Software Design Specification

Z-Wave Command Class Specification

Document No.:	SDS11060
Version:	14
Description:	This document describes the Command Classes and associated Commands used by Z-Wave enabled products ensuring that compliant products will be interoperable.
Written By:	JFR;JRM;ABR
Date:	2012-05-02
Reviewed By:	CHL;JRM;ABR;JFR;NTJ;BBR;DKING
Restrictions:	Partners Only

Approved by:				
Date	CET	Initials	Name	Justification
2012-05-02	08:50:59	NTJ	Niels Thybo Johansen	

CONFIDENTIAL

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
1	20070920	JFR	ALL	All Command Classes from SDS10242-12 added.
1	20070929	JFR	Section 4.36	Latitude and longitude only 16 bits in Geographic Command Class.
1	20071213	JFR	Appendix A	Manufacturer ID table updated.
1	20071227	JFR	Section 4.10	Association Command Configuration Command Class added.
1	20071227	JFR	Section 4.2 Section 4.5	Alarm Sensor Command Class added. Alarm Silence Command Class added.
1	20071227	JFR	Section 4.93	Sensor Configuration Command Class added.
1	20071227	JFR	Section 4.105	Thermostat Setback Command Class added.
2	20080121	JFR		Composite Command Class discontinued Association Command Class version 2 discontinued
3	20080315	JFR	Section 4.26	Configuration Command Class version 2 added
3	20080812	JFR	Section 4.93	Security Command Class added.
3	20080812	JFR	Section 4.60 Section 4.59 Section 4.10	Multi Channel Command Class, version 2 added as an extension to Multi Instance Command Class, version 1. Multi Channel Association Command Class, version 2 added as an extension to Multi Instance Association Command Class, version 1. Association Command Class, version 2 added.
3	20080814	JFR	Section 4.61	Multilevel Sensor Command Class extended to version 3.
3	20080814	JFR	Section 4.15	Basic Tariff Information Command Class added.
4	20080818	JFR	Section 4.10	Added clarification A) in Association Command Configuration Command Class.
4	20080826	JRM	Section 4.93	Secure Network Inclusion timeouts clarified.
4	20080827	JFR	Section 4.36	Firmware Update Meta Data Command Class, version 2 added.
4	20081118	JFR	Section 4.67	Clarified announcement of No Operation Command Class in Node Information Frame (NIF)
5	20081202	JFR	Section 4.107 & 4.108	Time Command Class is split into 2 versions; a simple one for transferring time and date info between Z-Wave devices (version 1) and a more complex one including time zones and daylight saving (version 2). Notice discontinuation of the old Time Command Class version 1.
5	20081202	JFR	Section 4.102 Section 4.106	Thermostat Mode Command Class version 2 added Thermostat Setpoint Command Class version 2 added
5	20090129	JFR	Section 4.20 & 4.65	Warning against using Toggle Switch Command Classes in new devices
5	20090212	JFR	Section 4.113.4 Appendix A	Clarification when using Wake Up Notification Command broadcast Manufacturer ID's updated
6	20090218	JFR	Section 4.50 Section 4.63.1	Meter Command Class version 2 added Text "100…254 (0x63…0xFE)" changed to "100…254 (0x64…0xFE)"
6	20090320	JRM	Section 4.93.1.3 Section 4.93.1 Section 4.93.2 Section 4.93.2.2 Section 4.93.2.1 Section 4.93.1 Section 4.93.2.1 Section 4.93.1 Section 4.93.2.1 Section 4.93.2.1.2	Added security frame flow diagram to illustrate sequencing and updated description. Sequence counter text updated. Network streaming made mandatory Updated to only support security 0 using low power to improve usability, but security scheme is extendable to offer stronger key exchange at a later stage. Notice: New security 0 value! Updated to only support security 0 using normal power to improve usability, but security scheme is extendable to offer stronger key exchange at a later stage. Nonce Timers described in more detail, and changed from 3 seconds to a defined variable Added / described inclusion timer in more detail. Timers MUST be started as soon as message is sent, not after Acknowledge is received. State diagram for inclusion timers added
7	20090404	JFR	Section 4.40 Section 4.41	HRV Status Command Class added HRV Control Command Class added
7	20090429	JFR	Section 4.65	Multilevel Switch Command Class version 3 added
8	20090731	JFR	Section 4.9.3 Section 4.6.1	Clarified Association Report Command content Clarified All Switch Set Command behavior when used in connection with Multilevel Switch Command Class

CONFIDENTIAL

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
8	20090820	JFR	Section 4.76.5.1 Section 4.95.1	Protection Exclusive Control Set Command functionality clarified. Simple AV Control Command Class version 3 added. New AV commands added to Simple AV Control Set Command.
8	20090827	JFR	Section 4.2 Section 4.85 Section 4.33	Alarm Command Class, Version 2 added Schedule Entry Lock Command Class added Door Lock Logging Command Class added
8	20090829	JFR	 Section 4.29 Section 4.30 Section 4.51 Section 4.54 Section 4.55 Section 4.72 Section 4.73 Section 4.78 Section 4.79 Section 4.96 Section 4.97	Following Advanced Energy Control Command Classes added DCP List Configuration DCP List Monitoring Meter Table Configuration Meter Table Monitor Meter Table Push Configuration Prepayment Prepayment Encapsulation Rate Table Configuration Rate Table Monitoring Tariff Table Configuration Tariff Table Monitor
8	20090905	JFR	Section 3.3	Common command class variables added
8	20090909	JFR	Section 4.59.4.1 Section 4.60	Added Multi Channel Association Remove Command examples Clarification regarding support of Multi Channel Command Class
9	20091218	ABR	Section 4	Z/IP Tunneling and Advanced Z/IP Command Classes discontinued.
9	20090414	JFR	Section 4.50 Section 4.61 Appendix A	Meter Command Class version 3 added (addition of scales) Multilevel Sensor Command Class version 4 added (new type) Manufacturer ID's updated
10	20090505	JFR	Section 4.112.2	Protocol versions updated in Version Command Class
10	20100610	JFR	Section 4.95.1	Simple AV Control Command Class version 4 added. New AV commands added to Simple AV Control Set Command. Keep Alive frequency change from 500 ms to 100-200 ms.
11	20100913	JFR	Section 4.102.5	Corrected and clarified example.
11	20101206	JRM	Section 4.93.2 Section 4.93.2.1	Clarified that the same network key is used by all nodes for the currently available security scheme 0.
11	20101212	JFR	Section 4.99	Thermostat Fan Mode Command Class version 2 added.
11	20110120	JFR	Section 4.54.8 Section 4.54.11 Section 4.54.15 Section 4.47	Clarified Meter Table Status Supported Report command parameter Clarified Meter Table Status Report command parameter Clarified Meter Table Historical Data Report command parameter. Clarified assignment of manufacturer ID.
12	20110322	JFR	Appendix A	Manufacturer ID's updated
12	20110503	JFR	Section 4.48 Section 4.28 Section 4.100 Section 4.37.3	Manufacturer Specific Command Class version 2 added CRC-16 Encapsulation Command Class version 1 added Thermostat Fan Mode Command Class version 3 added Typo w rt. sign bit in Geographic Location Report Command fixed
12	20110624	JFR	Section 4.87 Section 4.24	Added Schedule Entry Lock Command Class, version 2 and 3 Added Color Control Command Class, version 1
12	20110624	JFR	Section 4 Section 4.60 Section 4.62	Discontinued Multi Instance Command Class, version 1 Discontinued Multi Instance Association Command Class, version 1 Discontinued Multi Instance Commands within the Multi Channel Command Class, version 2 (now version 3) Multilevel Sensor Command Class, version 5 added.
12	20111214	JFR	Appendix A Section 4.112.2	Manufacturer ID's updated Protocol versions updated in Version Command Class
12	20111222	JFR	Section 3.3.1 Section 4.93.3	Clarified handling of undefined values of parameters in commands Clarified that a secure node supporting command classes insecurely must accept the same command classes security encapsulated.
12	20120120	JRM	Section 4.93.3.1	Added clarification regarding security encapsulation and multi endpoint handling.
13	20120214	JRM	Section 4.93.1 Section 4.93.2.2	Clarification that implicit ACK for Nonce Get is not allowed. Clarification that Scheme 0 bit value is 0
13	20120215	JFR	Section 4.85	Added Schedule Command Class, version 1

CONFIDENTIAL

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
13	20120227	JFR	Section 4.7 Section 4.67 Section 4.110 Section 4.115 Section 4.116	Added Application Capability Command Class, version 1 Added Network Management Command Classes - Network Management Proxy Command Class, version 1 - Network Management Basic Node Command Class, version 1 - Network Management Inclusion Command Class, version 1 - Network Management Primary Command Class, version 1 Added Transport Service Command Class, version 1 Added Z/IP Command Class, version 1 Added Z/IP-ND Command Class, version 1
13	20120229	JFR	Section 4.12 Section 4.31 Section 2.4	Added Association Group Information Command Class, version 1 Added Device Reset Locally Command Class, version 1 Clarification of requirement key words
14	20120329	JFR	Section 4.22	Added Central Scene Command Class, version 1
14	20120418	JFR	Section 4.55 Section 4.104 Section 4.70 Section 4.18	Added Meter Table Monitor Command Class, version 2 Added Thermostat Operating State Command Class, version 2 Added Notification Command Class, version 3 (former Alarm Command Class) Added Binary Sensor Command Class, version 2
14	20120430	JFR	Section 4.67.6.7	Added specified Mode field to command figure

Table of Contents

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

Table of Figures

CONFIDENTIAL

Table of Tables

CONFIDENTIAL

CONFIDENTIAL

1   ABBREVIATIONS

Abbreviation	Explanation
AEC	Advanced Energy Control
AMR	Automatic Meter Reading
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange. An ASCII code is the numerical representation of a character.
AV	Audio/Video
DHCP	Dynamic Host Configuration Protocol.
DNS	Dynamic Host Service
DST	Daylight Savings Time
HRV	Heat Recovery Ventilation
ID	Identifier
IP	Internet Protocol
IPV4	Internet Protocol version 4
IPV6	Internet Protocol version 6
LF	Linefeed character.
LSB	Less significant bit
MSB	Most significant bit
NIF	Node Information Frame
PIR	Pyroelectric Infrared Motion Sensor
SUC	Static Update Controller
TZO	Time Zone Offset
Unicode	Unicode is a standard for encoding of characters. For more information please visit http://www.unicode.org/
UTC	Universal Time (sometimes also called "Zulu Time") was called Greenwich Mean Time (GMT) before 1972
WMC	Windows Vista Media Center and Media Center 2005 remote controls
Z/IP	Z-Wave for IP

2   INTRODUCTION

This document describes the command classes and associated commands that MUST be used when designing and implementing Z-Wave™ products. A subset of command classes is typically mandatory for a given type of device. The application layer of the Z-Wave protocol handles all commands. Read this document in conjunction the Z-Wave Device Class Specification [1].

2.1   Purpose

The purpose of this document is to describe the command classes used by the application layer of the Z-Wave protocol.

2.2   Audience and prerequisites

The audience of this document is Z-Wave partners and Sigma Designs.

2.3   Precedence of definitions

In terms of reviewing products for Z-Wave Compliance, definitions in this document have precedence over the header file ZW_classcmd.h distributed as part of the Z-Wave Developer's Kit. However, assignments of hex identifiers for all device and command classes are located in the header file ZW_classcmd.h.

Device and Command Class Specifications approved as final version (ver. 1.00) during the device/command class development process have precedence over this document temporarily until integrated into this document.

2.4   Terms used in this document

This document describes mandatory and optional aspects of the required compliance of a Z-Wave product to the Z-Wave standard.

The guidelines outlined in IETF RFC 2119 [7] with respect to key words used to indicate requirement levels are followed. Essentially, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.
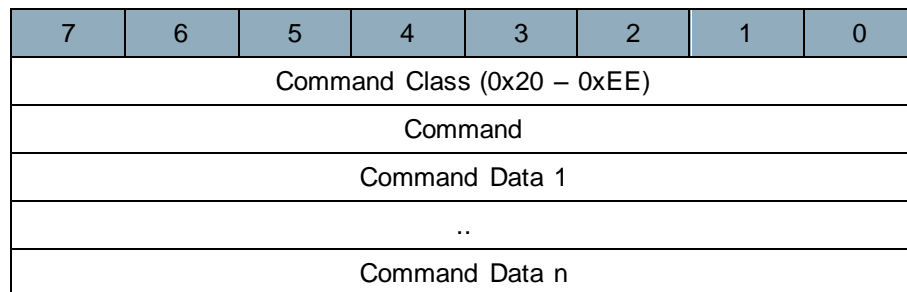
Products that are in violation any such statement are considered to be *not* Z-Wave compliant.

CONFIDENTIAL

3   COMMAND CLASS OVERVIEW
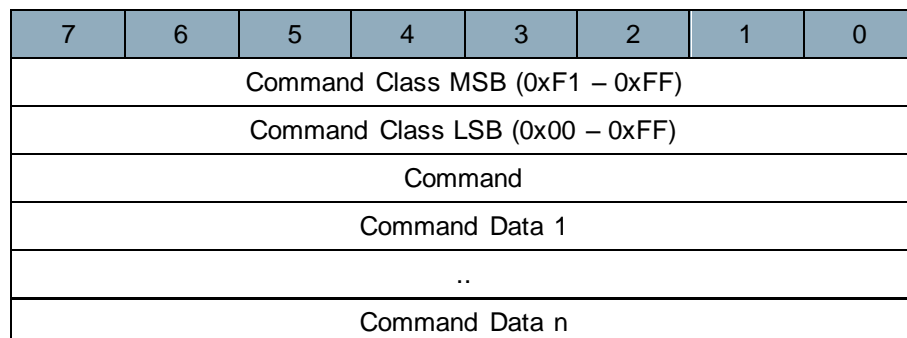
Interoperability between devices relies on Command Classes. If one device controls a command class and another device support the same command class then these devices are able to communicate.

3.1   Command class frame format

All commands have a common header consisting of a Command Class identifier and a Command identifier. Further, the command can have from zero to n bytes of command data. The bit-numbering scheme is "LSB 0" because bit numbering starts at zero for the least significant bit (Notice that LSB is denoted as 'Bit 0'... and MSB is denoted as 'Bit 7 throughout the document). The figures below show the generic command frame for the two possible formats:

7	6	5	4	3	2	1	0
Command Class (0x20 – 0xEE)							
Command							
Command Data 1							
..							
Command Data n							

Figure 1, Generic command format

7	6	5	4	3	2	1	0
Command Class MSB (0xF1 – 0xFF)							
Command Class LSB (0x00 – 0xFF)							
Command							
Command Data 1							
..							
Command Data n							

Figure 2, Generic extended command format

CONFIDENTIAL

3.1.1 Command class

The Command Class identifier range is as follows:

Command Class	Description
0x00	No Operation. Used by Z-Wave Protocol and OPTIONAL by the application.
0x01 – 0x1F	Reserved for the Z-Wave protocol
0x20 – 0xEE	Application Command Classes
0xEF	Support/Control Mark
0xF0	Non interoperable
0xF1 – 0xFF	Extended Application Command Classes

Table 1, Command Class identifier range

CONFIDENTIAL

A Command Class can contain up to 255 different Commands. If the Command Class field is set to 0xF1 through 0xFF then there is another Command Class byte added. This allows for future extensions of the Command Classes. The strategy of having an Extended Command Class followed by the actual command identifier provides the possibility of having more than 4000 Command Classes.

3.1.2 Command

The command field contains the specific command that SHOULD be executed and the field have a length of 8 bits.

3.1.3 Command data (0 – n*8 bits)

The command data field contains data related to the command. Simple commands, such as get commands, contain no command data. Other commands, such as set or report commands can contain several bytes of command data. Each data field has a length of 8 bits.

3.1.4 Command class versioning

All command classes have a version number. The following rules apply to avoid interoperability issues when introducing the same command class with different versions:

1. A node MUST NOT discard a frame based on the length field. Application MUST use command class identifier and the command identifier to interpret the application frame. This enables a device, which supports version 1 of a command class to interpret the version 1 part of a received version 2 of the command.

2. All implementations of a command class version higher then 1 MUST initialize all parameters associated with the version higher then 1. Thereby can a version 2 command class implementation in a device interpret a received version 1.

3. A device supporting a Command Class having a version higher than 1 MUST support the Version Command Class to be able to identify the supported version. In case the device does not support the Version Command Class then it can be assumed that all command classes are equal to version 1.

4. It is allowed to make devices only supporting an older version of the command class despite a newer version exists as long as the generic/specific device specification does not require a specific version implemented.

The number of data fields transmitted can be determined from the length field returned by the ApplicationCommandHandler. The length field used in cases where the same command has a variable number of command data fields.

