensed internally 1 – occupancy sensed remotely
3 - 7	Reserved

6.3.2.2.2.12 *ControlSequenceOfOperation* Attribute

The *ControlSequenceOfOperation* attribute specifies the overall operating environment of the thermostat, and thus the possible system modes that the thermostat can operate in. It shall be set to one of the non-reserved values in Table 6.14. (Note - it is not mandatory to support all values).

Table 6.14 ***ControlSequenceOfOperation*** **Attribute Values**

ControlSequenceOfOperation **Attribute Value**	**Description**	**Possible Values of *SystemMode***
0x00	Cooling Only	Heat and Emergency are not possible
0x01	Cooling With Reheat	Heat and Emergency are not possible
0x02	Heating Only	Cool and precooling (see 1.3.2) are not possible
0x03	Heating With Reheat	Cool and precooling are not possible
0x04	Cooling and Heating 4-pipes (see 1.3.2)	All modes are possible
0x05	Cooling and Heating 4-pipes with Reheat	All modes are possible
0x06 – 0xfe	Reserved	-

6.3.2.2.3 *SystemMode* Attribute

The *SystemMode* attribute specifies the current operating mode of the thermostat,. It shall be set to one of the non-reserved values in Table 6.15, as limited by Table 6.16. (Note - it is not mandatory to support all values).

Table 6.15 ***SystemMode*** **Attribute Values**

SystemMode **Attribute Value**	**Description**
0x00	Off
0x01	Auto
0x03	Cool
0x04	Heat
0x05	Emergency heating
0x06	Precooling (see 1.3.2)
0x07	Fan only
0x08 – 0xfe	Reserved

The interpretation of the Heat, Cool and Auto values of *SystemMode* is shown in Table 6.16.

Table 6.16 Interpretation of *SystemMode* Values

***SystemMode* Attribute Values**	**Temperature Below Heat Setpoint**	**Temperature Between Heat Setpoint and Cool Setpoint**	**Temperature Above Cool Setpoint**
Heat	Temperature below target	Temperature on target	Temperature on target
Cool	Temperature on target	Temperature on target	Temperature above target
Auto	Temperature below target	Temperature on target	Temperature above target

6.3.2.2.4 *AlarmMask* Attribute

The *AlarmMask* attribute specifies whether each of the alarms listed in Table 6.17 is enabled. When the bit number corresponding to the alarm code is set to 1, the alarm is enabled, else it is disabled. Bits not corresponding to a code in the table are reserved.

When the Alarms cluster is implemented on a device, and one of the alarm conditions included in this table occurs, an alarm notification is generated, with the alarm code field set as listed in the table.

Table 6.17 Alarm Codes

Alarm Code	**Alarm Condition**
0	Initialization failure. The device failed to complete initialization at power-up.
1	Hardware failure
2	Self-calibration failure

6.3.2.3 Commands Received

The command IDs for the Thermostat cluster are listed in Table 6.18.

Table 6.18 Command IDs for the Thermostat Cluster

Command Identifier Field Value	**Description**
0x00	Setpoint Raise/Lower
0x01 – 0xff	Reserved

6.3.2.3.1 Setpoint Raise/Lower Command

6.3.2.3.1.1 Payload Format

The Setpoint Raise/Lower command payload shall be formatted as illustrated in Figure 6.4.

Bits	8	8
Data Type	8-bit Enumeration	Signed 8-bit Integer
Field Name	Mode	Amount

Figure 6.4 Format of the Setpoint Raise/Lower Command Payload

6.3.2.3.1.2 Mode Field

The mode field shall be set to one of the non-reserved values in Table 6.19. It specifies which setpoint is to be configured. If it is set to auto, then both setpoints shall be adjusted.

Table 6.19 Mode field Values for the Setpoint Raise/Lower Command

Mode Field Value	Description
0x00	Heat (adjust Heat Setpoint)
0x01	Cool (adjust Cool Setpoint)
0x02	Both (adjust Heat Setpoint and Cool Setpoint)
0x03 – 0xff	Reserved

6.3.2.3.1.3 Amount Field

The amount field is a signed 8-bit integer that specifies the amount the setpoint(s) are to be a increased (or decreased) by, in steps of 0.1°C.

6.3.2.3.2 Effect on Receipt

The attributes for the indicated setpoint(s) shall be increased by the amount specified in the Amount field.

6.3.2.4 Commands Generated

No commands are generated by the server cluster.

6.3.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval and reportable change settings described in the ZCL Foundation specification (see 2.4.7) and whenever they change. The following attributes shall be reported:

- *LocalTemperature*
- *PICoolingDemand*
- *PIHeatingDemand*

Other attributes may optionally be reported.

6.3.3 Client

6.3.3.1 Dependencies

None.

6.3.3.2 Attributes

The Client cluster has no attributes.

6.3.3.3 Commands Received

The client receives no cluster specific commands.

6.3.3.4 Commands Generated

The client cluster generates the commands received by the server cluster, i.e. those detailed in 7.2.5, as required by the application.

6.4 Fan Control

6.4.1 Overview

This cluster specifies an interface to control the speed of a fan as part of a heating / cooling system.

6.4.2 Server

6.4.2.1 Dependencies

None.

6.4.2.2 Attributes

The Fan Control Status attribute set contains the attributes summarized in Table 6.20.

Table 6.20 Attributes of the Fan Control Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	FanMode	8-bit Enumeration	0x00 – 0x06	Read/ Write	0x05 (auto)	M
0x0001	FanModeSequence	8-bit Enumeration	0x00 – 0x04	Read/ Write	0x02	M

6.4.2.2.1 *FanMode* Attribute

The *FanMode* attribute is an 8-bit value that specifies the current speed of the fan. It shall be set to one of the non-reserved values in Table 6.21.