CONFIDENTIAL

3.2   Command classes

The Command Class field defines a group of commands representing a given functionality. The currently defined Command Classes are split in two tables depending on the purpose. The first table shows the current list of general purpose Command Classes applicable for many different device types:

General Command Class	ZW_classcmd.h
Alarm	COMMAND_CLASS_ALARM
Application Capability	COMMAND_CLASS_APPLICATION_CAPABILITY
Application Status	COMMAND_CLASS_APPLICATION_STATUS
Association	COMMAND_CLASS_ASSOCIATION
Association Command Configuration	COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION
Association Group Information	COMMAND_CLASS_ASSOCIATION_GRP_INFO
Battery	COMMAND_CLASS_BATTERY
Central Scene	COMMAND_CLASS_CENTRAL_SCENE
Clock	COMMAND_CLASS_CLOCK
Configuration	COMMAND_CLASS_CONFIGURATION
Controller Replication	COMMAND_CLASS_CONTROLLER_REPLICATION
CRC-16 Encapsulation	COMMAND_CLASS_CRC_16_ENCAP
Device Reset Locally	COMMAND_CLASS_DEVICE_RESET_LOCALLY
Firmware Update Meta Data	COMMAND_CLASS_FIRMWARE_UPDATE_MD
Geographic Location	COMMAND_CLASS_GEOGRAPHIC_LOCATION
Grouping Name	COMMAND_CLASS_GROUPING_NAME
Hail	COMMAND_CLASS_HAIL
Indicator	COMMAND_CLASS_INDICATOR
IP Configuration	COMMAND_CLASS_IP_CONFIGURATION
Language	COMMAND_CLASS_LANGUAGE
Manufacturer Proprietary	COMMAND_CLASS_MANUFACTURER_PROPRIETARY
Manufacturer Specific	COMMAND_CLASS_MANUFACTURER_SPECIFIC
Mark (Support/control mark)	COMMAND_CLASS_MARK
Multi Channel	COMMAND_CLASS_MULTI_CHANNEL
Multi Channel Association	COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION
Multi Command	COMMAND_CLASS_MULTI_COMMAND
Network Management Basic Node	COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC
Network Management Inclusion	COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION
Network Management Primary	COMMAND_CLASS_NETWORK_MANAGEMENT_PRIMARY
Network Management Proxy	COMMAND_CLASS_NETWORK_MANAGEMENT_PROXY
No Operation	COMMAND_CLASS_NO_OPERATION
Node Naming and Location	COMMAND_CLASS_NODE_NAMING
Non interoperable	COMMAND_CLASS_NON_INTEROPERABLE
Notification (former Alarm)	COMMAND_CLASS_NOTIFICATION
Proprietary	COMMAND_CLASS_PROPRIETARY
Remote Association Activate	COMMAND_CLASS_REMOTE_ASSOCIATION_ACTIVATE
Remote Association Configuration	COMMAND_CLASS_REMOTE_ASSOCIATION
Screen Attributes	COMMAND_CLASS_SCREEN_ATTRIBUTES
Screen Meta Data	COMMAND_CLASS_SCREEN_MD
Security	COMMAND_CLASS_SECURITY
Time	COMMAND_CLASS_TIME
Time Parameters	COMMAND_CLASS_TIME_PARAMETERS
Transport Service	COMMAND_CLASS_TRANSPORT_SERVICE
User Code	COMMAND_CLASS_USER_CODE
Version	COMMAND_CLASS_VERSION
Wake Up	COMMAND_CLASS_WAKE_UP
Z-Wave for IP	COMMAND_CLASS_ZIP
Z-Wave for IP-ND	COMMAND_CLASS_ZIP_ND

Table 2, General purpose Command Class identifiers

CONFIDENTIAL

The second table shows the current list of Command Classes targeted for certain types of devices:

Device Related Command Class	ZW_classcmd.h
Alarm Sensor	COMMAND_CLASS_SENSOR_ALARM
Alarm Silence	COMMAND_CLASS_SILENCE_ALARM
All Switch	COMMAND_CLASS_SWITCH_ALL
Basic	COMMAND_CLASS_BASIC
Basic Tariff Information	COMMAND_CLASS_BASIC_TARIFF_INFO
Basic Window Covering	COMMAND_CLASS_BASIC_WINDOW_COVERING
Binary Sensor	COMMAND_CLASS_SENSOR_BINARY
Binary Switch	COMMAND_CLASS_SWITCH_BINARY
Binary Toggle Switch	COMMAND_CLASS_SWITCH_TOGGLE_BINARY
Climate Control Schedule	COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE
Color Control	COMMAND_CLASS_COLOR_CONTROL
DCP List Configuration	COMMAND_CLASS_DCP_CONFIG
DCP List Monitoring	COMMAND_CLASS_ DCP_ MONITOR
Door Lock	COMMAND_CLASS_DOOR_LOCK
Door Lock Logging	COMMAND_CLASS_DOOR_LOCK_LOGGING
Energy Production	COMMAND_CLASS_ENERGY_PRODUCTION
HRV Status Command Class	COMMAND_CLASS_HRV_STATUS
HRV Control Command Class	COMMAND_CLASS_HRV_CONTROL
Lock	COMMAND_CLASS_LOCK
Meter	COMMAND_CLASS_METER
Meter Table Configuration	COMMAND_CLASS_METER_TBL_CONFIG
Meter Table Monitor	COMMAND_CLASS_METER_TBL_MONITOR
Meter Table Push Configuration	COMMAND_CLASS_METER_TBL_PUSH
Move To Position Window Covering	COMMAND_CLASS_MTP_WINDOW_COVERING
Multilevel Sensor	COMMAND_CLASS_SENSOR_MULTILEVEL
Multilevel Switch	COMMAND_CLASS_SWITCH_MULTILEVEL
Multilevel Toggle Switch	COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL
Powerlevel	COMMAND_CLASS_POWERLEVEL
Prepayment	COMMAND_CLASS_PREPAYMENT
Prepayment Encapsulation	COMMAND_CLASS_PREPAYMENT_ENCAPSULATION
Protection	COMMAND_CLASS_PROTECTION
Pulse Meter	COMMAND_CLASS_METER_PULSE
Rate Table Configuration	COMMAND_CLASS_RATE_TBL_CONFIG
Rate Table Monitoring	COMMAND_CLASS_RATE_TBL_MONITOR
Scene Activation	COMMAND_CLASS_SCENE_ACTIVATION
Scene Actuator Configuration	COMMAND_CLASS_SCENE_ACTUATOR_CONF
Scene Controller Configuration	COMMAND_CLASS_SCENE_CONTROLLER_CONF
Schedule Entry Lock	COMMAND_CLASS_SCHEDULE_ENTRY_LOCK
Sensor Configuration	COMMAND_CLASS_SENSOR_CONFIGURATION
Simple AV Control	COMMAND_CLASS_SIMPLE_AV_CONTROL
Tariff Table Configuration	COMMAND_CLASS_TARIFF_TBL_CONFIG
Tariff Table Monitor	COMMAND_CLASS_TARIFF_TBL_MONITOR
Thermostat Fan Mode	COMMAND_CLASS_THERMOSTAT_FAN_MODE
Thermostat Fan State	COMMAND_CLASS_THERMOSTAT_FAN_STATE
Thermostat Mode	COMMAND_CLASS_THERMOSTAT_MODE
Thermostat Operating State	COMMAND_CLASS_THERMOSTAT_OPERATING_STATE
Thermostat Setback	COMMAND_CLASS_THERMOSTAT_SETBACK
Thermostat Setpoint	COMMAND_CLASS_THERMOSTAT_SETPOINT

Table 3, Device related Command Class identifiers

Refer to ZW_classcmd.h source code file for the assigned Command Class identifiers. In ZW_classcmd.h additional constants and variables are defined to support the implementation.

CONFIDENTIAL

3.3   Common command class variables

This section defines of a number of variables that are common used across command classes.

3.3.1   Reserved values and reserved bits

Values of fields in commands that are marked as "reserved" MUST NOT be used by devices sending commands and SHALL be ignored by devices receiving commands.

Bits in commands that are marked as "reserved" SHALL be set to 0 by devices sending commands and SHALL be ignored by devices receiving commands.

Undefined values for a given parameter are treated as "reserved" and MUST NOT be used by devices sending commands and SHALL be ignored by devices receiving commands.

3.3.2 Meter Dataset (24 bit)

The Meter Dataset variable defines the parameters supported by a certain meter; this is used among others to report capabilities, and when requesting data from a meter. Meter Dataset currently used by the AEC framework.

Meter Type	Byte #	Bit #	Description
Single-E electric meter Twin-E electric meter 3P Single Direct electric meter 3P Single ECT electric meter 1 Phase Direct Electricity Meter	Byte 1	Bit 0	<u>Total Primary Active Energy (kWh)</u> Energy delivered to the unit before transformation.
	Byte 1	Bit 1	<u>Total Primary Reactive Energy (kvarh)</u> Energy delivered to the unit before transformation.
	Byte 1	Bit 2	<u>Total Secondary Active Energy (kWh)</u> Consumed energy after transformation.
	Byte 1	Bit 3	<u>Total Secondary Reactive Energy (kvarh)</u> Consumed energy after transformation.
	Byte 1	Bit 4	<u>Instantaneous Primary Active Power (kW)</u> Current power consumption.
	Byte 1	Bit 5	<u>Instantaneous Primary Reactive Power (kvar)</u> Current power consumption.
	Byte 1	Bit 6	<u>Primary Active Maximum Demand (kW)</u> All time maximum active power demand.
	Byte 1	Bit 7	<u>Primary Reactive Maximum Demand (kvar)</u> All time maximum reactive power demand.
	Byte 2	Bit 0	<u>Cumulative Primary Active Maximum Demand (kW)</u> Accumulated power over period.
	Byte 2	Bit 1	<u>Cumulative Primary Reactive Maximum Demand (kvar)</u> Accumulated power over period.
	Byte 2	Bit 2	<u>Voltage Phase 1 (V)</u> Voltage measured on phase 1.
	Byte 2	Bit 3	<u>Voltage Phase 2 (V)</u> Voltage measured on phase 2.
	Byte 2	Bit 4	<u>Voltage Phase 3 (V)</u> Voltage measured on phase 3.
	Byte 2	Bit 5	<u>Ampere Phase 1 (A)</u> Amperes measured on phase 1.
	Byte 2	Bit 6	<u>Ampere Phase 2 (A)</u> Amperes measured on phase 2.
	Byte 2	Bit 7	<u>Ampere Phase 3 (A)</u> Amperes measured on phase 3.
	Byte 3	Bit 0	<u>Pulse Input (Pulse Count)</u> Pulse count on pulse input port of meter.

CONFIDENTIAL

	Byte 3	Bit 1	Current Transformation Ratio (ratio) Transformation ratio between primary and secondary energy.
	Byte 3	Bit 2	Power Factor (%) Power factor contains the ratio of real power to total power, and is expressed as a percentage (%).
	Byte 3	Bit 3-7	Reserved Value reserved for future expansion.
Gas meter Water meter	Byte 1	Bit 0	Accumulated Volume (m^3) Total accumulated volume.
	Byte 1	Bit 1	Current Flow (l/h, m^3/h) Current flow value.
	Byte 1	Bit 2	Current Pressure (kPa) Current pressure value.
	Byte 1	Bit 3	Peak Flow (l/h, m^3/h) Peak flow demand recorded.
	Byte 1	Bit 4	Hour Counter (Hours) Hours in operation.
	Byte 1	Bit 5	Input A (m^3/h, kW, m^3, kWh, MWh, Pulse Count) Value of input port A on the meter.
	Byte 1	Bit 6	Input B (m^3/h, kW, m^3, kWh, MWh, Pulse Count) Value of input port B on the meter.
	Byte 1 Byte 2 Byte 3	Bit 7 Bit 0-7 Bit 0-7	Reserved Value reserved for future expansion.
Heating meter Cooling meter	Byte 1	Bit 0	Heat Energy (kWh, MWh, GJ, GCal) Total accumulated energy used for heating.
	Byte 1	Bit 1	Cooling Energy (kWh, MWh, GJ, GCal) Total accumulated energy used for cooling.
	Byte 1	Bit 2	Volume 1 (m^3, ton) Volume counter 1 – forward flow.
	Byte 1	Bit 3	Volume2 (m^3, ton) Volume counter 2 – return flow.
	Byte 1	Bit 4	Temperature 1 (Forward) (C^o, ^oF) Forward temperature.
	Byte 1	Bit 5	Temperature 2 (Return) (C^o, ^oF) Return temperature.
	Byte 1	Bit 6	Temperature 3 (C^o, ^oF) Temperature sensor 3.
	Byte 1	Bit 7	Actual Flow (Volume 1) (l/h, m^3/h) Current flow on volume 1.

CONFIDENTIAL

	Byte 2	Bit 0	Actual Flow (Volume 2) (l/h, m³/h) Current flow on volume 2.
	Byte 2	Bit 1	Actual Power (kW, MW) Current power.
	Byte 2	Bit 2	Peak Flow (l/h, m³/h) Peak flow recorded.
	Byte 2	Bit 3	Peak Power (kW, MW) Peak power recorded.
	Byte 2	Bit 4	Input A (m³/h, kW, m³, kWh, MWh, Pulse Count) Value of input port A on the meter.
	Byte 2	Bit 5	Input B (m³/h, kW, m³, kWh, MWh, Pulse Count) Value of input port B on the meter.
	Byte 2	Bit 6	Hour Counter (Hours) Hours in operation.
	Byte 2 Byte 3	Bit 7 Bit 0-7	Reserved Value reserved for future expansion.
Electric Sub-Meter	Byte 1	Bit 0	Accumulated (kWh)
	Byte 1	Bit 1	Accumulated (kVAh)
	Byte 1	Bit 2	Instant (W)
	Byte 1	Bit 3	Accumulated (Pulse Count)
	Byte 1	Bit 4	Instant (V)
	Byte 1	Bit 5	Instant (A)
	Byte 1	Bit 6	Power Factor (%)
	Byte 1 Byte 2 Byte 3	Bit 7 Bit 0-7 Bit 0-7	Reserved

3.3.3   Meter Rate Type (2 bit)

The Meter Rate Type variable to indicate which rate is reported/requested or supported. Meter Rate Type currently used by the AEC framework.

Rate Type	Value
Reserved	0x00
Import	0x01
Export	0x02
Reserved	0x03

CONFIDENTIAL

3.3.4   Meter Scale (5 bit)

The Meter Scale variable used to report the scale (unit) of a reported value. Meter Scale currently used by the AEC framework.

Meter Type	Scale	Value
Electric meter	kWh	0x00
	kVARh	0x01
	%	0x02
	Pulse count	0x03
	kVAR	0x04
	Voltage (V)	0x05
	Amperes (A)	0x06
	kW	0x07
	Ratio	0x08
	Reserved	0x09- 0x1F
Gas and water meter	Cubic meter	0x00
	Cubic feet	0x01
	US gallon	0x02
	Pulse count	0x03
	IMP gallon	0x04
	Liter	0x05
	kPa	0x06
	Centum cubic feet	0x07
	Cubic meter per hour	0x08
	Liter per hour	0x09
	kWh	0x0A
	MWh	0x0B
	KW	0x0C
	Hours	0x0D
	Reserved	0x0E-0x1F

Heating and Cooling Meter	Cubic meter (m^3)	0x00
	Metric Ton (tonne) (t)	0x01
	Cubic meter per hour (m^3/h)	0x02
	Liter per hour (l/h)	0x03
	kW	0x04
	MW	0x05
	kWh	0x06
	MWh	0x07
	Giga Joule (GJ)	0x08
	Giga Calorie (Gcal)	0x09
	Celsius (C^o)	0x0A
	Fahrenheit (^{o}F)	0x0B
	Hours	0x0C
	Reserved	0x0D-0x1F
Electric Sub-Meter	kWh	0x00
	kVAh	0x01
	W	0x02
	Pulse Count	0x03
	V	0x04
	A	0x05
	Power Factor (%)	0x06
	Reserved	0x07-0x1F

3.3.5 Meter Type (6 bit)

The Meter Type variable describes the various meter types. Meter Type currently used by the AEC framework.

Meter Type	Value
Single-E electric meter	0x01
Gas meter	0x02
Water meter	0x03
Twin-E electric meter	0x04
3P Single Direct electric meter	0x05
3P Single ECT electric meter	0x06
1 Phase Direct Electricity Meter	0x07
Heating meter	0x08
Cooling meter	0x09
Combined Heating and Cooling Meter	0x0A
Electric Sub-Meter	0x0B
Reserved	0x0C-0x3F

4   COMMAND CLASS DEFINITIONS

The following subchapters contain definitions of the Command Classes listed in Chapter 0.

4.1   Alarm Command Class, version 1

The Alarm Command Class allows applications to report alarm or service conditions. Since these parameters are not standardized across devices it is RECOMMENDED that the alarms/service parameters be described in the user manual (or an installer manual).

4.1.1   Alarm Get Command

The Alarm Get Command used to get the value of an alarm.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_GET							
Alarm Type							

Alarm Type (8 bits)

The Alarm Type field specifies which alarm is being requested. The alarm types are specific for each application.

4.1.2   Alarm Report Command

The Alarm Report Command used to report the type and level of an alarm. Application can send unsolicited Alarm Report Commands or because of receiving an Alarm Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_REPORT							
Alarm Type							
Alarm Level							

Alarm Type (8 bits)

Refer to explanation under the Alarm Get Command.

Alarm Level (8 bits)

The alarm level is application specific.

CONFIDENTIAL

4.2   Alarm Command Class, version 2

The Alarm Command Class intended for Z-Wave enabled devices capable of reporting alarm reports.

Version 2 of the Alarm Command Class is improved with the following functionalities:

- Alarm Types defined by the Z-Wave Alliance
- Interview process of supported Alarm Types

The commands not mentioned here will remain the same as specified for Alarm Command Class (Version 1).

4.2.1   Alarm Set Command

The Alarm Set Command is used to set the activity of the Z-Wave Alarm Type and in addition the node ID that is to receive unsolicited reports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_SET							
Z-Wave Alarm Type							
Z-Wave Alarm Status							

Z-Wave Alarm Type (8 bit)

See Alarm Report Command.

Z-Wave Alarm Status (8 bit)

This field is used to set the state of the Alarm Type. The value 0x00 will deactivate the alarm and 0xFF will activate the alarm i.e. unsolicited Alarm Report Command will be transmitted to the node(s) defined in the Node field(s) when triggered by an event. Any other value is reserved for future use.

Note! The factory default state MUST be described in the product manual. All Z-Wave enabled devices MUST be able to operate based on factory default settings i.e. an end-user MUST NOT be forced to set-up the states of the device in order to operate. The factory default state of the Z-Wave Alarm Status is RECOMMENDED to be enabled.

Note! For products that do not allow deactivation of a specific Alarm Type, a received Alarm Configuration Set deactivating the Alarm Type in question MUST be responded with Application Rejected Request Command of the Application Status Command Class.

CONFIDENTIAL

4.2.2   Alarm Get Command

The Alarm Get Command is used to request the alarm state for a specific alarm type announced as supported through the Alarm Type Supported Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_GET							
Alarm Type							
Z-Wave Alarm Type							

Alarm Type (8 bit)

This field refers to the Alarm Type of Alarm Command Class (Version 1) i.e. the application specific Alarm Type which is not defined by the Z-Wave Alliance. If the 'V1 Alarm' field is set to '0' as reported via the Alarm Type Supported Report Command, this field MUST be set to '0' when requesting the report.

Z-Wave Alarm Type (8 bit)

The Z-Wave Alarm Type field MUST contain the Alarm Type identifier described in Alarm Report Command. This parameter refers to the Alarm Types defined by the Z-Wave Alliance.

Note! A device receiving an Alarm Get Command containing a non-supported Z-Wave Alarm Type MUST ignore the command. The transmitter of this command is therefore strongly RECOMMENDED to interview the device for supported Alarm Types by means of Alarm Type Supported Get Command prior to Alarm Get.

CONFIDENTIAL

4.2.3   Alarm Report Command

The Alarm Report Command MUST be transmitted as a response to the Alarm Get Command and in addition it can also be transmitted unsolicited in case the OEM application deems this necessary. The OEM application MUST however ensure not to transmit unsolicited report commands using a broadcast frame, in order to avoid unintentional transmission of an unsolicited command to a destination not ratifying support for it.

The Alarm Report Command is used by the application to report the alarm state.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_REPORT							
Alarm Type							
Alarm Level							
Zensor Net Source Node ID							
Z-Wave Alarm Status							
Z-Wave Alarm Type							
Z-Wave Alarm Event							
Number of Event Parameters							
Event Parameter 1							
...							
Event Parameter N							

Alarm Type (8 bit)

Same as for Alarm Command Class (Version 1).

Alarm Level (8 bit)

Same as for Alarm Command Class (Version 1).

Zensor Net Source Node ID (8 bit)

Specify the Zensor Net Source Node ID, which detected the alarm condition. In Zensor Net it is not possible to determine the Source Node ID due to the broadcast forwarded frame is without this information on protocol level. If the device is not based on Zensor Net this field MUST be set to '0'.

Z-Wave Alarm Status (8 bit)

See Alarm Set Command.

Number of Event Parameters (8 bit)

Indicates the Number of Event Parameters fields used in bytes.

CONFIDENTIAL

Z-Wave Alarm Type (8 bit), Z-Wave Alarm Event (8 bit) and Event Parameters (N Byte)

The table below specifies the Alarm Types and its subordinate parameters defined by the Z-Wave Alliance. New entries can be requested from Sigma Designs. Contact Z-Wave Support at support@zen-sys.com. The fields that do not contain any definition of the Z-Wave Alarm Type MUST be set to '0' in the Alarm Report Command.

Note! The device MUST advertise support of the Command Class which is included for the specific Alarm Type. Example: for Smoke Alarm, Smoke Detected the Node Naming and Location Command Class MUST be advertised as supported in the Node Information Frame.

Z-Wave Alarm Type (8 bit)		Z-Wave Alarm Event (8 bit)		Event Parameters (N Byte)
Reserved	0x00	Reserved	0x00	
Smoke Alarm	0x01	Reserved	0x00	
		Smoke detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Smoke detected, Unknown Location	0x02	
		Reserved	0x03-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
CO Alarm	0x02	Reserved	0x00	
		Carbon monoxide detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Carbon monoxide detected, Unknown Location	0x02	
		Reserved	0x03-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
CO$_2$ Alarm	0x03	Reserved	0x00	
		Carbon dioxide detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Carbon dioxide detected, Unknown Location	0x02	
		Reserved	0x03-0xFD	
		Unknown Event	0xFE	

CONFIDENTIAL

Z-Wave Alarm Type (8 bit)		Z-Wave Alarm Event (8 bit)		Event Parameters (N Byte)
		Reserved	0xFF	
Heat Alarm	0x04	Reserved	0x00	
		Overheat detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Overheat detected, Unknown Location	0x02	
		Rapid Temperature Rise	0x03	Node Location Report (Node Naming and Location Command Class, version 1)
		Rapid Temperature Rise, Unknown Location	0x04	
		Underheat detected	0x05	Node Location Report (Node Naming and Location Command Class, version 1)
		Underheat detected, Unknown Location	0x06	
		Reserved	0x07-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
Water Alarm	0x05	Reserved	0x00	
		Water Leak detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Water Leak detected, Unknown Location	0x02	
		Water Level Dropped	0x03	Node Location Report (Node Naming and Location Command Class, version 1)
		Water Level Dropped, Unknown Location	0x04	
		Reserved	0x05-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
Access Control Alarm	0x06	Reserved	0x00	
		Manual Lock Operation	0x01	

Z-Wave Alarm Type (8 bit)		Z-Wave Alarm Event (8 bit)		Event Parameters (N Byte)
		Manual Unlock Operation	0x02	
		RF Lock Operation	0x03	
		RF Unlock Operation	0x04	
		Keypad Lock Operation	0x05	User Identifier of User Code Report (User Code Command Class)
		Keypad Unlock Operation	0x06	User Identifier of User Code Report (User Code Command Class)
		Reserved	0x07-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
Burglar Alarm	0x07	Reserved	0x00	
		Intrusion	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Intrusion, Unknown Location	0x02	
		Tampering, product covering removed	0x03	
		Tampering, Invalid Code	0x04	
		Glass Breakage	0x05	Node Location Report (Node Naming and Location Command Class, version 1)
		Glass Breakage, Unknown Location	0x06	
		Reserved	0x07-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
Power Management Alarm	0x08	Reserved	0x00	
		Power has been applied	0x01	
		AC mains disconnected	0x02	
		AC mains re-connected	0x03	
		Surge Detection	0x04	

Z-Wave Alarm Type (8 bit)		Z-Wave Alarm Event (8 bit)		Event Parameters (N Byte)
		Voltage Drop/Drift	0x05	
		Reserved	0x06-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
System Alarm	0x09	Reserved	0x00	
		System hardware failure	0x01	
		System software failure	0x02	
		Reserved	0x03-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
Emergency Alarm	0x0A	Reserved	0x00	
		Contact Police	0x01	
		Contact Fire Service	0x02	
		Contact Medical Service	0x03	
		Reserved	0x04-0xFD	
		Unknown Event	0xFE	
		Reserved	0xFF	
Alarm Clock	0x0B	Reserved	0x00	
		Wake Up	0x01	
Reserved for future use	0x0C-0xFE			
Return first Alarm on supported list	0xFF			

Alarm Type = 0xFF is used by the Alarm Get Command to retrieve the first alarm detection from the supported list.

Example: a device supports the Z-Wave Alarm Types: Smoke, CO_2 and Heat. The Heat Alarm is active e.g. overheat has been detected. When the device receives Alarm Get, Z-Wave Alarm Type (0xFF), it MUST return Alarm Report, Z-Wave Alarm Type (0x04), Z-Wave Alarm Event (0x01/0x02) and the accompanied parameters.

4.2.4   Alarm Type Supported Get Command

The Alarm Type Supported Get Command is used to request the supported alarm types.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_TYPE_SUPPORTED_GET							

4.2.5   Alarm Type Supported Report Command

The Alarm Type Supported Report Command used to report the supported alarm types in the application. The Alarm Type Supported Report Command is transmitted as a result of a received Alarm Type Supported Get Command and MUST NOT be sent unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ALARM							
Command = ALARM_TYPE_SUPPORTED_REPORT							
V1 Alarm	Reserved		Number of Bit Masks				
Bit Mask 1							
...							
Bit Mask N							

V1 Alarm (1 bit)

This field is set to '1' if the product has non Z-Wave Alliance defined Alarm Types implemented. When set to '0' only Z-Wave Alliance Alarm Types are used.

Reserved (2 bit)

This field is reserved and MUST be set to 0.

Number of Bit Masks (5 bit)

Indicates the Number of Bit Masks fields used in bytes.

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported <u>Z-Wave Alarm Types</u> by the device.

Bit 0 in Bit Mask 1 is not allocated to any Z-Wave Alarm Type and MUST therefore be zeroed.
Bit 1 in Bit Mask 1 field is used to indicate support of Z-Wave Alarm Type = 1 (Smoke).
Bit 2 in Bit Mask 1 field is used to indicate support of Z-Wave Alarm Type = 2 (CO).
Bit 3 in Bit Mask 1 field is used to indicate support of Z-Wave Alarm Type = 3 (CO_2) and so forth.
The Z-Wave Alarm Type is supported if the bit is 1 and the opposite if 0.
Z-Wave Alarm Type = 0xFF (Return first Alarm on supported list) can not be indicated by the Bit Masks.

Note! It is only necessary to transmit Bit Mask 1 and up to the Bit Mask N indicating the last supported sensor type. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.3 Alarm Command Class, version 3

Refer to Notification Command Class version 3.

CONFIDENTIAL

4.4   Alarm Sensor Command Class, version 1

> It´s RECOMMENDED using Alarm Command Class instead for new devices.

The Alarm Sensor Command Class can be used to realize Sensor Alarms.

4.4.1   Alarm Sensor Get Command

The Alarm Sensor Get Command can be used to request the status of a sensor.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_ALARM							
Command = SENSOR_ALARM_GET							
Sensor Type							

Sensor Type (8 bits)

Sensor type specifies what type of sensor this command originates from. Refer to the table below with respect to defined sensors. The sensor type value 0xFF returns the first found supported sensor type in the bit mask (starting from bit 0 in Bit Mask 1) by the Alarm Sensor Supported Report. New sensor types/values can be requested from Sigma Designs.

Sensor Type	Value
General Purpose Alarm	0x00
Smoke Alarm	0x01
CO Alarm	0x02
CO_2 Alarm	0x03
Heat Alarm	0x04
Water Leak Alarm	0x05
Return first Alarm on supported list	0xFF

CONFIDENTIAL

4.4.2   Alarm Sensor Report Command

Application can send unsolicited Alarm Sensor Report Commands when the alarm state changes or because of receiving an Alarm Sensor Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_ALARM							
Command = SENSOR_ALARM_REPORT							
Source Node ID							
Sensor Type							
Sensor State							
Seconds 1 (MSB)							
Seconds 2 (LSB)							

Source Node ID (8 bits)

Specify the source node ID, which detected the alarm condition. In a Zensor Net is it not possible to determine the source node ID because the frame is broadcast forwarded without this information on protocol level.

Sensor Type (8 bits)

See description under Alarm Sensor Get. The Sensor Type equal to 0xFF cannot be return by the report.

Sensor State (8 bits)

The Sensor State parameter returns the current alarm state. The value 0x00 indicates no alarm and 0xFF indicates alarm. Furthermore it can return values from 0x01 to 0x64 to indicate severity of the alarm in percentage.

The values 101…254 (0x65…0xFE) are reserved and SHALL be ignored by receiving devices.

Seconds 1..2 (16 bits)

The field Seconds indicates time the remote alarm MUST be active since last received report. The value 0x0000 indicates that the time field MUST be ignored.

4.4.3   Alarm Sensor Supported Get Command

The Alarm Sensor Supported Get Command is used to request the supported sensor types from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_ALARM							
Command = SENSOR_ALARM_SUPPORTED_GET							

4.4.4 Alarm Sensor Supported Report Command

The Alarm Sensor Supported Report used to report the supported sensor types from the device. It can be send as a result of receiving a Alarm Sensor Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_ALARM							
Command = SENSOR_ALARM_SUPPORTED_REPORT							
Number of Bit Masks							
Bit Mask 1							
…							
Bit Mask N							

Number of Bit Masks (8 bits)

Indicates the Number of Bit Masks fields used in bytes.

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported sensor types by the device. The bit 0 in Bit Mask 1 field used to indicate whether Sensor Type = 0 (General Alarm) is supported or not. The sensor type is supported if the bit is 1 and the opposite if 0. The bit 1 in Bit Mask 1 field used by Sensor Type = 1 (Smoke Alarm) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported sensor type. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.5   Alarm Silence Command Class, version 1

The Alarm Silence Command Class can be used to nuisance silence to temporarily disable the sounding of the alarm but still keep the alarm operating.

4.5.1   Alarm Silence Set Command

The Alarm Silence Set Command can be used to remotely silence the sensor alarm.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SILENCE_ALARM							
Command = SENSOR_ALARM_SET							
Mode							
Seconds 1 (MSB)							
Seconds 2 (LSB)							
Number of Bit Masks							
Bit Mask 1							
…							
Bit Mask N							

Mode (8 bits)

Sensor type specifies what type of sensor this command originates from. Refer to the table below with respect to defined sensors. New sensor types/values can be requested from Sigma Designs.

Mode	Value
Disable sounding of all sensor alarms independent of bit mask	0x00
Disable sounding of all sensor alarms independent of bit mask which have received the alarm via the Sensor Alarm Report command	0x01
Disable sounding of all sensor alarms according to bit mask	0x02
Disable sounding of all sensor alarms according to bit mask which have received the alarm via the Alarm Sensor Report Command	0x03

Seconds 1..2 (16 bits)

The field Seconds indicates the duration sounding of the alarm MUST be disable but still keep the alarm operating. If silence is engaged, the alarm will come back on when the duration expires unless the originating sensor clears the alarm. The value 0x0000 indicates that the time field MUST be ignored.

Number of Bit Masks (8 bits)

Indicates the Number of Bit Masks fields used in bytes.

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the sensor types to disable sounding from. The bit 0 in Bit Mask 1 field used to indicate whether Sensor Type = 0 (General Alarm) is to be disabled or not. The sensor type MUST be disabled if the bit is 1 and the opposite if 0. The bit 1 in Bit Mask 1 field used by Sensor Type = 1 (Smoke Alarm) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last sensor type to be disabled.

CONFIDENTIAL

4.6   All Switch Command Class, version 1

The All Switch Command Class used to switch all devices on or off. Devices can be excluded/included from the all on/all off functionality. The application determines which devices there are included in the all on/all off functionality as default.

4.6.1   All Switch Set Command

The All Switch Set Command used to tell a device if it SHOULD be included or excluded from the all on/all off functionality.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_SET							
Mode							

Mode (8 bits)

The mode field used to set the all on/all off functionality of the device.

Mode	Description
0x00	Indicate that the switch is excluded from the all on/all off functionality.
0x01	Indicate that the switch is excluded from the all on functionality but not all off.
0x02	Indicate that the switch is excluded from the all off functionality but not all on.
0xFF	Indicates that the switch is included in the all on/all off functionality.

4.6.2   All Switch Get Command

The All Switch Get Command can be used to ask a device if it is included or excluded from the all on/all off functionality.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_GET							

4.6.3   All Switch Report Command

The All Switch Report Command used to report if the device is included or excluded from the all on/all off functionality. Application can send unsolicited All Switch Report Commands or because of receiving an All Switch Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_REPORT							
Mode							

Mode (8 bits)

Refer to explanation under the All Switch Command.

4.6.4   All Switch On Command

The All Switch On Command used to inform a switch that it SHOULD be turned on. Notice that the command will cause the device to change to the level when the device last was on.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_ON							

4.6.5   All Switch Off Command

The All Switch Off Command used to inform a switch that it SHOULD be turned off.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_ALL							
Command = SWITCH_ALL_OFF							

4.7　　Application Capability Command Class, version 1

The Application Capability Command Class comprises commands for handling issues related to dynamic support for command classes.
Examples include nodes, which only support certain command classes when included securely, and controllers that only support command classes for primary controllers when they are actually operating as primary controllers.

The intention of the Application Capability Command Class is to provide tools for handling exceptions. Controlling nodes SHOULD maintain a local representation of supported command classes by destination nodes rather than relying on destination nodes returning Application Capability messages.

Destination nodes capable of returning Application Capability messages MUST list Application Capability Command Class as supported only.

4.7.1　　Not Supported Command Class Command

The Not Supported Command Class Command is used by a node to indicate to a requesting node that the requested command class is not supported. The offending command and command class are encapsulated.

A node SHOULD use this command to indicate to other nodes that a command is not supported.

A node receiving this command SHOULD use the information to clean up internal states by requesting up-to-date information on command class support from the sending node using Node Information frame. This includes adjusting user interface details accordingly.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_APPLICATION_CAPABILITY							
Command = COMMAND_COMMAND_CLASS_NOT_SUPPORTED							
Dyna-mic	Reserved						
Offending Command Class (0x20 – 0xEE)							
Offending Command							

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_APPLICATION_CAPABILITY							
Command = COMMAND_COMMAND_CLASS_NOT_SUPPORTED							
Dyna-mic	Reserved						
Offending Command Class MSB (0xF1 – 0xFF)							
Offending Command Class LSB (0x00 – 0xFF)							
Offending Command							

CONFIDENTIAL

Payload data following the offending command is omitted.

Dynamic (1 bit)

If the "Dynamic" flag is set ('1'), the sending node has been designed to support this command class but the current operational conditions prevent the node from supporting this command class.
One example is the AddNode command used for network management. This command is only supported when a controller is operating as primary controller.

Value	Meaning
'0'	Command not supported (Permanently)
'1'	Dynamic support. (Currently no support)

Reserved (7 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Offending Command Class (1 or 2 bytes)

This field indicates the offending command class received by the destination node. The size of the command class field depends on the value of the first byte.

Offending Command (1 byte)

This field indicates the offending command received by the destination node.

4.8   Application Status Command Class, version 1

All devices SHOULD as far as possible support the Application Status Command Class. This class contains commands that are not directly related to a specific functionality in the application, but are useful for maintaining an optimal Z-Wave system.

4.8.1   Application Busy Command

The Application Busy Command used to instruct a node that the node that it is trying to communicate with is busy and is unable to service the request right now.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_APPLICATION_STATUS							
Command = APPLICATION_BUSY							
Status							
Wait Time							

Status (8 bits)

The status field can have the following values

Status	Description
0	Try again later
1	Try again in Wait Time seconds
2	Request queued, executed later

Wait Time (8 bits)

The wait time field indicates the number of seconds a node SHOULD wait before retrying the request.

CONFIDENTIAL

4.8.2   Application Rejected Request Command

All supported commands are typically executed unconditionally and the only handshake is acknowledgement on the protocol level. Some applications can however be in a state where the application rejects to execute a supported command. The Application Rejected Request Command used to instruct a node that the command was rejected by the application in the receiving node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_APPLICATION_STATUS							
Command = APPLICATION_REJECTED_REQUEST							
Status							

Status (8 bits)

The status field can have the following values

Status	Description
0	Supported command rejected by the application in the receiving node

CONFIDENTIAL

4.9   Association Command Class, version 1

The Association Command Class defines the commands necessary to create and maintain associations in a device. The associations are a list of devices the device wants to control on application level. Some devices MAY wish to have several groupings of associated nodes so each can be controlled by different events. The groupings are addressed by a Grouping Identifier with up to 255 different groupings of nodes associated to different events.

To configure a controller from another node requires support of the Association Command Class because it needs information about which nodes to be used in the routing table.

A routing slave has not a routing table as found in a controller. This requires that the routing slave in addition is configured with return routes for the wanted associations to obtain a reliable and robust network.

4.9.1   Association Set Command

The Association Set Command used to add nodes to a given grouping identifier. The node receiving the set command SHOULD add the nodes received to the nodes already associated by this grouping until the grouping is full. Remember that routing slaves also MUST have assigned return routes by a controller using the API call ZW_AssignReturnRoute [1] to all the associated nodes. Delete all return routes by the API call ZW_DeleteReturnRoute before assignment. This is of course necessary for all the associated nodes out of direct range but SHOULD always be done default to all the associated nodes to create a reliable and robust network. It's OPTIONAL whether the return routes are assigned before or after the Association Set command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_SET							
Grouping identifier							
Node ID1							
..							
NodeIDn							

Grouping identifier (8 bits)

This grouping identifier used to identify a group of nodes. The grouping identifier values MUST be a sequence starting from 1. Ignore field in case the node only supports one grouping.

NodeID1 .. NodeIDn

These fields contain a list of node IDs that SHOULD be associated with the grouping.

4.9.2   Association Get Command

The Association Get Command is used to request the current association status of the node. The node receiving this command SHOULD answer with Association Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GET							
Grouping Identifier							

Grouping identifier (8 bits)

The grouping identifier field used to identify the nodes grouped together. The grouping identifier values MUST be a sequence starting from 1. Ignore field in case the node only supports one grouping.