Table 6.21 *FanMode* Attribute Values

FanMode Attribute Value	Description
0x00	Off
0x01	Low
0x02	Medium
0x03	High
0x04	On
0x05	Auto (the fan speed is self-regulated)
0x06	Smart (when the heated/cooled space is occupied, the fan is always on)
0x07– 0xfe	Reserved

Note that for Smart mode, information must be available as to whether the heated/ cooled space is occupied. This may be accomplished by use of the Occupancy Sensing cluster (see 4.8).

6.4.2.2.2 *FanModeSequence* Attribute

The *FanModeSequence* attribute is an 8-bit value that specifies the possible fan speeds that the thermostat can set. It shall be set to one of the non-reserved values in Table 6.22. (Note:-. 'Smart' is not in this table, as this mode resolves to one of the other modes depending on occupancy).

Table 6.22 *FanSequenceOperation* Attribute Values

FanSequenceOfOperation **Attribute Value**	**Description**
0x00	Low/Med/High
0x01	Low/High
0x02	Low/Med/High/Auto
0x03	Low/High/Auto
0x04	On/Auto
0x05 – 0xfe	Reserved

6.4.2.3 Commands Received

No cluster specific commands are received by the server.

6.4.2.4 Commands Generated

No cluster specific commands are generated by the server.

6.4.3 Client

6.4.3.1 Dependencies

None.

6.4.3.2 Attributes

The Client cluster has no attributes.

6.4.3.3 Commands Received

No cluster specific commands are received by the server.

6.4.3.4 Commands Generated

No cluster specific commands are generated by the server.

6.5 Dehumidification Control

6.5.1 Overview

This cluster provides an interface to dehumidification functionality.

6.5.2 Server

6.5.2.1 Dependencies

None.

6.5.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute set for the dehumidification control cluster is listed in Table 6.23.

Table 6.23 Dehumidification Control Attribute Sets

Attribute Set Identifier	Description
0x000	Dehumidification Information
0x001	Dehumidification Settings
0x002 – 0xfff	Reserved

6.5.2.2.1 Dehumidification Information Attribute Set

The Dehumidification Information attribute set contains the attributes summarized in Table 6.24.

Table 6.24 Attributes of the Dehumidification Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	RelativeHumidity	Unsigned 8-bit Integer	0x00 – 0x64	Read only	-	O
0x0001	DehumidificationCooling	Unsigned 8-bit Integer	0 - *Dehumidification MaxCool*	Read only	-	M

6.5.2.2.1.1 *RelativeHumidity* Attribute

The *RelativeHumidity* attribute is an 8-bit value that represents the current relative humidity (in %) measured by a local or remote sensor. The valid range ix 0x00 – 0x64 (0% to 100%).

6.5.2.2.1.2 *DehumidificationCooling* Attribute

The *DehumidificationCooling* attribute is an 8-bit value that specifies the current dehumidification cooling output (in %). The valid range is 0 to *DehumidificationMaxCool.*

6.5.2.2.2 Dehumidification Settings Attribute Set

The Dehumidification Settings attribute set contains the attributes summarized in Table 6.25.

Table 6.25 Attributes of the Dehumidification Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	RHDehumidificationSetpoint	Unsigned 8-bit Integer	0x1E – 0x64	Read/ Write	0x32	M
0x0011	RelativeHumidityMode	8-bit Enumeration	0x00 – 0x01	Read/ Write	0x00	O
0x0012	DehumidificationLockout	8-bit Enumeration	0x00 – 0x01	Read/ Write	0x01	O
0x0013	DehumidificationHysteresis	Unsigned 8-bit Integer	0x02 – 0x14	Read/ Write	0x02	M
0x0014	DehumidificationMaxCool	Unsigned 8-bit Integer	0x14 – 0x64	Read/ Write	0x14	M
0x0015	RelativeHumidityDisplay	8-bit Enumeration	0x00 – 0x01	Read/ Write	0x00	O

6.5.2.2.2.1 *RHDehumidificationSetpoint* Attribute

The *RHDehumidificationSetpoint* attribute is an 8-bit value that represents the relative humidity (in %) at which dehumidification occurs. The valid range ix 0x1E – 0x64 (30% to 100%).

6.5.2.2.2.2 *RelativeHumidityMode* Attribute

The *RelativeHumidityMode* attribute is an 8-bit value that specifies how the *RelativeHumidity* value is being updated. It shall be set to one of the non-reserved values in Table 6.26.

Table 6.26 *RelativeHumidityMode* Attribute Values

RelativeHumidityMode Attribute Value	Description
0x00	*RelativeHumidity* measured locally
0x01	*RelativeHumidity* updated over the network
0x02 – 0xff	Reserved

6.5.2.2.2.3 *DehumidificationLockout* Attribute

The *DehumidificationLockout* attribute is an 8-bit value that specifies whether dehumidification is allowed or not. It shall be set to one of the non-reserved values in Table 6.27.

Table 6.27 *DehumidificationLockout* Attribute Values

Dehumidification **Attribute Value**	**Description**
0x00	Dehumidification is not allowed.
0x01	Dehumidification is allowed.
0x02 – 0xff	Reserved

6.5.2.2.2.4 *DehumidificationHysteresis* Attribute

The *DehumidificationHysteresis* attribute is an 8-bit value that specifies the hysteresis (in %) associated with *RelativeHumidity* value. The valid range ix 0x02 – 0x14 (2% to 20%).