CONFIDENTIAL

4.9.3   Association Report Command

The Association Report Command SHOULD be used to report all nodes associated with the matching grouping identifier. Be aware that it's only possible to get information about how many groupings a given node are associated to, but not the total number of groupings. Therefore the Grouping Identifiers SHOULD be allocated with care starting from 0x01 to avoid unnecessary overhead in finding the groupings with associations. A remote with up to 6 different groupings there are controlled by 6 buttons numbered 1...6 could use the same numbers as grouping identifiers. Application can send unsolicited Association Report Commands or because of receiving an Association Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_REPORT							
Grouping Identifier							
Max Nodes Supported							
Reports to Follow							
NodeID1							
…							
NodeIDn							

Grouping identifier (8 bits)

This grouping identifier used to identify the requested report.

Max Nodes Supported (8 bits)

Maximum number of nodes grouping identifier above supports.

Reports to Follow (8 bits)

This value indicates how many report frames left before transferring the entire list of node IDs associated with the given grouping identifier.

NodeID1 .. NodeIDn

These fields contain a list of node IDs that are associated with the given grouping identifier. Return no NodeIDx fields in case of an empty list.

4.9.4   Association Remove Command

The Association Remove Command used to remove nodes from a given grouping at the node receiving the command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_REMOVE							
Grouping identifier							
Node ID1							
..							
Node IDn							

Grouping identifier (8 bits)

This grouping identifier used to determine from which grouping the supplied node ID's SHOULD be removed.

NodeID1 .. NodeIDn

These fields contain a list of node ID's that SHOULD be removed from the specified grouping identifier. In case no node ID's are supplied the whole grouping SHOULD be cleared.

4.9.5   Association Supported Groupings Get Command

The Association Supported Groupings Get Command is used to request the number of groupings that this node supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GROUPINGS_GET							

CONFIDENTIAL

4.9.6   Association Supported Groupings Report Command

The Association Supported Groupings Report Command used to report the maximum number of groupings the given node supports. Application can send unsolicited Association Supported Groupings Report Commands or because of receiving an Association Supported Groupings Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_GROUPINGS_REPORT							
Supported Groupings							

Supported Groupings (8 bits)

The number of groupings this node supports.

CONFIDENTIAL

4.10 Association Command Class, version 2

The Association Command Class defines the commands necessary to create and maintain associations in a device. The following is a list of the command that has been changed or added in version 2. The commands not mentioned here will remain the same.

4.10.1 Association Remove Command

The Association Remove Command used to remove nodes from a given grouping at the node receiving the command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_REMOVE							
Grouping identifier							
Node ID1							
..							
Node IDn							

Grouping identifier (8 bits)

This grouping identifier used to determine from which grouping the supplied node ID's SHOULD be removed.

NodeID1 .. NodeIDn

These fields contain a list of node ID's that SHOULD be removed from the specified grouping identifier. In case no node ID's are supplied the whole grouping SHOULD be cleared.

In version 2 grouping identifier in conjunction with sequence of NodeID's are interpreted as follows:

	Grouping identifier	Number of node ID's in list
Clear all node ID's in grouping X	$1 \leq X \leq N$	0
Clear specified node ID's in all groupings	0	>0
Clear all node ID's in all groupings	0	0

4.10.2 Association Specific Group Get Command

The Association Specific Group Get Command is used to request the current active group from a node. This can be used to set up an association to a specific group using a remote.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_SPECIFIC_GROUP_GET							

CONFIDENTIAL

4.10.3   Association Specific Group Report Command

The Association Specific Group Report Command used to report the current active group. Application can send unsolicited Association Specific Group Report command or because of receiving an Association Specific Group Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION							
Command = ASSOCIATION_SPECIFIC_GROUP_REPORT							
Group							

Group (8 bits)

The current active group number (1-255). Set to 0 if undefined (the RS might not support the feature).

CONFIDENTIAL

4.11 Association Command Configuration Command Class, version 1

The Association Command Configuration Command Class defines the commands necessary for a 2nd node to add and delete commands to Node IDs in a group as defined in the Association Command Class in a 1st node.

Mandatory requirement: The device MUST implement the Association Command Class as 'supported'

Mandatory requirement: The Association Command Class and the Command Configuration Command Class MUST be linked through the following dependencies

A) Nodes added to an association through an Association Set or Association Composite Set MUST be reported in Command Configuration Reports with the command class and command identifiers transmitted to the nodes.

B) Nodes added to an association through a Command Configuration Set MUST also be reported in an Association Report and Association Composite Report

C) All commands associated to a grouping identifier//Node ID pair will be removed as a result of an Association Remove. The related command records will be released

D) Command(s) associated to a grouping identifier/Node ID/endpoint pair will be removed as a result of an Association Composite remove. The related command(s) record will be released.

The memory consumption of supporting full command sizes in all combinations of groupings identifiers and Node IDs is very extensive; hence the command class supports a memory flexible implementation.

The command class allows a device to support a number of command records. A command record consists of the grouping identifier, the Node ID and the command. The size of the command can be restricted by the device through the Max command length field. The command MUST be the complete command needed (I.e. All relevant encapsulations MUST be included in the command).

When no command records are free (all has been used), no new commands can be allocated to Node IDs before one or more command records have been freed up (through the Association Remove Command)

If the 2nd node runs out of free command records before it has finalized its command configurations, it MUST accept that the application on the device has full control of the remaining Node IDs. Alternatively the 2nd node can abort the command configuration process. In this case it is RECOMMENDED that the 2nd node free up the used command records in the aborted command configuration attempt.

A device can report the maximum number of command records, the number of free command records, and the max command length supported through the Command Records Supported Report.

In order to support sharing knowledge of how a device controls nodes without using extensive memory resources a Configurable Cmd field is supported. When configurable Cmd= 0x0 then a 2nd node can only monitor the commands. It can not control them.

The V/C field allows a device to decrease the memory utilization to a minimum. When a device reports V/C=0x01, then the Command Configuration Set and Command Configuration Report MUST always use command class identifier and command identifier equal to a Basic Set Command Records Supported Get. This allows the device to only store the value field and thereby save memory resources.

CONFIDENTIAL

4.11.1 Command Records Supported Get Command

The Command Records Supported Get Command is used to request the number of free command records available (grouping Identifier, Node ID, command), the maximum command records supported in the device and information regarding the maximum command length supported in the device.

The maximum number of groupings the given node supports is available through the Association Command Class [1].

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION							
Command = COMMAND_RECORDS_SUPPORTED_GET							

4.11.2 Command Records Supported Report Command

The Command Records Supported Report Command used to report information regarding the Command records. Application can send unsolicited Command Records Supported Report Commands or because of receiving a Command Records Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION							
Command = COMMAND_RECORDS_SUPPORTED_REPORT							
Max command length						V/C	Conf. Cmd
Free Command records 1							
Free Command records 2							
Max Command records 1							
Max Command records 2							

CONFIDENTIAL

Configurable Cmd (1 bit)

Configurable Cmd	Functionality
0	The local application has full control of the commands associated with the grouping. The commands can be monitored from a 2nd network node.
1	The specific commands associated with the grouping can be controlled and monitored from a 2nd network node. This option includes also the level field used by the command *Transfer Scene* in the *Controller Replication Command* Class. In this case the level value is transferred via the command *Basic Set* in the *Basic Command Class*.

V/C (1 bit)

V/C	Functionality
0	Command type. A Z-Wave command can be added to the node
1	Value type. A Value field is specified for a Node ID using the command *Basic Set* in the *Basic Command Class*. Level field originates from the command *Transfer Scene* in the *Controller Replication Command* Class.

Max command length (6 bits)

If Configurable Cmd is equal to 0x0, the field SHOULD return 0x0.

If Configurable Cmd is equal to 0x1 the field SHOULD return the maximum length of a command which can be associated to a node in a grouping. The minimum max command length allowed is 0x03.

Example:

A product reports a Max command length = 0x03. In this case the product can be programmed with a Switch Multilevel Set, but not a Switch Multilevel Start Level Change.

Switch Multilevel Set has a command length of 3

Switch Multilevel Start Level Change has a command length of 4

Free Command Records 1-2 (16 bits)

The field specifies the current number of free Command Records which can be configured in the device through the Command Configuration Set Command.

Max Command records 1-2 (16 bits)

The field specifies the maximum number of Command Records which can be configured in the device through the Command Configuration Set Command.

4.11.3 Command Configuration Set Command

The Command Configuration Set Command used to specify which commands SHOULD be sent to nodes within a given Grouping identifier.

Every Command Configuration Set will utilize one Command record from the pool of free Command records.

If the device has no free command records when receiving the Command Configuration Set, the command will be ignored.

The Application on the node MAY alter the commands when needed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION							
Command = COMMAND_CONFIGURATION_SET							
Grouping identifier							
Node ID							
Command length							
Command Class identifier							
Command identifier							
Command byte 1							
..							
Command byte n							

Grouping identifier (8 bits)

This grouping identifier used to specify how nodes are grouped together. The grouping identifier values MUST be a sequence starting from 1. This field can be ignored in case the node only supports one group.

Node ID (8 bits)

This field contains the node ID within the grouping specified, that SHOULD receive the command.

Command length (8 bits)

This field specifies the complete command length (including command class and command identifiers).

Example:

Switch Multilevel Set 0x20. The command length field MUST be equal to 0x03.

Command Class Identifier (8 bits)

This field contains the identifier of the command class which SHOULD be sent to the Node ID. In case the Configurable Cmd field is equal to 0x0 is this field and the following ignored and can therefore be omitted.

CONFIDENTIAL

In case the device has reported V/C=0x1, the Command Class and Command Identifiers MUST be equal to Basic Set command.

Command identifier (8 bits)

This field contains the identifier of the Command, which SHOULD be sent to the specified Node ID. In case the Configurable Cmd field is equal to 0x0 is this field ignored and can therefore be omitted.

In case the device has reported V/C=0x1, the Command Class and Command Identifiers MUST be equal to Basic Set Command.

Command byte1 .. Command byte n

These fields contain the command parameters, which SHOULD be sent to the Node ID. In case the Configurable Cmd field is equal to 0x0 is these fields ignored and can therefore be omitted.

4.11.4 Command Configuration Get Command

The Command Configuration Get Command is used to request the commands specified for to a Node ID within a given Grouping identifier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION							
Command = COMMAND_CONFIGURATION_GET							
Grouping identifier							
Node ID							

Grouping identifier (8 bits)

This group identifier used to specify how nodes are grouped together. The group identifier values MUST be a sequence starting from 1. This field MUST be ignored in case the node only supports one group.

Node ID (8 bits)

This field specifies the node ID within the grouping.

4.11.5 Command Configuration Report Command

The Command Configuration Report Command used to report the commands specified for a Node ID within a given Grouping identifier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_COMMAND_CONFIGURATION							
Command = COMMAND_CONFIGURATION_REPORT							
Grouping identifier							
Node ID							
First	Reserved			Reports to follow			
Command length							
Command Class identifier							
Command identifier							
Command byte 1							
..							
Command byte n							

Grouping identifier (8 bits)

This group identifier identifies the group. The group identifier values MUST be a sequence starting from 1.

Node ID (8 bits)

This field contains the node ID as requested in the Association Command Get.

Reports to follow (4 bits)

This value indicates how many report frames left before transferring the entire list of commands.

First (1 bit)

This field indicates that this report is the first report relating to a Grouping identifier/Node ID pair

Command length (8 bits)

This field specifies the complete command length (including command class and command identifiers).

Example:

CONFIDENTIAL

Switch Multilevel Set 0x20. The command length field MUST be equal to 0x03.

Command Class Identifier (8 bits)

This field contains the identifier of the command class which is sent to the Node ID.

In case the device has reported V/C=0x1, the Command Class and Command Identifiers MUST be equal to Basic Set command

Command identifier (8 bits)

This field contains the identifier of the Command which is sent to the specified Node ID.

In case the device has reported V/C=0x1, the Command Class and Command Identifiers MUST be equal to Basic Set Command

Command byte1 .. Command byte n

These fields contain the command parameter which is sent to the Node ID.

4.12   Association Group Information command class, version 1

The purpose of the Association Group Information Command Class is to allow Controlling devices to report information pertaining to the control capabilities of the device for each association group. This will allow plug-and-play association in the following cases:

- Controller-assisted Button-to-button Association
- Button-to-button Association
- GUI-based Association

In another case, the Association Group Information command class allows discovery of specific sensors, e.g. "Look for devices supporting Command Class = Multilevel Sensor and Profile = sensor Temperature".

This command class comprises a number of commands. A combination of commands MUST be used to gain access to all properties of an association group. For a full configuration, the following commands MUST be issued:

- Association Group Info Get
- Association Group Command List Get
- Association Group Name Get

In addition, a requesting node SHOULD get the currently associated target nodes using the classic association command class.

4.12.1   Association Principles

A classic Z-Wave remote control has a number of control groups and typically a few speed keys. The remote control outlined in Figure 3 has 6 speed keys. Each speed key also maps to one of the control groups.

Figure 3, Remote Control for Z-Wave

Figure 4, Lamp module for Z-Wave

A remote control speed key is associated with one or more Z-Wave target devices, e.g. lamp modules like the one depicted in Figure 4.

A remote control intended for AV control typically has a number of keys for different functions applying to the same target, e.g. "Play", "Stop", "Pause", etc. An example is outlined in Figure 5.

CONFIDENTIAL

Figure 5, Remote Control for AV control

Figure 6, TV, DVD and Satellite Receiver

When creating associations from an AV remote control to an AV target device, e.g. a TV, a group of keys are represented by a single Z-Wave association group. Considering the universal remote control depicted in Figure 5, five association groups are defined for the following target devices: "CABLE", "TV", "VCR", "DVD", "SAT".

Each association group represents a unique profile in the remote control. Each profile is selected via a button press.

As an example, the AV stack shown in Figure 6 is controlled via the association groups representing the "TV", "DVD" and "SAT" profiles. When using the "TV" profile, the remote control sends all commands to the TV while with the "SAT" profile, all commands are sent to the Satellite Receiver (also the "Volume Up" command which MAY not be supported by a satellite receiver).

4.12.1.1   Event Codes – introducing groupings

A vendor MAY want to provide a fully configurable remote control where each key MAY be configured individually. It is however impractical to ship a Remote Control with 10 profiles where each key maps to an individual Association Group. This would quickly use up all 255 available Association Groups and at the same time, it would be impossible for a do-it-yourself user to do a quick configuration to get started.

The Simple AV command class defines numerous message codes. The Simple AV specification does not indicate any grouping of keys. Some keys are however logically grouped in all remote controls.

Each association group of a controlling device is described by a Profile and an Event Code. The event code MAY be any normal Z-Wave Simple AV command or a special code representing a group, e.g. grpAll.

CONFIDENTIAL

4.12.2   The Association Group Information Table

A Controlling device typically implements a number of association groups. The following tables provide examples of the use of association groups in different device types.

The use of the fields is explained in the following.

Table 4 outlines a number of groups in a sensor device.

Group #	Profile 2 bytes	Event Code 2 bytes	Command Class & Command (list) N bytes	Group Name (UTF-8)
1	0	0	WakeUp Notification	Lifeline
2	0	0	Basic Set	On/Off (Temperature)
3	Sensor: Temperature	0	Sensor Report	Temperature
4	0	0	Tamper Alarm, Battery Alarm	Device Status

Table 4, Layout of Association Group Information for a sensor device

Universal remote controls only constitute one functional unit. A high number of control groups MAY be defined in one association group information table.

Group #	Profile 2 bytes	Event Code 2 bytes	Command Class & Command (list) N bytes	Group Name (UTF-8)
1	0	0	WakeUp Notification	Lifeline
2	TV	grp All	Simple AV	TV
3	DVD	grp All	Simple AV	DVD
4	Satellite	grp All	Simple AV	Satellite
5	Cable	grp All	Simple AV	Cable
6	PVR	grp All	Simple AV	Recorder
7	0	0	Tamper alarm, Battery alarm	Device Status

Table 5, Layout of Association Group for a universal remote control

CONFIDENTIAL

4.12.2.1   Group number

The Group number indicates which association group the actual table line relates to. Since association groups are always numbered from one and upwards, an actual implementation does not have to store the group number in physical memory.

4.12.2.2   Profile

Some products MAY implement a number of profiles. A sensor MAY use the Profile field for advertizing for example Sensor:Temperature.
A remote control MAY advertize the profiles "TV" and "DVD".
The profile field carries values representing Generic and Specific device classes.

When no device type is applicable, the profile value MUST be set to 0 (2 bytes). Examples include Basic commands sent to a power switch from a sensor and device status reports such as the Battery command class.

4.12.2.2.1   Sensor type profiles

The Profile field is also used by sensors to advertize the sensor device type.
The Event Code field MUST be set to zero for sensors.

A sensor product MAY implement multiple sensor functions; e.g. one temperature sensor and one humidity sensor. In this case the device MUST implement two multichannel end points; one for the "sensor Temperature" sensor type and the other for the "sensor Relative humidity" sensor type profile.
An Association Group Information table MUST be presented for each endpoint device.

The use of multichannel endpoints allows for control of and service discovery for specific sensor types – even if they reside in the same physical device.
Classic Z-Wave nodes use a dedicated Multichannel command to encapsulate commands intended for multichannel endpoints. The encapsulation allows for the addressing of multichannel endpoints.
An IP enabled device uses a unique IPv6 address and one endpointID for each multichannel endpoint, as well as an endpointID (zero) for the entire device.

4.12.2.3   Event Code

A device MAY advertize a high number of association groups supporting the same command class. The Event Code is used to distinguish apparently identical association groups. For AV remotes, generally the Event Code grpAll SHOULD be used, to indicate that the remote control maps all buttons to the actual association group. Other groups MAY be used if only a section of the keys are mapped, e.g. grpPreset 1-10.

Association groups used for Z-Wave control purposes also use the Event code for button mapping. The Event Code Preset 1 indicates that the remote control uses numerical key 1 as a speed key for the actual Z-Wave control group. When clicking key 1 the remote control then sends MultiLevel Switch, Binary Switch or Basic command class, depending on the device type.

The event codes map directly to the command table, specified in Simple AV Control command class. A number of events have been added to the table to allow the different groupings, such as grpAll – as shown in Table 6.

CONFIDENTIAL

Group ID	Typical use
grp All Functions	All commands for one device
grp Audio Adjust	Volume down, up
grp Video Adjust	Brightness down, up
grp TV Adjust	Channel down, up
grp Navigation	Down, up, OK
grp Media Control	Play, Stop
grp Preset 1-10	Numbered buttons 0,1,2,…
grp Color Keys	Red, Green, Blue, Yellow
grp Power Control	Power off

Table 6, Event codes for Simple AV Control Groups in the Association Group Information table

The Event Code MUST be set to zero (2 bytes) if not used.

4.12.2.4   Command Class and Command Identifier

The command class and the command that MAY be sent to targets associated with this association group.
This MAY be a list of command classes. The list MAY contain a combination of extended and normal command classes.

4.12.2.5   Association group name

A vendor MAY use the association group name for any meaningful purpose.
42 bytes are available for text. Text MUST be encoded using UTF-8. The available capacity for characters depends on the actual characters encoded. Plain ASCII characters only occupy one byte while special characters MAY need two or more bytes for representation.

It is RECOMMENDED that association groups are named to reflect the sensor profile, e.g. "Temperature", "Humidity", the AV profile, e.g. "TV", "DVD" or if no applicable profile: The purpose of the group, e.g. "Device status".

Group 1 MUST be named "Lifeline". The Lifeline group MUST map to the WakeUp Notification. A gateway MAY create an association from this association group to itself without knowing anything about other functions of the actual device. A gateway SHOULD verify that the group name is "Lifeline".

4.12.3   Association Group Name Get

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_GRP_INFO							
Command = ASSOCIATION_GROUP_NAME_GET							
Grouping Identifier							

Grouping Identifier (1 byte)

The number of the actual association group.

4.12.4   Association Group Name Report

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_GRP_INFO							
Command = ASSOCIATION_GROUP_NAME_REPORT							
Grouping Identifier							
Length of Name							
Name 1							
...							
Name N							

Grouping Identifier (1 byte)

The number of the actual association group.

Length of Name (1 byte)

How many bytes to read for Name field. Valid values are in the range 1..42.

Name (N bytes)

The name of the requested group. Byte string with no zero termination. The length is defined by the "Length of Name" field. Bytes MUST be encoded in UTF-8 format.

4.12.5   Association Group Info Get

Function used to get properties for an association group.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_GRP_INFO							
Command = ASSOCIATION_GROUP_INFO_GET							
Refresh cache	List Mode	Reserved					
Grouping Identifier							

List Mode (1 bit)

If List Mode is true ('1') then one report returns info on all association groups. A receiving node MAY return many reports, e.g. due to memory constraints.

If List Mode is false ('0') then multiple reports return information on one association group at a time.

Refresh Cache (1 bit)

A Service Router MUST cache information for all nodes; also listening nodes. A Service Router SHOULD return a cached response on behalf of the specified target.
This parameter indicates that a Service Router MUST, on the first chance given, gather new information from the specified target.
A Service Router MUST refresh the cache if it does not hold cached information for the specified target when receiving an Association Group Information Get command.

Reserved

Bits currently not used. MUST be set to zero by a transmitting node. MUST be ignored by a receiving node.

Grouping Identifier (1 byte)

The number of the actual association group. This value MAY be ignored if the "List Mode" field is "1".

CONFIDENTIAL

4.12.6   Association Group Info Report

Report format for list mode and report-by-report mode.

The message returns properties for one or more association groups.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_GRP_INFO							
Command = ASSOCIATION_GROUP_INFO_REPORT							
List mode	Dynamic Info	Group Count					
Grouping Identifier							
Mode = 0							
Profile MSB							
Profile LSB							
Reserved							
Event Code MSB							
Event Code LSB							
...							
Grouping Identifier							
Mode = 0							
Profile MSB							
Profile LSB							
Reserved							
Event Code MSB							
Event Code LSB							

List Mode (1 bit)

If List Mode is true ('1') then information is returned on all association groups. Each block of data is fixed length. Each block starts with the Grouping Identifier representing the actual block of data.

If List Mode is false ('0') then information is returned in one report at a time.

Note that even if List Most is used, several reports MAY be received if the sending node is only able to send shorter lists or if the sending node implements a high number of association groups.

Dynamic Info (1 bit)

If bit is set to 1, the information MAY change and the Service Router SHOULD perform periodic cache refresh for this node. Nodes MUST set this bit if they are able to change information.

If this bit is cleared, a Service Router only requests this information once. The cache entry remains valid as long at WakeUp Notifications are received at the REQUIRED intervals.

CONFIDENTIAL

Group Count (6 bit)

Number of groups in the frame.

Grouping Identifier (1 byte)

The number of the actual association group.

Mode = 0 (1 byte)

Mode field intended for advanced applications. In the context of the AGI command class, all transmitted commands MUST carry Mode = 0. If a node only implementing the AGI command class[1] receives a report signaling Mode values other than zero, the entire report MUST be ignored.

Profile (2 bytes)

Mode=0: Generic and Specific device type that indicates the intended device type of the target device. Profile MSB carries the Generic device type. Profile LSB carries the Specific device type.

Reserved (1 byte)

Bits currently not used. MUST be set to zero by a transmitting node. MUST be ignored by a receiving node.

Event Code (2 bytes)

A Simple AV Control code defining the intended mechanism for controlling targets in this association group. The Event Code MAY represent a unique event like "Preset 1" or an entire group like "grpAll".

---

[1] i.e. the node does NOT implement the AAGI command class

CONFIDENTIAL

4.12.7   Association Group Command List Get

Command used to request the commands that MAY be sent to targets of this the association group.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_GRP_INFO							
Command = ASSOCIATION_GROUP_COMMAND_LIST_GET							
Allow cache	Reserved						
Grouping Identifier							

Allow Cache (1 bit)

This parameter indicates that a Service Router is allowed to intercept the request and return a cached response on behalf of the specified target.
If this bit is not set, a Service Router MUST forward the request to the specified target.

A requesting node SHOULD allow caching to save network bandwidth. A Service Router MUST cache information for all nodes; also listening nodes. This will save network bandwidth.

A Service Router MUST forward the request if it does not hold cached information for the specified target. The Service Router MUST cache the data returned.

Reserved (7 bit)

Bits currently not used. MUST be set to zero by a transmitting node. MUST be ignored by a receiving node.

Grouping Identifier (1 byte)

The number of the actual association group.

CONFIDENTIAL

4.12.8   Association Group Command List Report

Command reporting the commands that MAY be sent from the association group.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ASSOCIATION_GRP_INFO							
Command = ASSOCIATION_GROUP_COMMAND_LIST_REPORT							
Grouping Identifier							
List Length							
Command A 1							
Command A 2							
Command A 3							
Command B 1							
Command B 2							
Command C 1							
Command C 2							
Command C 3							
…							

Grouping Identifier (1 byte)

The number of the actual association group.

List Length (1 byte)

The length (in bytes) of the command list.

Command (n bytes)

Commands MAY be normal or extended commands.
Normal command classes are represented as one byte while extended command classes are represented as two bytes. Thus, a command list entry (command class + command) is either 2 or 3 bytes long.

The receiving node MUST parse individual command list entries to determine if the individual command class is a normal or an extended command class.
The first byte of normal command classes is in the range 0x20 – 0xEE, while the first byte of extended command classes is in the range 0xF1 – 0xFF.

No command payload bytes are included in the command list.

4.13 Basic Command Class, version 1

All devices will if possible support the Basic Commands Class. This class contains a small number of very basic commands that can be used to control the basic functionality of a device. The Commands include the possibility to set a given level, get a given level and report a level.

The basic Commands enables a controller application to use the basic Commands on a device that is unknown to the controller and thereby give the user control over the main functionality of the device. The actual usage of the basic Commands is described for each generic/specific device.

Typically, generic and/or specific device classes define mappings from the Basic Command Class to specific Commands. A device SHALL implement the mappings as specified in the generic and/or specific device classes that are published by the device in its node info frame. Specifications for the mapping of the Basic Command Class in the corresponding specific device class have precedence over specifications in the generic device class.

NOTE: To avoid unintentionally operation of devices it is RECOMMENDED not using broadcasts for the Basic Command Class because many devices will support this command class.

4.13.1 Basic Set Command

The Basic Set Command will be used by the different devices, to set dim level, temperature, state, water level, speed etc., in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC							
Command = BASIC_SET							
Value							

Value (8 bits)

The value field can be used to set different levels in the device.

A controlling device MAY use any of the values 0…255 (0x00…0xFF) in a Set Command. Since the Z-Wave compliant mapping of the Basic Command MAY define that certain values are ignore by the target device that received a Basic Set Command from a controlling device, a controlling device SHALL NOT assume that the device will always act on any Basic Set Command. It is the responsibility of the device that is controlled with a Basic Set Command to ignore reserved / undefined values.

A device that is controlled by Basic Commands SHALL implement Basic Set as specified in the corresponding generic / specific device class. If there is no further specification beyond the definition of a mapping to another Command defined, the device SHALL provide exactly the same behavior, as if the set Command would have been given with that Command.

4.13.2 Basic Get Command

The Basic Get Command will be used by the different devices to get dim level, temperature, state, water level, speed etc., from a device.

Devices SHALL implement Basic Get and Basic Report as specified in the corresponding generic / specific device class. If there is no further explanation beyond the definition of a mapping to another Command defined, the device SHALL provide exactly the same behavior, as if the set Command would have been given with that Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC							
Command = BASIC_GET							

4.13.3 Basic Report Command

The Basic Report Command will be used by the different devices to report dimming level, temperature, state, water level, speed etc., from a device. Application can send unsolicited Basic Report Commands or because of receiving a Basic Get Command.

A device SHALL respond to a Basic Get always with a Basic Report, i.e. not with the Report Command of the mapped-to device class.

A controlling device MUST be able to accept any of the values 0…255 (0x00…0xFF) in a Basic Report. It is beyond the scope of the Z-Wave specification if and which values received with a Basic Report a controlling device MAY ignore.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC							
Command = BASIC_REPORT							
Value							

Value (8 bits)

The value field can be use to report different levels in the device.

CONFIDENTIAL

4.14   Basic Tariff Information Command Class, version 1

This Basic Tariff Information Command Class for use with a single element or dual element meter, and for use with import (electricity received from grid) rates only. The command class is kept as simple as possible without any pricing information.

This command class supports a GET and REPORT.

No SET command is supported, as it is not appropriate to set any of the parameters through Z-Wave.

4.14.1   Basic Tariff Information Get Command

The Basic Tariff Information Get command is used to request current tariff information from the meter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC_TARIFF_INFO							
Command = BASIC_TARIFF_INFO_GET							

The response is the Basic Tariff Information Report.

CONFIDENTIAL

4.14.2 Basic Tariff Information Report Command

The Basic Tariff Information Report command returns information on the number of import rates supported, and current import rate information. Application can send unsolicited Basic Tariff Report commands or requested by the Basic Tariff Get Information command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC_TARIFF_INFO							
Command = BASIC_TARIFF_INFO_REPORT							
Dual	Reserved			Total No. Import Rates			
Reserved				E1 – Current Rate in Use			
E1 - Rate Consumption Register – MSB							
E1 - Rate Consumption Register							
E1 - Rate Consumption Register							
E1 - Rate Consumption Register – LSB							
E1 – Time for Next Rate – Hours							
E1 – Time for Next Rate – Minutes							
E1 – Time for Next Rate – Seconds							
Reserved				E2 – Current Rate in Use			
E2 - Rate Consumption Register – MSB							
E2 - Rate Consumption Register							
E2 - Rate Consumption Register							
E2 - Rate Consumption Register – LSB							

Dual (1 bit)

Single Element = 0, Two Elements = 1.

If single element the E2 fields are skipped in the frame. E1 – Time for Next Rate – Seconds will be the last byte of the message and the number of data bytes will be 9.

If two elements the E2 fields are present and the number of data bytes will be 14.

Total Number of Import Rates Supported (7 bits)

Field specifies the number of import rates E1 (and E2) supported by the meter. Range of legal decimal values are 1…8. No units used. The decimal values 0 and 9…15 are reserved and SHALL be ignored by receiving devices.

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

E1 – Current Rate in Use (8 bits)

Field specifies the current rate in use. Range of legal decimal values are 1…8. No units used. The decimal values 0 and 9…15 are reserved and SHALL be ignored by receiving devices.

E1 – Rate Consumption Register (32 bits)

The meter has a 32-bit consumption register, for the energy used in each rate. This register is the rate consumption register for the current rate in use now on element 1. Units are in Wh.

E1 – Time for Next Rate – Hours (8 bits)

Field specifies the hour value of the time that the rate is due to change on element 1. Range of legal decimal values are 0…23, or 255. The values 24…254 are reserved and SHALL be ignored by receiving devices.

E1 – Time for Next Rate – Minutes (8 bits)

Field specifies the minute value of the time that the rate is due to change on element 1. Range of legal decimal values are 0…59, or 255. The decimal values 60…254 are reserved and SHALL be ignored by receiving devices.

E1 – Time for Next Rate – Seconds (8 bits)

Field specifies the second value of the time that the rate is due to change on element 1. Range of legal decimal values are 0…59, or 255. The decimal values 60…254 are reserved and SHALL be ignored by receiving devices.

NOTE: 255 in each field of the Time to Next Rate specifies no switching time is in use, which is appropriate for single rate meters. 255 is only a legal value if used in all three Time to Next Rate fields.

E2 – Current Rate in Use (8 bits)

Field specifies the current rate in use on element 2. Range of legal decimal values are 1…8. No units used. The decimal values 0 and 9…15 are reserved and SHALL be ignored by receiving devices.

E2 – Rate Consumption Register (32 bits)

The meter has a 32-bit consumption register, for the energy used in each rate. This register is the rate consumption register for the current rate in use now on element 2. Units are in Wh.

CONFIDENTIAL

4.15 Basic Window Covering Command Class, version 1

It´s RECOMMENDED using Multi-level switch generic device class with the Multi-position Motor specific device class instead for new devices.

This section contains Commands that can be used to control a Basic Window Covering Command Class.

4.15.1 Basic Window Covering Start Level Change Command

The Basic Window Covering Start Level Change Command used to start moving drapes, shades, blinds in a given direction. The speed of the movement is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC_WINDOW_COVERING								
Command = BASIC_WINDOW_COVERING_START_LEVEL_CHANGE								
Reserved	Open/ Close	Reserved					

Open/Close (1 bit)

If the Open/Close bit is set to 0 the window covering SHOULD open. If field is set to 1 the window covering SHOULD close.

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.15.2 Basic Window Covering Stop Level Change Command

The Basic Window Covering Stop Level Change Command used to stop moving drapes, shades, blinds in a given direction.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BASIC_WINDOW_COVERING								
Command = BASIC_WINDOW_COVERING_STOP_LEVEL_CHANGE								

4.16 Battery Command Class, version 1

The Battery Command Class is used to request and report battery levels for a given device.

CONFIDENTIAL

4.16.1 Battery Level Get Command

The Battery Level Get Command are used to request the level of a battery.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BATTERY							
Command = BATTERY_GET							

4.16.2 Battery Level Report Command

The Battery Level Report Command used to report the battery level of a battery operated device. Application can send unsolicited Battery Level Report Commands or because of receiving a Battery Level Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_BATTERY							
Command = BATTERY_REPORT							
Battery Level							

Battery Level (8 bits)

The battery level reports percentage of the full battery. The field can take values from 0 to 100% (0x00 – 0x64). The value 0xFF indicates a battery low warning.

CONFIDENTIAL

4.17   Binary Sensor Command Class, version 1

It´s RECOMMENDED using Alarm Command Class instead for new devices.

The Binary Sensor Command Class can be used to realize binary sensors, such as movement sensors.

4.17.1   Binary Sensor Get Command

The Binary Sensor Get Command can be used to request the status of a sensor.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_BINARY							
Command = SENSOR_BINARY_GET							

4.17.2   Binary Sensor Report Command

Application can send unsolicited Binary Sensor Report Commands or because of receiving a Binary Sensor Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_BINARY							
Command = SENSOR_BINARY_REPORT							
Sensor Value							

Sensor Value (8 bits)

If the Sensor value is 0x00 indicates that the sensor is idle and 0xFF indicates that the sensor has detected an event.

CONFIDENTIAL

4.18　Binary Sensor Command Class, version 2

> **It´s RECOMMENDED using Alarm Command Class instead for new devices.**

The Binary Sensor Command Class can be used to realize binary sensors, such as movement sensors and door/window sensors. Version 2 of this command class is extended with the following functionalities:

- A "get-supported" mechanism for the controlling device to interview the binary sensor for its supported sensor types

- A list of defined sensor types capable of reporting a binary value

NOTE: A binary sensor is defined as a sensor unit capable of providing a binary value in the report i.e. that an event was "triggered" or "not triggered" and not other intermediate values. For sensor units providing multiple values the Multilevel Sensor Command Class is more suitable. The Binary Sensor Command Class can further advance through implementation of the Notification Command Class to communicate details about the event.

4.18.1　Binary Sensor Get Command

The Binary Sensor Get Command can be used to request the status of the specific sensor device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_BINARY							
Command = SENSOR_BINARY_GET							
Sensor Type							

CONFIDENTIAL

Sensor Type (8 bits)

Sensor Type specifies what type of sensor this command originates from. Refer to the table below with respect to defined sensors. The sensor type value 0xFF returns the first found supported sensor type in the bit mask (starting from bit 0 in Bit Mask 1) by the Binary Sensor Supported Report. New sensor types/values can be requested from Sigma Designs.

Sensor Type	Value
Reserved	0x00
General purpose	0x01
Smoke	0x02
CO	0x03
CO_2	0x04
Heat	0x05
Water	0x06
Freeze	0x07
Tamper	0x08
Aux	0x09
Door/Window	0x0A
Tilt	0x0B
Motion	0x0C
Glass Break	0x0D
Reserved	0x0E-0xFE
Return 1st Sensor Type on supported list	0xFF

4.18.2   Binary Sensor Report Command

Application can send unsolicited Binary Sensor Report Command or as a reply to a received Binary Sensor Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_BINARY							
Command = SENSOR_BINARY_REPORT							
Sensor Value							
Sensor Type							

Sensor Value (8 bits)

Sensor Value = 0x00 indicates that the sensor is idle and 0xFF indicates that the sensor has detected an event.

CONFIDENTIAL

4.18.3   Binary Sensor Get Supported Sensor Command

The Binary Sensor Get Supported Sensor Command is used to request the supported sensor types from the binary sensor device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_BINARY							
Command = SENSOR_BINARY_SUPPORTED_GET_SENSOR							

4.18.4   Binary Sensor Supported Sensor Report Command

This command must be sent as requested by a received Binary Sensor Get Supported Sensor Command. This command indicates the supported sensor types of the binary sensor device in a bit mask format and shall not be transmitted unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_BINARY							
Command = SENSOR_BINARY_SUPPORTED_SENSOR_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported sensor types by the binary sensor device and refer to the Sensor Type table of Binary Sensor Report command.

Bit 0 in Bit Mask 1 is not allocated to any Binary Sensor Type and must therefore be zeroed.
Bit 1 in Bit Mask 1 field is used to indicate support of Sensor Type = 1 (General Purpose).
Bit 2 in Bit Mask 1 field is used to indicate support of Sensor Type = 2 (Smoke).
Bit 3 in Bit Mask 1 field is used to indicate support of Sensor Type = 3 (CO) and so forth. The Sensor Type is supported if the bit is 1 and the opposite if 0.
Sensor Type = 0xFF (Return 1st Sensor Type on supported list) cannot be indicated by the Bit Masks.