6.5.2.2.2.5 *DehumidificationMaxCool* Attribute

The *DehumidificationMaxCool* attribute is an 8-bit value that specifies the maximum dehumidification cooling output (in %). The valid range ix 0x14 – 0x64 (20% to 100%).

6.5.2.2.2.6 *RelativeHumidityDisplay* Attribute

The *RelativeHumidityDisplay* attribute is an 8-bit value that specifies whether the *RelativeHumidity* value is displayed to the user or not. It shall be set to one of the non-reserved values in Table 6.28.

Table 6.28 *RelativeHumidityMode* Attribute Values

RelativeHumidityMode **Attribute Value**	**Description**
0x00	*RelativeHumidity* is not displayed
0x01	*RelativeHumidity* is displayed
0x02 – 0xff	Reserved

6.5.2.3 Commands Received

No commands are received by the server cluster except those to read / write attributes

6.5.2.4 Commands Generated

No commands are generated by the server cluster except responses to commands to read/write attributes, and attribute reports.

6.5.2.5 Attribute Reporting

This cluster shall support attribute reporting using the Report Attributes command and according to the minimum and maximum reporting interval settings described in the ZCL Foundation specification (see 2.4.7).

The following attribute shall be reported: *DehumidificationCooling*

This attribute shall also be reported whenever it changes (a minimum change is 1%).

Reports of this attribute may be used to control a remote dehumidifier device.

6.5.3 Client

6.5.3.1 Dependencies

None

6.5.3.2 Attributes

The client cluster has no attributes.

6.5.3.3 Commands Received

No commands are received by the server cluster except responses to commands to read/write attributes, and attribute reports.

6.5.3.4 Commands Generated

No commands are generated by the server cluster except those to read / write attributes, as required by the application

6.6 Thermostat User Interface Configuration Cluster

6.6.1 Overview

This cluster provides an interface to allow configuration of the user interface for a thermostat, or a thermostat controller device, that supports a keypad and LCD screen.

6.6.2 Server

6.6.2.1 Dependencies

None.

6.6.2.2 Attributes

The attributes of this cluster are summarized in Table 6.29.

Table 6.29 Attributes of the Thermostat User Interface Configuration Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	TemperatureDisplayMode	8-bit Enumeration	0x00 – 0x01	Read/ Write	0x00 (Celsius)	M
0x0001	KeypadLockout	8-bit Enumeration	0x00 – 0x05	Read/ Write	0x00 (no lockout)	M

6.6.2.2.1 *TemperatureDisplayMode* Attribute

The *TemperatureDisplayMode* attribute specifies the units of the temperature displayed on the thermostat screen. This attribute shall be set to one of the non-reserved values in Table 6.30.

Table 6.30 *DisplayMode* Attribute Values

***TemperatureDisplayMode* Attribute Value**	**Description**
0x00	Temperature in ^{o}C
0x01	Temperature in ^{o}F
0x02 – 0xff	Reserved

6.6.2.2.2 *KeypadLockout* Attribute

The *KeypadLockout* attribute specifies the level of functionality that is available to the user via the keypad. This attribute shall be set to one of the non-reserved values Table 6.31.

Table 6.31 *KeypadLockout* Attribute Values

***KeypadLockout* Attribute Value**	**Description**
0x00	No lockout
0x01	Level 1 lockout
0x02	Level 2 lockout
0x03	Level 3 lockout
0x04	Level 4 lockout
0x05	Level 5 lockout (least functionality available to the user)
0x06– 0xff	Reserved

The interpretation of the various levels is device dependent.

6.6.2.3 Commands Received

No commands are received by the server except those to read and write the attributes of the server.

6.6.2.4 Commands Generated

No commands are generated by the server except responses to commands to read and write the attributes of the server.

6.6.3 Client

6.6.3.1 Dependencies

None.

6.6.3.2 Attributes

The Client cluster has no attributes.

6.6.3.3 Commands Received

No commands are received by the server except responses to commands to read and write the attributes of the server.

6.6.3.4 Commands Generated

No commands are generated by the client except those to read and write the attributes of the server, as required by the application.

CHAPTER 7

CLOSURES SPECIFICATION

7.1 General Description

7.1.1 Introduction

The clusters specified in this document are for use typically in ZigBee applications involving closures (e.g. shades, windows doors), but may be used in any application domain.

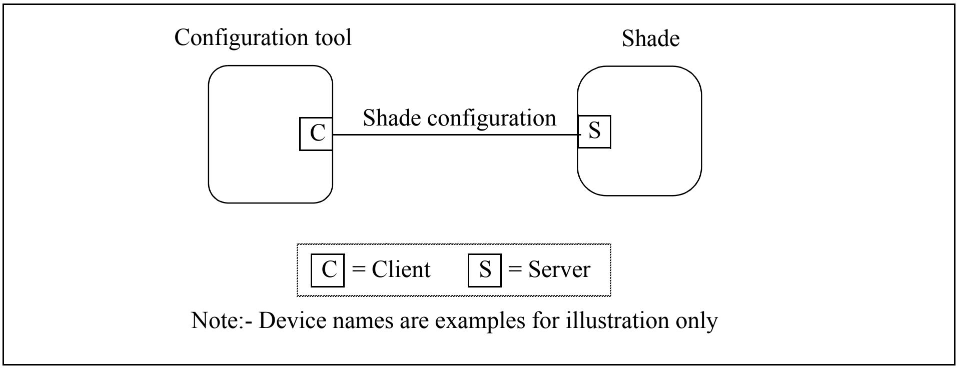
7.1.2 Cluster List

This section lists the clusters specified in this document, and gives examples of typical usage for the purpose of clarification.

The clusters defined in this document are listed in Table 7.1.