NOTE! It is only necessary to transmit Bit Mask 1 and up to the Bit Mask N indicating the last supported sensor type. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.19 Binary Switch Command Class, version 1

This chapter describes the Commands that can be used to make binary switches. These Commands allow applications to set and get the status of a binary switch.

4.19.1 Binary Switch Set Command

The Binary Switch Set Command can be used to set a device on or off (enable or disable).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_BINARY								
Command = SWITCH_BINARY_SET								
Value								

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). The values from 1 to 99 (0x01 – 0x63) SHALL mapped to 0xFF upon receipt of the Command in the device.

All other values are reserved and SHALL be ignored by the receiving device.

Controlling devices MAY send any of the values 0x00…0x63 and 0xFF to the device. Controlling devices MUST NOT send any of the reserved values to the device.

Note:
Based on this specification of the Binary Switch Commands and the Basic Command Class mappings for the corresponding generic / specific devices, sending a single Basic Set Command to control a mixed group of Binary and Multilevel Switches is possible. Upon receipt of such a Command, enables that the Binary Switches would simply turn ON, while Multilevel Switches would turn to the selected level.

4.19.2 Binary Switch Get Command

The Binary Switch Get Command can be used to get the status of the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_BINARY							
Command = SWITCH_BINARY_GET							

4.19.3 Binary Switch Report Command

Application can send unsolicited Binary Switch Report Commands or because of receiving a Binary Switch Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_BINARY							
Command = SWITCH_BINARY_REPORT							
Value							

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). Devices MUST NOT respond with a Binary Switch Report with any other value.

4.20   Binary Toggle Switch Command Class, version 1

Do not use this command class for new devices.

Use instead Binary Switch Generic Device Class for such devices.

This chapter describes the Commands that can be used to make binary toggle switches. These Commands allow applications to set and get the status of a binary toggle switch.

4.20.1   Binary Toggle Switch Set Command

The Binary Toggle Switch Set Command can be used to toggle a device e.g. from on to off and from off to on.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_BINARY							
Command = SWITCH_TOGGLE_BINARY_SET							

4.20.2   Binary Toggle Switch Get Command

The Binary Switch Get Command can be used to request the state of the load controlled by the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_BINARY							
Command = SWITCH_TOGGLE_BINARY_GET							

4.20.3   Binary Toggle Switch Report Command

Application can send unsolicited Binary Toggle Switch Report Commands or because of receiving a Binary Toggle Switch Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_BINARY							
Command = SWITCH_TOGGLE_BINARY_REPORT							
Value							

Value (8 bits)

The value can be either 0x00 (off) or 0xFF (on).

CONFIDENTIAL

4.21   Climate Control Schedule Command Class, version 1

> It´s RECOMMENDED using Schedule Command Class instead for new
> devices.

The Climate Control Schedule Command Class allows devices to exchange schedules and overrides,
which specify when to perform a setback on the setpoint.

Note: The setpoint is the temperature a device will try to maintain. The setback is a deviation from the
         setpoint. When a setback is in use the device will apply the setback to the setpoint, resulting in a
         different temperature. When using schedules and overrides it is possible to define several
         setbacks occurring at specific times.

Schedules SHALL be exchanged using the Schedule Commands.
Overrides of schedules SHALL be exchanged using the Schedule Override Commands.
Detection of updated schedules SHALL be done using the Schedule Changed Commands.

The Climate Control Schedule uses the Schedule State type to define each setback. The Schedule State
type has the following format:

7	6	5	4	3	2	1	0
Schedule State							

Schedule State (8 bits)

The values are as follows:

Schedule State		Description
Hexadecimal	**Decimal**	
0x80 … 0xFF 0x00 0x01 … 0x78	-128 … -1 0 1 … 120	The setback in 1/10 degrees (Kelvin) Example: 0 = 0 degrees setback 1 = 0.1 degrees is added to the setpoint 2 = 0.2 degrees is added to the setpoint -1 = 0.1 degrees is subtracted from the setpoint -2 = 0.2 degrees is subtracted from the setpoint
0x79	121	Frost Protection
0x7A	122	Energy Saving Mode
0x7B – 0x7E	123 – 126	Reserved
0x7F	127	Unused State

**When converting between Celsius and Fahrenheit proper rounding MUST be applied with at least
two decimals in the internal calculations of a device to avoid rounding errors.**

CONFIDENTIAL

When displaying converted Fahrenheit values it is RECOMMENDED that the displayed value is rounded to nearest quarter of a degree.

4.21.1 Schedule Set Command

The Schedule Set Command used to set the climate control schedule in a device for a specific weekday. A climate control schedule defines when to use a setback on the setpoint in a device. A schedule can hold a maximum of 9 switchpoints. A switchpoint defines one setback from the current setpoint.

The entire list of switchpoints in the Command MUST be ordered by time, ascending from 00:00 towards 23:59. Switchpoints which have a Schedule State set to "Unused" SHALL be placed last. No duplicates SHALL be allowed for Switchpoints which have a Schedule State different from "Unused".

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_SET							
Reserved				Weekday			
Switchpoint 0 Byte 1							
Switchpoint 0 Byte 2							
Switchpoint 0 Byte 3							
Switchpoint 1 Byte 1							
Switchpoint 1 Byte 2							
Switchpoint 1 Byte 3							
…							
Switchpoint 8 Byte 1							
Switchpoint 8 Byte 2							
Switchpoint 8 Byte 3							

Weekday (3 bits)

The possible values are:

Binary	Decimal	Description
0b000	0	Reserved
0b001	1	Monday
0b010	2	Tuesday
0b011	3	Wednesday
0b100	4	Thursday
0b101	5	Friday
0b110	6	Saturday
0b111	7	Sunday

Switchpoint (24 bits)

7	6	5	4	3	2	1	0	
Reserved			Hour					Byte 1
Reserved		Minute						Byte 2
Schedule State								Byte 3

Hour (5 bits):

Specifies the hour during a day when this switchpoint SHALL be used. Possible values are:

Binary	Decimal	Description
0b00000	0	0 hour
0b00001	1	1st hour
0b00010	2	2nd hour
…	…	…
0b10111	23	23rd hour
0b11000	24	Reserved
…	…	Reserved
0b11111	31	Reserved

Minute (6 bits):

Specifies the minute during the specified hour when this switchpoint SHALL be used. Possible values are:

Binary	Decimal	Description
0b000000	0	0 minute
0b000001	1	1st minute
0b000010	2	2nd minute
…	…	…
0b111011	59	59th minute
0b111100	60	Reserved
…	…	Reserved
0b111111	63	Reserved

Schedule State (8 bits):

Schedule State uses the Schedule State type format, see section 0. If Schedule State has the value of "Unused", then the Hour and Minute field SHALL be ignored. Once a Schedule State of "Unused" is encountered, the parsing of switchpoints SHALL stop.

4.21.2 Schedule Get Command

The Schedule Get Command is used to request the climate control schedule in a device for a specific weekday.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_GET							
Reserved					Weekday		

Weekday (3 bits)

The possible values are:

Binary	Decimal	Description
0b000	0	Reserved
0b001	1	Monday
0b010	2	Tuesday
0b011	3	Wednesday
0b100	4	Thursday
0b101	5	Friday
0b110	6	Saturday
0b111	7	Sunday

4.21.3 Schedule Report Command

The Schedule Report Command used to report the climate control schedule in a device for a specific weekday.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_REPORT							
Reserved						Weekday	
Switchpoint 0 Byte 1							
Switchpoint 0 Byte 2							
Switchpoint 0 Byte 3							
Switchpoint 1 Byte 1							
Switchpoint 1 Byte 2							
Switchpoint 1 Byte 3							
…							
Switchpoint 8 Byte 1							
Switchpoint 8 Byte 2							
Switchpoint 8 Byte 3							

Weekday (3 bits)

The possible values are:

Binary	Decimal	Description
0b000	0	Reserved
0b001	1	Monday
0b010	2	Tuesday
0b011	3	Wednesday
0b100	4	Thursday
0b101	5	Friday
0b110	6	Saturday
0b111	7	Sunday

Switchpoint (24 bits)

7	6	5	4	3	2	1	0	
Reserved			Hour					Byte 1
Reserved			Minute						Byte 2
Schedule State								Byte 3

CONFIDENTIAL

Hour (5 bits):

Specifies the hour during a day when this switchpoint SHALL be used. Possible values are:

Binary	Decimal	Description
0b00000	0	0 hour
0b00001	1	1st hour
0b00010	2	2nd hour
…	…	…
0b10111	23	23rd hour
0b11000	24	Reserved
…	…	Reserved
0b11111	31	Reserved

Minute (6 bits):

Specifies the minute during the specified hour when this switchpoint SHALL be used. Possible values are:

Binary	Decimal	Description
0b000000	0	0 minute
0b000001	1	1st minute
0b000010	2	2nd minute
…	…	…
0b111011	59	59th minute
0b111100	60	Reserved
…	…	Reserved
0b111111	63	Reserved

Schedule State (8 bits):

Schedule State uses the Schedule State type format, see section 0.
If Schedule State has the value of "Unused", then the Hour and Minute field SHALL be ignored. Once a Schedule State of "Unused" is encountered, the parsing of switchpoints SHALL stop.

4.21.4 Schedule Changed Get Command

The Schedule Changed Get Command used to check if the climate control schedule has changed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_CHANGED_GET							

CONFIDENTIAL

4.21.5 Schedule Changed Report Command

The Schedule Changed Report Command used to report if the climate control schedule has changed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_CHANGED_REPORT							
ChangeCounter							

ChangeCounter (8 bits)

The ChangeCounter is a timestamp for a climate control schedule and it is kept in devices which exchange climate control schedules.

One device holds a climate control schedule and other devices uses this climate control schedule. Whenever the climate control schedule changes, the device which holds it SHALL internally update its ChangeCounter, and the other devices SHALL regularly use the Schedule Changed Get Command on the device which holds the Climate Control Schedule to see if the ChangeCounter is different from last time – indicating a change in a climate control schedule.

The possible values are:

Hexadecimal	Decimal	Description
0x00	0	The climate control schedule change mechanism is temporarily disabled by the override function.
0x01 … 0xFF	1 - 255	The climate control schedule change mechanism is enabled. ChangeCounter is incremented by one every time climate control schedule changes. When ChangeCounter eventually reach 0xFF, then the next increment, will rollover to 0x01.

When a device is fresh and has no climate control schedule it SHALL retrieve a climate control schedule using the Schedule Get Command and it SHALL also use the Schedule Changed Get to get the first copy of the current ChangeCounter, thus avoiding getting the climate control schedule initially twice.

When a device is awake after sleep mode it SHOULD use this Command to detect if the schedule has been changed.

CONFIDENTIAL

4.21.6 Schedule Override Set Command

The Schedule Override Set Command used to set the override in a device.

The purpose of an override is to inform a device to ignore its current climate control schedule and assume the setting provided by the Override Type and Override State fields.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_OVERRIDE_SET							
Reserved						Override Type	
Override State							

Override Type (2 bits)

The override type field can assume the following values:

Binary	Decimal	Description
0b00	0	No override
0b01	1	Temporary override
0b10	2	Permanent override
0b11	3	Reserved

Note: The difference between a temporary and a permanent override is that a temporary override only overrides the current switchpoint in the climate control schedule.

Both temporary and permanent overrides MAY be cancelled in the device, which receives the SCHEDULE_OVERRIDE_SET. This cancellation SHALL be notified in an unsolicited SCHEDULE_OVERRIDE_REPORT as specified in the following sequence diagram:
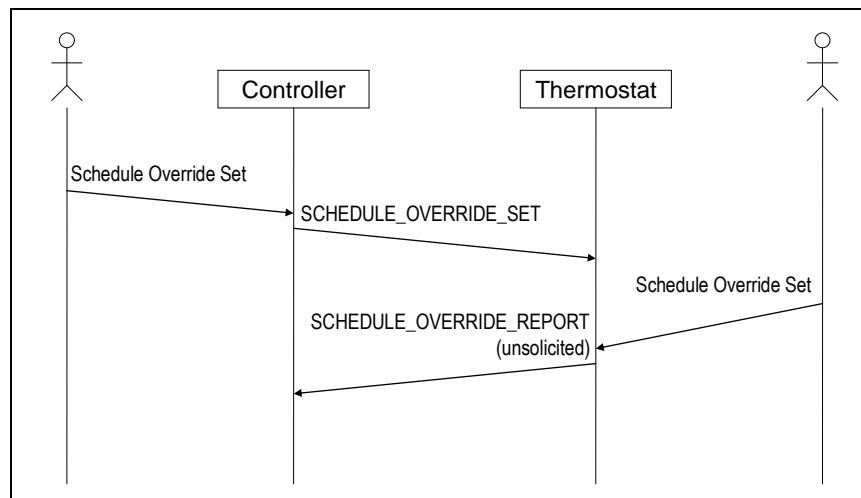
Figure 7, Sequence diagram for cancellation of a Schedule Override Set

CONFIDENTIAL

Override State (8 bits)

The Override State uses the Schedule State type format, see section 0

4.21.7   Schedule Override Get Command

The Schedule Override Get Command is used to request the override, currently in use in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_OVERRIDE_GET							

4.21.8   Schedule Override Report Command

The Schedule Override Report Command used to report the override, currently in use in a device. Application can send unsolicited Schedule Override Report Command or because of receiving a Schedule Override Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLIMATE_CONTROL_SCHEDULE							
Command = SCHEDULE_OVERRIDE_REPORT							
Reserved						Override Type	
Override State							

Override Type (2 bits)

The override type field can assume the following values:

Binary	Decimal	Description
0b00	0	No override
0b01	1	Temporary override
0b10	2	Permanent override
0b11	3	Reserved

Note: The difference between a temporary and a permanent override is that a temporary override only overrides the current switchpoint in the climate control schedule.

Both temporary and permanent overrides MAY be cancelled in the device, which receives the SCHEDULE_OVERRIDE_SET. This cancellation SHALL be notified in an unsolicited SCHEDULE_OVERRIDE_REPORT as shown on figure in section 4.21.6

Override State (8 bits)

The Override State uses the Schedule State type format, see section 0

CONFIDENTIAL

4.22   Central Scene Command Class, version 1

The Central Scene Command Class used to communicate central scene activations to a central controller using the lifeline concept. The central scene controller only need to configure lifeline association in relevant nodes before the home control application can take action in the central scene controller. The typical application contains up to 10-15 selected nodes creating a nice "out of the box" experience with minimum effort by the customer. A scene is typically activated via a push button on the device in question.

The Central Scene Command Class can instruct the central scene controller to perform the relevant actions as shown on figure below.
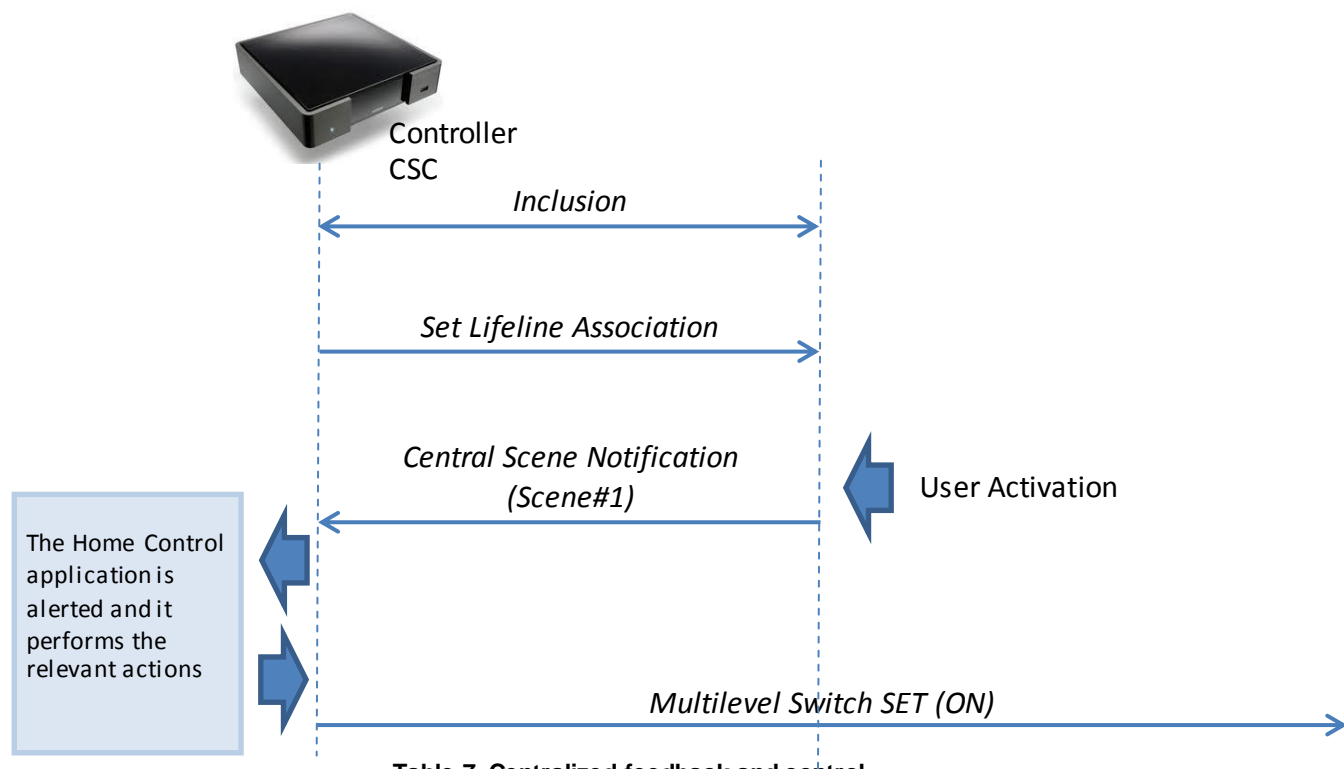
Table 7, Centralized feedback and control

Notice that this configuration has a single point of failure in case the central scene controller is broken. The likelihood to experience the "popcorn effect" has also increase because the central scene controller can be located far from the push button activated and therefore out of direct range with respect to devices to control.

To compensate for single point of failure in case the central scene controller is broken the devices could do fallback to alternative association groups to control devices directly. These association groups are activated only if the device does not receive acknowledge from configured lifeline. This will ensure some basic level of light control in the house until the central scene controller is restored. Alternatively, a backup central scene controller could be added to the system to take over in case the first one fails.

CONFIDENTIAL

4.22.1 Central Scene Supported Get Command

The Central Scene Supported Get Command is used to request the maximum number of scenes that this device supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_SUPPORTED_GET							

4.22.2 Central Scene Supported Report Command

The Central Scene Supported Report Command is used to report the maximum number of scenes that the requested device supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_SUPPORTED_REPORT							
Supported Scenes							

Supported Central Scenes (8 bits)

This field indicates the maximum number of scenes supported by the requested device.

CONFIDENTIAL

4.22.3 Central Scene Notification Command

The Central Scene Notification Command ,if enabled, is used to report activated scene on device in question including how notification must be interpreted.

In case a lifeline is configured in group #1 using Association Command Class then device must use this lifeline for transmitting the Central Scene Notification Command. It is allowed to define other association groups handling Central Scene Notification Command and setup rules between the groups. However, lifeline has precedence over other groups handling Central Scene Notification Command in case lifeline is defined.

NOTICE: The Central Scene Notification Command is by default enabled when device supporting it is included into a network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CENTRAL_SCENE							
Command = CENTRAL_SCENE_NOTIFICATION							
Sequence Number							
Reserved					Key Attributes		
Scene Number							

Sequence Number (8 bit)

The sequence number is incremented each time a Central Scene Notification Command is issued. The receiving device uses the sequence number to ignore duplicates.

CONFIDENTIAL

Key Attributes (3 bits)

The key attributes specifies the state of the key. Currently the following key attribute definitions exist:

Key Attribute	Description
0x00	Key Pressed –When a new key is pressed then start a 100ms timeout. In case key is released before timeout expires then send Central Scene Notification Command having Key Attribute = 0x00. Typically used to switch light on/off. See also description for Key Attribute = 0x02.
0x01	Key Released – Send Central Scene Notification Command having Key Attribute = 0x01 when the key is released but only as a termination of a Key Held Down sequence. Do not send command when previous event was a Key Pressed. Only the sequence number and key attribute parameter is changed in the Command. Typically used to stop dimming of light.
0x02	Key Held Down – When a new key is pressed then start a timer having a typical timeout equal to 100ms (max. timeout is 200ms). In case key is not released before timeout expires then send Central Scene Notification Command having Key Attribute = 0x02. Send Central Scene Notification Command having Key Attribute = 0x02 every max. 200ms while the key is still held down. Event used to signal continuation of key held down operation. After first received Key Held Down event will receiving device interpret a missing Key Held Down event within 400ms as a Key Released operation. Only the sequence number is changed in the Command. Typically used to dimming light up/down.

This list may evolve in the future. In case a key attribute is not supported then it should be ignored.

Reserved (5 bits)

The reserved field is for future use. The implementation shall zero these fields and shall make no assumptions on the values of these fields nor perform processing based on their content.

Scene Number (8 bits)

This field indicates the scene activated typically via a push button. It is possible to differentiate each scene by a unique number counting from 1. Upper left push button on device must be 1 numbering them from left to right and top to button (see example below).

CONFIDENTIAL
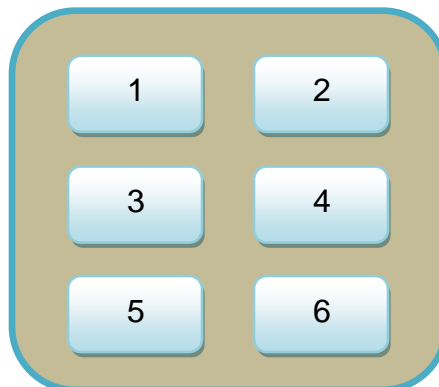
4.23 Clock Command Class, version 1

The Clock Command Class can be used to implement a simple clock functionality that can be used to displaying time or creating timers.

4.23.1 Clock Set Command

The Clock Set Command used to set the clock in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLOCK							
Command = CLOCK_SET							
Weekday			Hour				
Minute							

Weekday (3 bits)

The weekday field can take the following values 0 = Unused (24 hour clock), 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday and 7 = Sunday.

Hour (5 bits)

The hour field can take values from 0 to 23.

Minute (8 bits)

The minute field can take values from 0 to 59.

4.23.2 Clock Get Command

The Clock Get Command is used to request the clock report from a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLOCK							
Command = CLOCK_GET							

4.23.3 Clock Report Command

The Clock Report Command used to report the actual weekday and clock in a device. Application can send unsolicited Clock Report Commands or because of receiving a Clock Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CLOCK							
Command = CLOCK_REPORT							
Weekday			Hour				
Minute							

Weekday (3 bits)

The weekday field can take the following values 0 = Unused (24 hour clock), 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday, 5 = Friday, 6 = Saturday and 7 = Sunday.

Hour (5 bits)

The hour field can take values from 0 to 23.

Minute (8 bits)

The minute field can take values from 0 to 59.

CONFIDENTIAL

4.24   Color Control Command Class, version 1

The Color Control Command Class can be used to control the color of a lighting device. These commands allow applications to fully control a simple or advanced lighting device including RGB/CMY color mixing, CCT Color temperature and color.

For classic lighting intensity control, the Binary Switch Command Class or Multilevel Command Class MUST be used.

4.24.1   Color Control Capability Get

The Color Control Capability Get Command is used to request the color capability of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_COLOR_CONTROL							
Command = CAPABILITY_GET							

4.24.2   Color Control Capability Report

The Color Control Capability Report Command is used to report the color capability of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_COLOR_CONTROL							
Command = CAPABILITY_REPORT							
Capability mask 0							
Capability mask 1							

Capability mask 1..0 (16 bit)

The field specifies which functions is supported

Capability mask 0 – bit0 asserted: Capability ID 0 is supported
Capability mask 0 – bit1 asserted: Capability ID 1 is supported
….
Capability mask 1 – bit7 asserted: Capability ID 15 is supported

For more info regarding the capability IDs please refer to Color Control State SET command

CONFIDENTIAL

4.24.3 Color Control State Get

The Color Control State Get Command is used to request the color state of a device

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_COLOR_CONTROL							
Command = STATE_GET							
Capability ID							

Capability ID (8 bit)

The field specifies the Capability ID which current state is requested

For more info please refer to Color Control State SET command

4.24.4 Color Control State Report

The Color Control State Report Command is used to report the color state of a device

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_COLOR_CONTROL							
Command = STATE_REPORT							
Capability ID							
state							

Capability ID (8 bit)

The field specifies the Capability ID which current state is requested

State (8 bit)

The field specifies the current State of the Capability ID specified

CONFIDENTIAL

4.24.5 Color Control State Set

The Color Control State Set Command is used to control one or more capabilities in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_COLOR_CONTROL							
Command = STATE_SET							
Reserved				State data Length			
Capability ID							
state							
…….							
Capability ID							
state							

State data Length (5bit)

The field specifies the length of the appended (Capability ID, State) datasets.

Capability ID (8 bit)

The field specifies the Capability ID which current state is requested

State (8 bit)

The field specifies the current Level of the Capability ID specified

CONFIDENTIAL

Capability ID	Label	State
0	Warm White	0x00 – 0xFF: 0 – 100%
1	Cold White	0x00: - 0xFF: 0 – 100%
2	Red	0x00 – 0xFF: 0 – 100%
3	Green	0x00 – 0xFF: 0 – 100%
4	Blue	0x00 – 0xFF: 0 – 100%
5	Amber (for 6ch Color mixing)	0x00 – 0xFF: 0 – 100%
6	Cyan (for 6ch Color mixing)	0x00 – 0xFF: 0 – 100%
7	Purple (for 6ch Color mixing)	0x00 – 0xFF: 0 – 100%
8	Indexed Color	0x00 – 0x0FF: Color Index 0-255
9 - F	Reserved	

4.24.6   Start Capability State Change Command

The Start Capability State Change Command can change a color control capability. The speed that the product increases or decreases the state is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_COLOR_CONTROL							
Command = START_CAPABILITY_LEVEL_CHANGE							
Res	Up/ Down	Ignore Start State	Res				
Capability ID							
Start State							

Up/Down (1 bit)

If the Up/Down bit is set to 0 the capability SHALL increase its state. If field is set to 1 the switch SHALL decrease its state.

Ignore Start State (1 bit)

An Ignore Start state bit set to 0 indicates to use the start state specified in the Command. An Ignore Start state bit set to 1 indicates to start from the actual state in the device.

Res (1bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Start State (8 bits)

The Start State field contains the initial state for the Capability to assume when starting to change the state.

Capability ID (8 bits)

Please refer to Color Control State Set command.

4.24.7   Stop Capability State Change Command

The Stop Capability State Change Command is used to instruct a product to stop changing the state.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = STOP_STATE_CHANGE							
Capability ID							

CONFIDENTIAL

4.25 Configuration Command Class, version 1

With the Configuration Command Class it's possible to change the default factory settings in a device. This could for example be the dimming rate in a lighting dimmer device. When implementing this class in a controller it's RECOMMENDED to be able to set all parameters manually. Since the content of the configuration parameters are not standardized in the Z-Wave framework, it's the vendors responsibility to document this functionality in the products user manual (or an installer manual).

NOTE: All Z-Wave enabled devices MUST be able to operate based on the default factory setting.

4.25.1 Configuration Set Command

The Configuration Set Command used to set the value of configuration parameter(s).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_SET							
Parameter Number							
Default	Reserved					Size		
Configuration Value 1							
Configuration Value 2							
…							
Configuration Value N							

Parameter Number (8 bits)

The parameter number field specifies which configuration parameter is being set. The parameter numbers are specific for each application. It is RECOMMENDED that parameter number values are a sequence starting from 1.

Default (1 bit)

If the default bit is set to 1 the device is set to default factory setting and the configuration values is ignored. If the default bit is set to 0 then the configuration values is used.

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Size (3 bits)

The size field indicates the number of bytes used for the configuration value. This field can take values 1 (001b), 2 (010b) or 4 (100b).

CONFIDENTIAL

Configuration Value 1 … Configuration Value N (variable)

The configuration value is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal	Signed 2 bytes decimal value	Hexadecimal
127	0x7F	32767	0x7FFF
25	0x19	1025	0x0401
2	0x02	2	0x0002
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-2	0xFE	-2	0xFFFE
-25	0xE7	-1025	0xFBFF
-128	0x80	-32768	0x8000

4.25.2 Configuration Get Command

This Configuration Get Command used to get the value of a configuration parameter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_GET							
Parameter Number							

Parameter Number (8 bits)

The parameter number field specifies which configuration parameter is being requested. The parameter numbers are specific for each application. It is RECOMMENDED that parameter number values are a sequence starting from 1.

CONFIDENTIAL

4.25.3 Configuration Report Command

This Configuration Report Command used to report the actual value of a given configuration parameter in the device. Application can send unsolicited Configuration Report Commands or because of receiving a Configuration Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_REPORT							
Parameter Number							
Reserved					Size		
Configuration Value 1							
Configuration Value 2							
..							
Configuration Value n							

Refer to explanation under the Configuration Set Command.

CONFIDENTIAL

4.26 Configuration Command Class, version 2

Configuration Command Class v2 enables the device to exchange up to 65.535 product specific configuration parameters in the Z-Wave Interoperability community. In addition it is possible to set multiple configuration parameters with one Command.

It is mandatory for Z-Wave™ devices supporting Configuration Command Class v2 to also support Configuration Command Class v1; in order to communicate with Z-Wave™ enabled devices that only support v1. Therefore Z-Wave™ devices advertising Configuration Command Class v2 in the NIF SHALL implicit support Configuration Command Class v1.

NOTE! All Z-Wave™ enabled devices MUST be able to operate based on default factory settings. Supported configuration parameters MUST be a sequence starting from one (1).

4.26.1 Configuration Bulk Set Command

The Configuration Bulk Set Command used to set the value of configuration parameter(s).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_BULK_SET							
Parameter Offset MSB							
Parameter Offset LSB							
Number of Parameters							
Default	Hand-shake	Reserved				Size	
Parameter 1 – Configuration Value 1 (MSB)							
…							
Parameter 1 – Configuration Value N (LSB)							
…							
Parameter N – Configuration Value 1 (MSB)							
…							
Parameter N – Configuration Value N (LSB)							

Parameter Offset MSB + LSB (16 bits)

The parameter offset is a 16 bit value to address the offset to a parameter. E.g. if parameter offset = 900, then depended on the parameter number field, the first parameter to be addressed is 900.

Number of Parameters (8 bits)

CONFIDENTIAL

The Number of Parameters field specifies the number of configuration parameters that will be addressed with reference to the parameter offset. Valid values are 1-255 e.g. Number of Parameters = 3 and Parameter Offset = 900 then parameters 900, 901 and 902 are being addressed.

Default (1 bit)

If the default bit is set to 1 the device i.e. all configuration parameters are set to default factory settings and the parameter values are ignored. If the default bit is set to 0 then the configuration values are used. Note that when REQUIRED the configuration values can be overwritten by default values at any time.

Handshake (1 bit)

If the Handshake bit is set to 1 the receiving device MUST reply with a Configuration Bulk Report containing the same configuration parameter numbers (Parameter Offset + Number of Parameters), reporting that data has been stored in non-volatile memory for the requested configuration parameters and the receiver is now ready for next Command. In case the Configuration Bulk Report is missing despite successful transmission of Configuration Bulk Set, the transmitter MUST wait one second before sending the next Configuration Bulk Set Command.
If the Handshake bit is set to 0, then no Configuration Bulk Report is requested immediately after the Set Command.

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Size (3 bits)

The size field indicates the number of bytes that are used for the configuration value. This field can take values 1 (001b), 2 (010b) or 4 (100b).

Parameter 1 – Configuration Value 1 … Parameter N – Configuration Value N (variable)

The parameter[x] – data[x] is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Decimal	Hexadecimal	Decimal	Hexadecimal
127	0x7F	32767	0x7FFF
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-128	0x80	-32768	0x8000

CONFIDENTIAL

4.26.2 Configuration Bulk Get Command

The Configuration Bulk Get Command used to get the value of configuration parameter(s).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_BULK_GET							
Parameter Offset MSB							
Parameter Offset LSB							
Number of Parameters							

Parameter Offset MSB + LSB (16 bits)

See Configuration Bulk Set Command description.

Number of Parameters (8 bits)

See Configuration Bulk Set Command description.

4.26.3 Configuration Bulk Report Command

The Configuration Bulk Report Command used to report the actual value of the requested configuration parameter(s). The Configuration Bulk Report Command can be send as a result of receiving a Configuration Bulk Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONFIGURATION							
Command = CONFIGURATION_BULK_REPORT							
Parameter Offset MSB							
Parameter Offset LSB							
Number of Parameters							
Reports to follow							
Default	Hand-shake	Reserved			Size		
Parameter 1 – Configuration Value 1 (MSB)							
…							
Parameter 1 – Configuration Value N (LSB)							
…							
Parameter N – Configuration Value 1 (MSB)							
…							
Parameter N – Configuration Value N (LSB)							

Parameter Offset MSB + LSB (16 bits)

The Parameter Offset is a 16 bit value to address the offset to a parameter. E.g. if Parameter Offset = 900, then depended on the Number of Parameters field, the first parameter in the report is 900.

Number of Parameters (8 bits)

The Number of Parameters field specifies the number of configuration parameters that are presented in this report with reference to the Parameter Offset. Valid values are 1-255 e.g. if Number of Parameters = 2 and Parameter Offset = 900, then Configuration Parameter 900 and 901 are being reported.

Report to follow (8 bits)

This value indicates how many report frames there are left before all requested configuration parameters values have been transferred. 0 indicates no more Configuration Bulk Reports to follow.

Default (1 bit)

If default parameter is 1, then all returned configuration parameter values are in factory default state.

Handshake (1 bit)

If Handshake is 1, then this report indicates the receiver is ready for next frame.

Size (3 bits)

The size field indicates the number of bytes that are used for the configuration value. This field can take values 1 (001b), 2 (010b) or 4 (100b). All parameters in the same report MUST be of the same size.

Parameter 1 – Configuration Value 1 … Parameter N – Configuration Value N (variable)

See Configuration Bulk Set Command description.

CONFIDENTIAL
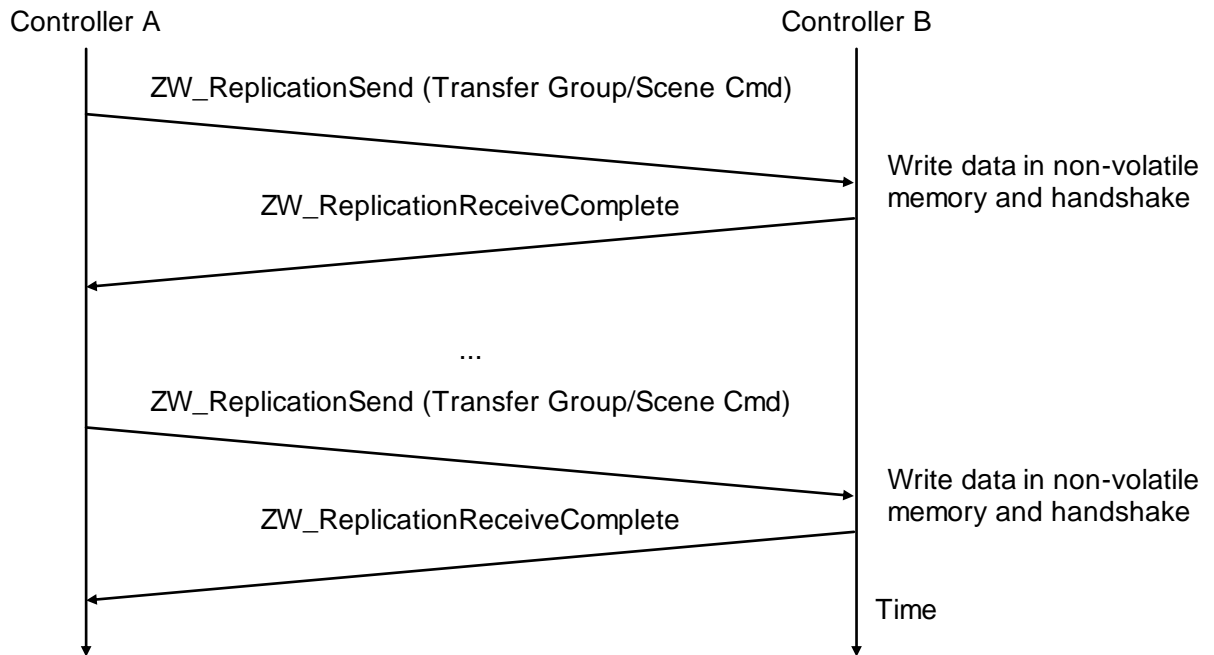
4.27 Controller Replication Command Class, version 1

The Controller Replication Command Class contains Commands that can be used to copy scene and group data to another controller. The Command Class MUST only be used in conjunction with a controller shift or when including a new controller to the network. It's OPTIONAL to use this command class during a controller shift or when including a new controller to the network. The API call ZW_ControllerChange used in a controller shift and ZW_AddNodeToNetwork when including a new controller to the network [1].