Table 7.1 Clusters Specified in the Closures Functional Domain

Cluster Name	Description
Shade Configuration	Attributes and commands for configuring a shade



Note:- Device names are examples for illustration only

Figure 7.1 Typical Usage of the Closures Clusters

7.2 Shade Configuration Cluster

7.2.1 Overview

This cluster provides an interface for reading information about a shade, and configuring its open and closed limits.

7.2.2 Server

7.2.2.1 Dependencies

None

7.2.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set

and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 7.2.

Table 7.2 Shade Configuration Attribute Sets

Attribute Set Identifier	Description
0x000	Shade information
0x001	Shade settings
0x002 – 0xfff	Reserved

7.2.2.2.1 Shade Information Attribute Set

The Shade Information attribute set contains the attributes summarized in Table 7.3.

Table 7.3 Attributes of the Shade Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	PhysicalClosedLimit	Unsigned 16-bit integer	0x0001 – 0xfffe	Read only	-	O
0x0001	MotorStepSize	Unsigned 8-bit integer	0x00 – 0xfe	Read only	-	O
0x0002	Status	8-bit Bitmap	0000 xxxx	Read / write	0000 0000	M

7.2.2.2.1.1 *PhysicalClosedLimit* Attribute

The *PhysicalClosedLimit* attribute indicates the most closed (numerically lowest) position that the shade can physically move to. This position is measured in terms of steps of the motor, taking the physical most open position of the shade as zero.

This attribute is for installation informational purposes only.

The value 0xffff indicates an invalid or unknown *PhysicalClosedLimit* .

7.2.2.2.1.2 *MotorStepSize* Attribute

The *MotorStepSize* attribute indicates the angle the shade motor moves for one step, measured in 1/10ths of a degree.

This attribute is for installation informational purposes only.

The value 0xff indicates an invalid or unknown step size.

7.2.2.2.1.3 *Status* Attribute

The *Status* attribute indicates the status of a number of shade functions, as shown in Table 7.4 Writing a value to this attribute only affects those bits with Read / Write access.

Table 7.4 Bit Values for the *Status* Attribute

Status Attribute Bit Number	Meaning	Access
0	Shade operational 0 = no 1 = yes	Read only
1	Shade adjusting 0 = no 1 = yes	Read only
2	Shade direction 0 = closing 1 = opening	Read only
3	Direction corresponding to forward direction of motor 0 = closing 1 = opening	Read / write
4 – 7	Reserved	-

7.2.2.2.2 Shade Settings Attribute Set

The Shade Settings attribute set contains the attributes summarized in Table 7.5.

Table 7.5 Attributes of the Shade Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	ClosedLimit	Unsigned 16-bit integer	0x0001 – 0xfffe	Read/write	0x0001	M
0x0011	Mode	8-bit Enumeration	0x00 – 0xfe	Read/write	0x00	M

7.2.2.2.2.1 *ClosedLimit* Attribute

The *ClosedLimit* attribute indicates the most closed position that the shade can move to. This position is measured in terms of steps of the motor, taking the physical most open position of the shade as zero. This attribute is set either by directly writing it, or by the following method.

When the Mode attribute is set to Configure, the shade is opening, and either the shade is stopped or it reaches its physical most open limit (if there is one – the motor may continue to turn at the top), the zero point for the motor-step measurement system is set to the current position of the shade.

When the Mode attribute is set to Configure, the shade is closing, and either the shade is stopped or it reaches its physical closed limit*,* the *ClosedLimit* attribute is set to the current position of the shade, relative to the zero point set as described above.

7.2.2.2.2.2 Mode Attribute

The *Mode* attribute indicates the current operating mode of the shade, as shown in Table 7.6.

The value 0xff indicates an invalid or unknown mode.

Table 7.6 Values of the *Mode* Attribute

***Mode* Attribute Value**	**Meaning**
0x00	Normal
0x01	Configure
0x02 – 0xfe	Reserved

In configure mode, the *ClosedLimit* attribute may be set as described above.

7.2.2.3 Commands Received

No cluster specific commands are received by the server.

7.2.2.4 Commands Generated

No cluster specific commands are generated by the server.

7.2.3 Client

7.2.3.1 Dependencies

None

7.2.3.2 Attributes

The client has no attributes.

7.2.3.3 Commands Received

No cluster specific commands are received by the client.

7.2.3.4 Commands Generated

No cluster specific commands are generated by the client.

CHAPTER 8

SECURITY AND SAFETY SPECIFICATION

8.1 General Description

8.1.1 Introduction

The clusters specified in this document are for use in ZigBee security and safety related applications.

The clusters currently defined are those that are used by wireless Intruder Alarm Systems (IAS). Intruder Alarm systems include functions for the detection of intruders and/or triggering, processing of information, notification of alarms and the means to operate the IAS.

Functions additional to those may be included in IAS providing they do not influence the correct operation of the mandatory functions. Components of other applications may be combined or integrated with a IAS, providing the performance of the IAS components is not adversely influenced.

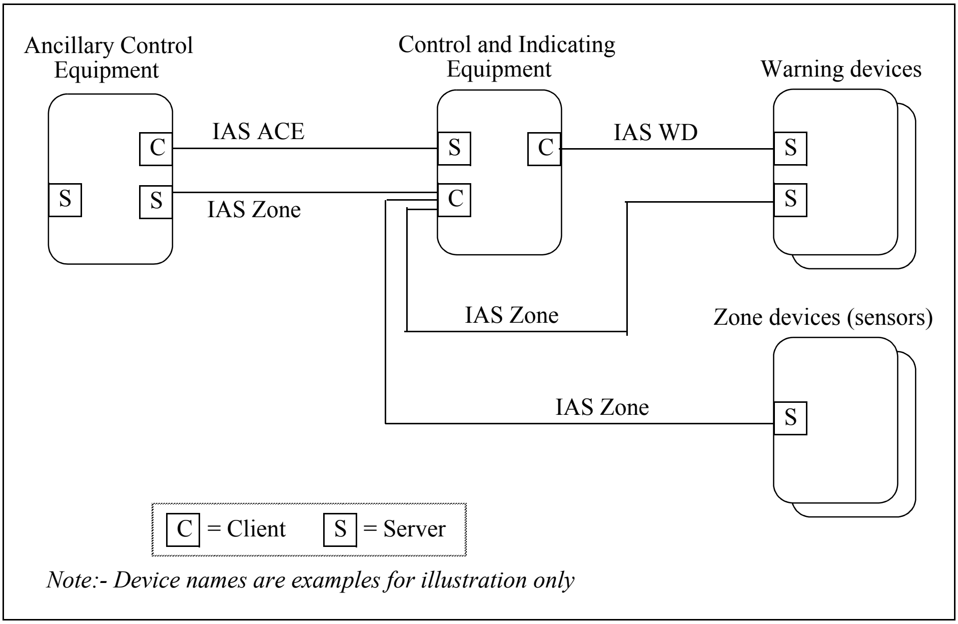
8.1.2 Cluster List

This section lists the clusters specified in this document, and gives examples of typical usage for the purpose of clarification.

The clusters defined in this document are listed in Table 8.1.

Table 8.1 Clusters Specified in the Security and Safety Functional Domain

Cluster Name	Description
IAS Zone	Attributes and commands for IAS security zone devices.
IAS ACE	Attributes and commands for IAS Ancillary Control Equipment.
IAS WD	Attributes and commands for IAS Warning Devices



Note:- Device names are examples for illustration only

Figure 8.1 Typical Usage of the IAS Clusters

8.2 IAS Zone Cluster

8.2.1 Overview

The IAS Zone cluster defines an interface to the functionality of an IAS security zone device. IAS Zone supports up to two alarm types per zone, low battery reports and supervision of the IAS network.

8.2.2 Server

8.2.2.1 Dependencies

None.

8.2.2.2 Attributes

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such that the most significant three nibbles specify the attribute set

and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in Table 8.2.

Table 8.2 Attribute Sets for the IAS Zone Cluster

Attribute Set Identifier	Description
0x000	Zone information
0x001	Zone settings
0x002 – 0xfff	Reserved

8.2.2.2.1 Zone Information Attribute Set

The Zone Information attribute set contains the attributes summarized in Table 8.3.

Table 8.3 Attributes of the Zone Information Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	ZoneState	8-bit Enumeration	All	Read only	0x00	M
0x0001	ZoneType	16-bit Enumeration	All	Read only	-	M
0x0002	ZoneStatus	16-bit bitmap	All	Read only	0x00	M

8.2.2.2.1.1 *ZoneState* Attribute

The *ZoneState* attribute contains the values summarized in Table 8.4.

Table 8.4 Values of the *ZoneState* Attribute

ZoneState Value	Meaning
0x00	Not enrolled
0x01	Enrolled (the server will react to Zone State Change Notification commands from the client)
0x02-0xff	Reserved

8.2.2.2.1.2 *ZoneType* Attribute

The *ZoneType* attribute values are summarized in Table 8.5. The Zone Type dictates the meaning of Alarm1 and Alarm2 bits of the *ZoneStatus* attribute, as also indicated in this table.

Table 8.5 Values of the *ZoneType* Attribute

ZoneType attribute value	Zone Type	Alarm1	Alarm2
0x0000	Standard CIE	System Alarm	-
0x000d	Motion sensor	Intrusion indication	Presence indication
0x0015	Contact switch	1st portal Open-Close	2nd portal Open-Close
0x0028	Fire sensor	Fire indication	-
0x002a	Water sensor	Water overflow indication	-
0x002b	Gas sensor	CO indication	Cooking indication
0x002c	Personal emergency device	Fall / Concussion	Emergency button
0x002d	Vibration / Movement sensor	Movement indication	Vibration
0x010f	Remote Control	Panic	Emergency
0x0115	Key fob	Panic	Emergency
0x021d	Keypad	Panic	Emergency
0x0225	Standard Warning Device (see [B2] part 4)	-	-
Other values < 0x7fff	Reserved	-	-
0x8000-0xfffe	Reserved for manufacturer specific types	-	-
0xffff	Invalid Zone Type	-	-

8.2.2.2.1.3 *ZoneStatus* Attribute

The *ZoneStatus* attribute is a bit map. The meaning of each bit is summarized in Table 8.6.

Table 8.6 Values of the *ZoneStatus* Attribute

ZoneStatus **Attribute Bit Number**	**Meaning**	**Values**
0	Alarm1	1 – opened or alarmed 0 – closed or not alarmed
1	Alarm2	1 – opened or alarmed 0 – closed or not alarmed
2	Tamper	1 – Tampered 0 – Not tampered
3	Battery	1 – Low battery 0 – Battery OK
4	Supervision reports (Note 1)	1 – Reports 0 – Does not report
5	Restore reports (Note 2)	1 – Reports restore 0 – Does not report restore
6	Trouble	1 – Trouble/Failure 0 – OK
7	AC (mains)	1 – AC/Mains fault 0 – AC/Mains OK
8-15	Reserved	-

Note 1: This bit indicates whether the Zone issues periodic Zone Status Change Notification commands. The CIE device may use these periodic reports as an indication that a zone is operational. Zones that do not implement the periodic reporting are required to set this bit to zero (the CIE will know not to interpret the lack of reports as a problem).

Note2: This bit indicates whether or not a Zone Status Change Notification command will be sent to indicate that an alarm is no longer present. Some Zones do not have the ability to detect that alarm condition is no longer present, they only can tell that an alarm has occurred. These Zones must set the "Restore" bit to zero, indicating to the CIE not to look for alarm-restore notifications.

8.2.2.2.2 Zone Settings Attribute Set

The Zone settings attribute set contains the attributes summarized in Table 8.7.

Table 8.7 Attributes of the Zone Settings Attribute Set

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0010	IAS_CIE_Address	IEEE address	Valid 64bit IEEE address	Read/ Write	-	M

8.2.2.2.2.1 *IAS_CIE_Address* Attribute

The *IAS_CIE_Address* attribute specifies the address that commands generated by the server shall be sent to. All commands received by the server must also come from this address.

It is up to the zone's specific implementation to permit or deny change (write) of this attribute at specific times. Also, it is up to the zone's specific implementation to implement some auto-detect for the CIE (example: by requesting the ZigBee cluster discovery service to locate a Zone Server cluster.) or require the intervention of a CT in order to configure this attribute during installation.

8.2.2.3 Commands Received

The command IDs received by the IAS Zone server cluster are listed in Table 8.8.

Table 8.8 Received Command IDs for the IAS Zone Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Zone Enroll Response	M
0x01 – 0xff	Reserved	

8.2.2.3.1 Zone Enroll Response Command

8.2.2.3.1.1 Payload Format

The Zone Enroll Response command payload shall be formatted as illustrated in Figure 8.2.

Bits	8	8
Data Type	8-bit Enumeration	Unsigned 8-bit Integer
Field Name	Enroll response code	Zone ID

Figure 8.2 Format of the Zone Enroll Response Command Payload

The permitted values of the Enroll Response Code are shown in Table 8.9.

Table 8.9 Values of the Enroll Response Code

Code	Meaning	Details
0x00	Success	Success
0x01	Not supported	This specific Zone type is not known to the CIE and is not supported.
0x02	No enroll permit	CIE does not permit new zones to enroll at this time.
0x03	Too many zones	CIE reached its limit of number of enrolled zones
0x04-0xfe	Reserved	-

The Zone ID field is the index into the zone table of the CIE (Table 8.11). This field is only relevant if the response code is success.

8.2.2.3.1.2 Effect on Receipt

On receipt, the device embodying the Zone client is notified that it is now enrolled as an active alarm device

The device embodying the Zone client must authenticate received messages by checking the address of their sender against IAS_CIE_Address. This is to ensure that only messages from the correct CIE are accepted.

8.2.2.4 Commands Generated

The generated command IDs for the IAS Zone server cluster are listed in Table 8.10.

Table 8.10 Generated Command IDs for the IAS Zone Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Zone Status Change Notification	M
0x01	Zone Enroll Request	M
0x02 – 0xff	Reserved	

8.2.2.4.1 Zone Status Change Notification Command

8.2.2.4.1.1 Payload Format

The Zone Status Change Notification command payload shall be formatted as illustrated in Figure 8.3.

Bits	16	8
Data Type	16-bit Enumeration	8-bit Enumeration
Field Name	Zone Status	Extended Status

Figure 8.3 Format of the Zone Status Change Notification Command Payload

The Zone Status field shall be the current value of the *ZoneStatus* attribute.

The Extended Status field is reserved for additional status information and shall be set to zero.

8.2.2.4.1.2 When Generated

The Zone Status Change Notification command is generated when a change takes place in one or more bits of the *ZoneStatus* attribute.

8.2.2.4.2 Zone Enroll Request Command

8.2.2.4.2.1 Payload Format

The Zone Enroll Request command payload shall be formatted as illustrated in Figure 8.4.

Bits	16	16
Data Type	16-bit Enumeration	Unsigned 16-bit Integer
Field Name	Zone Type	Manufacturer Code

Figure 8.4 Format of the Zone Enroll Request Command Payload

The Zone Type field shall be the current value of the *ZoneType* attribute.

The Manufacturer Code field shall be the manufacturer code as held in the node descriptor for the device. Manufacturer Codes are allocated by the ZigBee Alliance.

8.2.2.4.2.2 When Generated

The Zone Enroll Request command is generated when a device embodying the Zone client cluster wishes to be enrolled as an active alarm device. It must do this immediately it has joined the network (during commissioning).

8.2.3 Client

8.2.3.1 Dependencies

None.

8.2.3.2 Attributes

No attributes are currently defined for this cluster.

8.2.3.3 Commands Received

The client receives the cluster specific commands detailed in 8.2.2.4.

8.2.3.4 Commands Generated

The client generates the cluster specific commands detailed in 8.2.2.3, as required by the application.

8.3 IAS ACE Cluster

8.3.1 Overview

The IAS ACE cluster defines an interface to the functionality of any Ancillary Control Equipment of the IAS system. Using this cluster, a ZigBee enabled ACE device can access a IAS CIE device and manipulate the IAS system, on behalf of a level-2 user (see [B2]).

The client is usually implemented by the IAS ACE device. It allows the IAS ACE device to control the IAS CIE device, which typically implements the server side.

8.3.2 Server

8.3.2.1 Dependencies

None.

8.3.2.2 Attributes

No attributes are currently defined for this cluster.

8.3.2.3 Zone Table

The Zone Table is used to store information for each Zone enrolled by the CIE. The maximum number of entries in the table is 256.

The format of a group table entry is illustrated in Table 8.11.

Table 8.11 Format of the Zone Table

Field	Type	Valid Range	Description
Zone ID	Unsigned 8-bit integer	0x00 – 0xfe	The unique identifier of the zone
Zone Type	16-bit Enumeration	0x0000 – 0xfffe	See Table 8.5.
Zone Address	IEEE Address	Valid 64bit IEEE address	Device address

The Zone ID is a unique reference number allocated by the CIE at zone enrollment time.

The Zone ID is used by IAS devices to reference specific zones when communicating with the CIE. The Zone ID of each zone stays fixed until that zone is un-enrolled.

8.3.2.4 Commands Received

The received command IDs for the IAS ACE server cluster are listed in Table 8.12

Table 8.12 Received Command IDs for the IAS ACE Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Arm	M
0x01	Bypass	M
0x02	Emergency	M
0x03	Fire	M
0x04	Panic	M
0x05	Get Zone ID Map	M
0x06	Get Zone Information	M
0x07 – 0xff	Reserved	-

8.3.2.4.1 Arm Command

8.3.2.4.1.1 Payload Format

The Arm command payload shall be formatted as illustrated in Figure 8.5.

Bits	8
Data Type	8-bit Enumeration
Field Name	Arm Mode

Figure 8.5 Format of the Arm Command Payload

8.3.2.4.1.2 Arm Mode Field

The Arm Mode field shall have one of the values shown in Table 8.13.

Table 8.13 Arm Mode Field Values

Arm Mode Field Value	Meaning
0x00	Disarm
0x01	Arm Day/Home Zones Only
0x02	Arm Night/Sleep Zones Only
0x03	Arm All Zones
0x08-0xff	Reserved

8.3.2.4.1.3 Effect on Receipt

On receipt of this command, the receiving device sets its arm mode according to the value of the Arm Mode field, as detailed in Table 8.13. It is not guaranteed that an Arm command will succeed. Based on the current state of the IAS CIE, and its related devices, the command can be rejected. The device shall generate an Arm Response command (see 8.3.2.5.1) to indicate the resulting armed state.

8.3.2.4.2 Bypass Command

8.3.2.4.2.1 Payload Format

The Bypass command payload shall be formatted as illustrated in Figure 8.6.

Bits	8	8	- - - - -	8
Data Type	Unsigned 8-bit Integer	Unsigned 8-bit Integer	- - - - -	Unsigned 8-bit Integer
Field Name	Number of Zones	Zone ID	- - - - -	Zone ID

Figure 8.6 Format of the Bypass Command Payload

8.3.2.4.2.2 Number of Zones Parameter

This is the number of Zone IDs included in the payload.

8.3.2.4.2.3 Zone ID Parameter

Zone ID is the index of the Zone in the CIE's zone table (Table 8.11).

8.3.2.4.3 Emergency, Fire and Panic Commands

These commands indicate the emergency situations inherent in their names. They have no payload.

8.3.2.4.4 Get Zone ID Map Command

8.3.2.4.4.1 Payload Format

This command has no payload.

8.3.2.4.4.2 Effect on Receipt

On receipt of this command, the device shall generate a Get Zone ID Map Response command. See 7.2.5.2

8.3.2.4.5 Get Zone Information Command

8.3.2.4.5.1 Payload Format

The Get Zone Information command payload shall be formatted as illustrated in Figure 8.7.

Bits	8
Data Type	Unsigned 8-bit Integer
Field Name	Zone ID

Figure 8.7 Format of the Get Zone Information Command Payload

8.3.2.4.5.2 Effect on Receipt

On receipt of this command, the device shall generate a Get Zone Information Response command. See 7.2.5.3.

8.3.2.5 Commands Generated

The generated command IDs for the IAS ACE server cluster are listed in Table 8.14.

Table 8.14 Generated Command IDs for the IAS ACE Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Arm Response	M
0x01	Get Zone ID Map Response	M
0x02	Get Zone Information Response	M
0x03 – 0xff	Reserved	

8.3.2.5.1 Arm Response Command

8.3.2.5.1.1 Payload Format

The Arm Response command payload shall be formatted as illustrated in Figure 8.8.

Bits	8
Data Type	8-bit Enumeration
Field Name	Arm Notification

Figure 8.8 Format of the Arm Response Command Payload

8.3.2.5.1.2 Arm Notification Field

The Arm Notification field shall have one of the values shown in Table 8.15.

Table 8.15 Arm Notification Values

Arm Mode Attribute Value	Meaning
0x00	All Zones Disarmed
0x01	Only Day/Home Zones Armed
0x02	Only Night/Sleep Zones Armed
0x03	All Zones Armed
0x04 – 0xfe	Reserved

8.3.2.5.2 Get Zone ID Map Response Command

8.3.2.5.2.1 Payload Format

The Get Zone ID Map Response command payload shall be formatted as illustrated in Figure 8.9.

Bits	16	. . . .	16
Data Type	16-bit bitmap	. . . .	16-bit bitmap
Field Name	Zone ID Map section 0	. . . .	Zone ID Map section 15

Figure 8.9 Format of the Get Zone ID Map Response Command Payload

The 16 fields of the payload indicate whether each of the Zone IDs from 0 to 0xff is allocated or not. If bit n of Zone ID Map section N is set to 1, then Zone ID (16 x N + n) is allocated, else it is not allocated.

8.3.2.5.3 Get Zone Information Response Command

8.3.2.5.3.1 Payload Format

The Get Zone Information Response command payload shall be formatted as illustrated in Figure 8.10.

Bits	8	16	64
Data Type	Unsigned 8-bit Integer	16-bit Enumeration	IEEE address
Field Name	Zone ID	Zone Type	IEEE address

Figure 8.10 Format of the Get Zone Information Response Command Payload

The fields of the payload are equal to the fields of the Group Table entry corresponding to the ZoneID field of the Get Zone Information command to which this command is a response.

If the Zone ID is unallocated, this shall be indicated by setting the Zone Type and IEEE Address fields to 0xffff (see Table 8.5) and 0xffffffffffffffff respectively.

8.3.3 Client

8.3.3.1 Dependencies

None.

8.3.3.2 Attributes

No attributes are currently defined for this cluster.

8.3.3.3 Commands Received

No cluster specific commands are received by the server.

8.3.3.4 Commands Generated

The client cluster generates the commands detailed in 8.3.2.4, as required by the application.

8.4 IAS WD Cluster

8.4.1 Overview

The IAS WD cluster provides an interface to the functionality of any Warning Device equipment of the IAS system. Using this cluster, a ZigBee enabled CIE device can access a ZigBee enabled IAS WD device and issue alarm warning indications (siren, strobe lighting, etc.) when a system alarm condition is detected (according to [B2]).

8.4.2 Server

8.4.2.1 Dependencies

None.

8.4.2.2 Attributes

The attributes defined for the server cluster are detailed in Table 8.16.

Table 8.16 Attributes of the IAS WD (Server) Cluster

Identifier	Name	Type	Range	Access	Default	Mandatory / Optional
0x0000	MaxDuration	Unsigned 16-bit integer	0x0000 – 0fffe	Read/Write	240	M
0x0001-0xffff	Reserved	-	-	-	-	-

8.4.2.2.1 *MaxDuration* Attribute

The *MaxDuration* attribute specifies the maximum time in seconds that the siren will sound continuously, regardless of start/stop commands.

8.4.2.3 Commands Received

The received command IDs are listed in Table 8.17.

Table 8.17 Received Command IDs for the IAS WD Server Cluster

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Start warning	M
0x01	Squawk	M
0x02 – 0xff	Reserved	

8.4.2.3.1 Start Warning Command

This command starts the WD operation. The WD alerts the surrounding area by audible (siren) and visual (strobe) signals.

A Start Warning command shall always terminate the effect of any previous command that is still current.

8.4.2.3.1.1 Payload Format

The Start Warning command payload shall be formatted as illustrated in Figure 8.11.

Bits	4	2	2	16
Data Type	8-bit Data			Unsigned 16-bit Integer
Field Name	Warning mode	Strobe	Reserved	Warning duration

Figure 8.11 Format of the Start Siren Command Payload

The Warning mode and Strobe subfields are concatenated together to a single 8-bit Bitmap field. The groups of bits these subfields occupy are used as follows.

8.4.2.3.1.2 Warning Mode Field

The Warning Mode field is used as an 4-bit enumeration, can have one of the values set in Table 8.18. The exact behavior of the WD device in each mode is according to the relevant security standards.

Table 8.18 Warning Modes

Warning Mode	Meaning
0	Stop (no warning)
1	Burglar
2	Fire
3	Emergency
4-15	Reserved

8.4.2.3.1.3 Strobe Field

The Strobe field is used as a 2-bit enumeration, and determines if the visual indication is required in addition to the audible siren, as indicated in Table 8.19. If the strobe field is "1" and the Warning Mode is "0" ("Stop") then only the strobe is activated.

Table 8.19 Values of the Strobe Field

Value	Meaning
0	No strobe
1	Use strobe in parallel to warning
2-3	Reserved

8.4.2.3.1.4 Warning Duration Field

Requested duration of warning, in seconds. If both Strobe and Warning Mode are "0" this field shall be ignored.

8.4.2.3.2 Squawk Command

This command uses the WD capabilities to emit a quick audible/visible pulse called a "squawk". The squawk command has no effect if the WD is currently active (warning in progress).

8.4.2.3.2.1 Payload Format

The Squawk command payload shall be formatted as illustrated in Figure 8.12.

Bits	4	1	1	2
Data Type	8-bit Data			
Field Name	Squawk mode	Strobe	Reserved	Squawk level

Figure 8.12 Format of the Start Siren Command payload

8.4.2.3.2.2 Squawk Mode Field

The Squawk Mode field is used as a 4-bit enumeration, and can have one of the values shown in Table 8.20. The exact operation of each mode (how the WD "squawks") is implementation specific.

Table 8.20 Squawk Mode Field

Warning Mode	Meaning
0	Notification sound for "System is armed"
1	Notification sound for "System is disarmed"
2-15	Reserved

8.4.2.3.2.3 Strobe Field

The strobe field is used as a boolean, and determines if the visual indication is also required in addition to the audible squawk., as shown in Table 8.21.

Table 8.21 Strobe Bit

Value	Meaning
0	No strobe
1	Use strobe blink in parallel to squawk

8.4.2.3.2.4 Squawk Level Field

The squawk level field is used as a 2-bit enumeration, and determines the intensity of audible squawk sound as shown in Table 8.22.

Table 8.22 Squawk Level Field Values

Value	Meaning
0	Low level sound
1	Medium level sound
2	High level sound
3	Very High level sound

8.4.2.4 Commands Generated

No cluster specific commands are generated by the server cluster.

8.4.3 Client

The client side is implemented by the CIE. The CIE is a client of the warning service provided by this cluster. Usually a WD would implement an IAS WD cluster server and an IAS Zone cluster server.

8.4.3.1 Dependencies

None.

8.4.3.2 Attributes

No attributes are currently defined for the client cluster.

8.4.3.3 Received Commands

The client receives no cluster specific commands.

8.4.3.4 Commands Generated

The client cluster generates the cluster specific commands detailed in 8.4.2.3, as required by the application.

This page left intentionally blank