Devices supporting this Command Class SHOULD accept all the Commands. If some Commands are not used in the particular implementation, then they SHOULD be ignored.

The API call ZW_ReplicationSend MUST be used by the sending controller when transferring the group and scene command classes to another controller. The API call ZW_ReplicationReceiveComplete MUST be used by the receiving controller as acknowledge on application level because the data MUST first be stored in non-volatile memory before it can receive the next group or scene data.

Figure 8, Controller Replication sequence

A controller not supporting the Controller Replication Command Class MUST implement the acknowledge on application level when receiving Controller Replication Commands to avoid that the sending controller is locked due to a missing acknowledge on application level. The receiving controller will then ignore the content of the Controller Replication Commands but acknowledge on application level using the API call ZW_ReplicationReceiveComplete.

4.27.1 Transfer Group Command

The Transfer Group Command used to replicate mappings between Group ID and Node ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONTROLLER_REPLICATION							
Command = CTRL_REPLICATION_TRANSFER_GROUP							
Sequence Number							
Group ID							
Node ID							

Sequence Number (8 bits)

The sequence number used by the Z-Wave protocol and SHOULD NOT be filled in by the sending device or evaluated by the receiving device.

Group ID (8 bits)

Group ID of the group that the node is member of.

Node ID (8 bits)

Node ID of slave device.

4.27.2 Transfer Group Name Command

The Transfer Group Name Command used to replicate group names.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONTROLLER_REPLICATION							
Command = CTRL_REPLICATION_TRANSFER_GROUP_NAME							
Sequence Number							
Group ID							
Group Name 1							
Group Name 2							
Group Name 3							
…							
Group Name n							

Sequence Number (8 bits)

The sequence number used by the Z-Wave protocol and SHOULD NOT be filled in by the sending device or evaluated by the receiving device.

Group ID (8 bits)

Group ID associated with a specific group.

Group Name (n bytes)

The Group Name fields contain the assign group name in ASCII characters.

CONFIDENTIAL

4.27.3 Transfer Scene Command

The Transfer Scene Command used to replicate mappings between Scene ID and Node ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONTROLLER_REPLICATION							
Command = CTRL_REPLICATION_TRANSFER_SCENE							
Sequence Number							
Scene ID							
Node ID							
Level							

Sequence Number (8 bits)

The sequence number used by the Z-Wave protocol and SHOULD NOT be filled in by the sending device or evaluated by the receiving device.

Scene ID (8 bits)

The scene ID is the parameter used to link together the different devices that takes part of a scene.

Node ID (8 bits)

The Node ID for a device that is part of the scene.

Level (8 bits)

The level is the parameter used for the specified scene.

CONFIDENTIAL

4.27.4 Transfer Scene Name Command

The Transfer Scene Name Command used to replicate scene names.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CONTROLLER_REPLICATION							
Command = CTRL_REPLICATION_TRANSFER_SCENE_NAME							
Sequence Number							
Scene ID							
Scene Name 1							
Scene Name 2							
Scene Name 3							
…							
Scene Name n							

Sequence Number (8 bits)

The sequence number used by the Z-Wave protocol and SHOULD NOT be filled in by the sending device or evaluated by the receiving device.

Scene ID (8 bits)

Scene ID associated with a specific scene.

Scene Name (n bytes)

The Scene Name fields contain the assign scene name in ASCII characters.

CONFIDENTIAL

4.28   CRC-16 Encapsulation Command Class, version 1

The CRC-16 Encapsulation Command Class is used to encapsulate a command with an additional CRC-16 checksum to ensure integrity of the payload. The purpose for this command class is to ensure a higher integrity level of payloads carrying important data using 9.6/40kbps communication, in case the LRC checksum (8 bit) provided on protocol level is not sufficient to ensure integrity.

Notice: The CRC-16 Encapsulation Command Class MUST NOT be encapsulated by Multi Command Command Class, Security Command Class, and CRC-16 Encapsulation Command Class.

Alternatives to using CRC-16 Encapsulation Command Class could be:

- Z-Wave Security solution to ensure privacy and integrity of data.

- A pure 100kbps solution that already provides a CRC-16 checksum solution on protocol level.

Devices supporting CRC-16 Encapsulation Command Class

A device that support the CRC-16 Encapsulation Command Class MAY receive a combination of encapsulated and normal non-encapsulated requests and the correct response SHOULD be as follows:

a) If the request is sent encapsulated, the response MUST be returned encapsulated.

b) If the request is sent as a normal frame, i.e. non-encapsulated, the response MUST be sent non-encapsulated. Only if c) below is observed the response can be send encapsulated.

c) Before any setting, request, or response is send encapsulated to any other device, the sending device MUST ensure, that the destination will be able to understand the commands involved. This applies also to unsolicited commands.

A device that supports the CRC-16 Encapsulation Command Class MUST be able to receive and interpret the encapsulated version of all the command classes that it list in the NIF.

Devices controlling devices supporting CRC-16 Encapsulation Command Class

A device MUST request the NIF [1] from the destination and ensuring that the CRC-16 Encapsulation Command Class is listed as "supported" before sending request commands. Requesting NIF every time a CRC-16 Encap. CC is issued MUST be avoided; instead support of CRC-16 Encap. CC SHOULD be stored in requesting device to avoid traffic overhead.

For clarification it SHOULD be emphasized that a part of implementing the CRC-16 Encapsulation Command Class as "controller" the device MUST also be able to decode the encapsulated responses that MAY come back as a result of sending encapsulated request (e.g. get command to retrieve a report).

4.28.1 CRC-16 Encapsulated Command

The CRC-16 Encapsulation Command is used to encapsulate a command with an additional checksum to ensure integrity of the payload. Be aware of the payload limitations with respect to a routed single cast frame [1].

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_CRC_16_ENCAP							
Command = CRC_16_ENCAP							
Command Class							
Command							
Data 1							
…							
Data N							
Checksum 1							
Checksum 2							

Command Class (8 bits)

The Command Class field indicates the command class identifier of the encapsulated Command.

Command (8 bits)

The Command field indicates the Command identifier of the encapsulated Command.

Data 1 .. Data N (N x 8 bits)

The Data fields in the encapsulated Command of the respective command class and Command identifiers.

Checksum (16 bits)

The checksum field used to ensure consistency of the command class identifier, command identifier, report number and firmware data downloaded. The checksum is derived from the bytes starting with the command class identifier = COMMAND_CLASS_CRC_16_ENCAP and until the last data field (Data N). The first byte Checksum 1 is the most significant byte. The checksum algorithm implements a CRC-CCITT using initialization value equal to 0x1D0F and 0x1021 (normal representation) as the poly. Refer to Appendix C with respect to CRC_CCITT source code.

CONFIDENTIAL

4.28.2 Example

The BASIC GET command encapsulated in CRC-16 Encapsulation Command has the following format:

```
CRC-16 Encapsulation Command Class = 0x56
CRC-16 Encapsulation Command = 0x01
Basic Command Class = 0x20
Basic Get Command = 0x02
CRC1 = 0x4D
CRC2 = 0x26
```

CONFIDENTIAL

4.29 Demand Control Plan Configuration Command Class, version 1

The Demand Control Plan Configuration Command Class allows Utility Suppliers to issue and manage a list of Demand Control Plan (DCP) events to the end consumer.

The DCP configuration commands are separated for the DCP monitoring commands in the Demand Control Plan Monitor Command Class, allowing the classes to be OPTIONAL supported at different Z-Wave security levels. (E.g. DCP monitoring commands could be supported in any device, while enabling a strict and certificate based security solution for the DCP configuration command class). Please refer to the Hybrid Security Command class for more details regarding Z-Wave security levels.

A DCP event contains information regarding criticality, products involved, requested reduction, time duration and if a certain rate (identified by a Demand Control Plan Rate ID- please refer to section 4.78) is associated with the event. When a DCP event is outdated, it's removed from the list. It is the utility supplier responsibility to prevent overflow of the list by query the number of free positions in the list before submitting a new DCP event to the list. Each DCP event is uniquely identified by a timestamp issued the Utility Supplier.

A DCP event MAY also include information, which enables devices not supporting this class to use in the Demand Control Plan through the Start & Stop Association Group functionality. The installer, the end user or Utility Supplier (remote management) configures the Association groups. During the configuration process the devices are selected and the association entries are created. These associations can additionally be configured with the specific Z-Wave commands (through the Association Command Configuration Command Class). If no Z-Wave commands are specified in the Associations groups, it is the responsibility of the device to issue the relevant commands based on Utility Supplier specific algorithms.

4.29.1 DCP List Supported Get Command

The DCP List Supported Get Command is used to request the total size of the DCP list along with the number of free entries in the list.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_CONFIG							
Command = DCP_LIST_SUPPORTED_GET							

4.29.2 DCP List Supported Report Command

The DCP List Supported Report Command used to provide the total size of the DCP list along with the number of free entries in the list.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_CONFIG							
Command = DCP_LIST_SUPPORTED_REPORT							
DCP List Size							
Free DCP List entries							

DCP List Size (8 bit)

This value specifies the DCP list size. 0x00 is reserved

Free DCP List entries (8 bit)

This value specifics the number of free entries for new DCP events. The value 0x00 specifies a full list.

4.29.3 DCP List Set Command

The DCP List Set Command used to place a new DCP event in the DCP list. Each DCP event is time stamped for future reference.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_CONFIG								
Command = DCP_LIST_SET								
Timestamp -Year 1								
Timestamp -Year 2								
Timestamp -Month								
Timestamp -Day								
Timestamp -Hour Local Time								
Timestamp -Minute Local Time								
Timestamp -Second Local Time								
DCP Rate ID								
Reserved					Number of DC		
Generic Device Class 1								
Specific Device Class 1								
….								
Generic Device Class N								
Specific Device Class N								
Start Year 1								
Start Year 2								
Start Month								
Start Day								
Start Hour Local Time								
Start Minute Local Time								
Start Second Local Time								
Duration Hour Time								
Duration Minute Time								
Duration Second Time								
Event Priority								
Load shedding								
Start Association Group								
Stop Association Group								
Randomization interval								

CONFIDENTIAL

Timestamp -Year 1..2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Timestamp -Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Timestamp -Day (8 bit)

Specify the day of the month between 01 and 31.

Timestamp -Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Timestamp -Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Timestamp -Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.
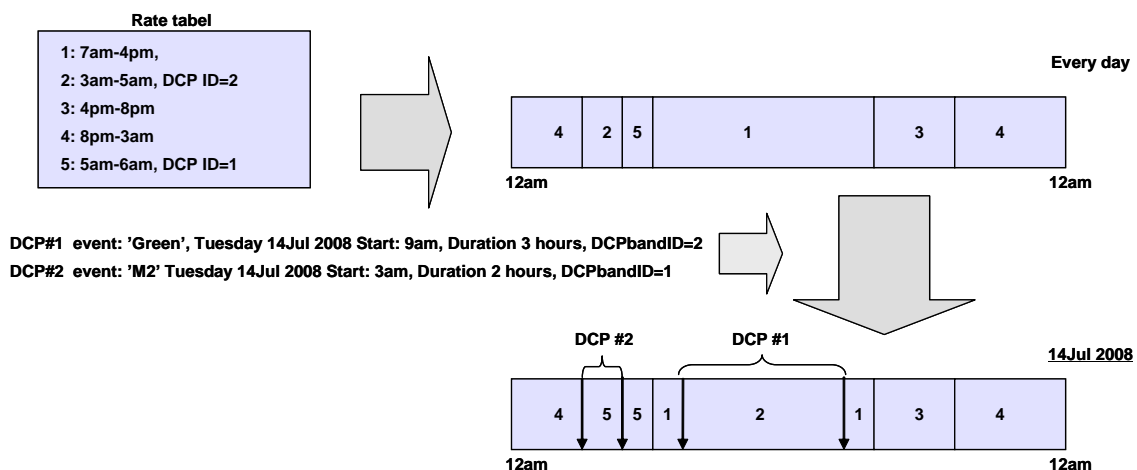
DCP Rate ID (8 bit)

Specify if a specific rate is applicable when participating in the DCP event. If an entry in the Rata table contains a matching DCP Rate ID this rate will be active for the duration of the event regardless of other parameters in the rate is not met.

<u>DCP Rate ID usage Example</u>

Prior to the DCP events a given rate table is configured defining when rates are active during a day. The Table contains two entries with DCP Rate IDs allowing the Utility Supplier to activate the rates outside of the time defined in the Table when a DCP event is placed in the DCP list with the corresponding DCP rate ID.

Prior to Jul14 2008 two DCP events are placed in the List by the Utility Supplier, which changes the Rate profile of Jul14 compared to an 'standard' day.

CONFIDENTIAL

Randomization interval (8 bit)

Specify the randomization interval in units of seconds, which MUST be applied as an offset to the start and stopping of the events as requested in the DCP event.

E.g. A value of 0x10 specifies that every device SHOULD randomly select a start and stop time offset between 0 and 16 seconds. This offset SHOULD be applied to the start and duration fields in the DCP event.

Number of DC (2 bit)

Specify the number of Generic/Specific Device Classes, which are requested to participate in the DCP event

Generic Device Class (8bit)

Specify the Generic Device Class identifier. Please refer to [1].

Specific Device Class (8 bit)

Specify the Specific Device Class identifier. Please refer to [1].

Start Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar for the start of the event. The first byte (Year 1) is the most significant byte.

Start Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December) for the start of the event.

Start Day (8 bit)

Specify the day of the month for the start of the event between 01 and 31.

Start Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time for the start of the event.

Start Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time for the start of the event.

Start Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

CONFIDENTIAL

Duration Hour Time (8 bit)

Specify the number of complete hours of the event.

Duration Minute Time (8 bit)

Specify the number of complete minutes of the event.

Duration Second Time (8 bit)

Specify the number of complete seconds of the event.

Event Priority (8 bit)

The parameter specifies the priority of the DCP event. The High priority is used by the utility Supplier to mandate device participation and lower priorities are used by devices to voluntary to participate in the event and to which degree.

Event Priority	Description	Device participation
0x00	Reserved	Voluntary
0x01	V1 - Green Energy	Voluntary
0x02	V2	Voluntary
0x03	V3	Voluntary
0x04	V4	Voluntary
0x05	V5	Voluntary
0x06	V6	Voluntary
0x07	V7	Voluntary
0x08	M1 - Emergency	Mandatory
0x09	M2	Mandatory
0x0A	M3	Mandatory
0x0B	M4	Mandatory
0x0C	Utility defined	Utility defined
0x0D	Utility defined	Utility defined
0x0E	Utility defined	Utility defined
0x0F	Utility defined	Utility defined
0x10-0xFF	Reserved for future use	Reserved for future use

Load shedding (8 bit)

Specify the load shedding in percentage of average consumption requested for the event. This load shedding will be applied to the devices with the device classes specified in the command.

E.g. 0x01 = 1%, 0x60 = 99%, 0x00 and 0x61-0xFF is reserved.

CONFIDENTIAL

Start Association group (8 bit)

Specify which association group (see the Association Command Class) SHOULD be activated when the event is started. The value 0x00 specifies no group SHOULD be activated.

If the device also supports the Association Command Configuration Command Class it possible to dynamically and remotely (from the Energy Supplier or others) to specify precisely which current and future Z-Wave commands SHOULD be send to which Z-Wave nodes.

The configuration of the Start Association group is done either manually by the installer or automatically when the device detect other relevant devices

Stop Association group (8 bit)

Specify which association group (see the Association Command Class) SHOULD be activated when the event is stopped. The value 0x00 specifies no group SHOULD be activated.

The configuration of the Stop Association group is done either manually by the installer or automatically when the device detect other relevant devices

Start Associating group and Stop Associating Group usage example

A small energy control system consisting of a device supporting the DCP command class and 3 z-wave devices which can participate in the application. NodeId 1: Setback Thermostat device, NodeId 8: Simple Thermostat device, NodeId 5: Multilevel Power Switch device.

Through the use of the Association Command Class and the Association Command Configuration Command Class the following Associations has been established in the device.

Group 1
Node1, Thermostat_Setback_SetPpermanent override,energy saving mode)
Node8, Thermostat_Setpoint_Set(Heating setpoint#1, 19,5^{o}C)
Node5, Multilevel_Switch:Set (Dimlevel =0x20)

Group 2
Node1, Thermostat_Setback_Set(No override,Setback = 0°C)
Node8, Thermostat_Setpoint_Set(Heating setpoint#1, 21,5^{o}C)
Node5, Multilevel_Switch:Set (Dimlevel =0x40)

A DCP event can now specifically invoke Group1 when starting the event and invoking group2 when stopping the event by specifying Start Association Group=0x01 and Stop Association Group=0x02.

4.29.4 DCP List Remove

The DCP List Remove Command used to remove a DCP event from the DCP list.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_CONFIG							
Command = DCP_LIST_REMOVE							
Timestamp -Year 1							
Timestamp -Year 2							
Timestamp -Month							
Timestamp -Day							
Timestamp -Hour Local Time							
Timestamp -Minute Local Time							
Timestamp -Second Local Time							

Please refer to DCP List report command class (section 4.30.2) for description of fields

CONFIDENTIAL

4.30   Demand Control Plan Monitor Command Class, version 1

The Demand Control Plan Monitor Command Class allows devices to monitor the list of Demand Control Plan (DCP). A DCP event contains information regarding criticality, products involved, requested reduction, time duration and if a certain rate (identified by a Demand Control Plan ID- please refer to section 4.78) is associated with the event. When a DCP event is outdated, it's removed from the list. Each DCP event is uniquely identified by a timestamp issued the Utility Supplier.

4.30.1   DCP List Get Command

The DCP List Get Command is used to request the pending DCP event in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_MONITOR							
Command = DCP_LIST_GET							

4.30.2   DCP List Report Command

The DCP List Report Command reports the pending DCP event in a device. If more than one DCP event is pending – the reports will be submitted in chronically order as to when the events was placed on the list. Newest entry will be reported first.

CONFIDENTIAL

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_MONITOR							
Command = DCP_LIST_REPORT							
Reports to Follow							
Timestamp -Year 1							
Timestamp -Year 2							
Timestamp -Month							
Timestamp -Day							
Timestamp -Hour Local Time							
Timestamp -Minute Local Time							
Timestamp -Second Local Time							
DCP ID							
Reserved					Number of DC		
Generic Device Class 1							
Specific Device Class 1							
….							
Generic Device Class N							
Specific Device Class N							
Start Year 1							
Start Year 2							
Start Month							
Start Day							
Start Hour Local Time							
Start Minute Local Time							
Start Second Local Time							
Duration Hour Time							
Duration Minute Time							
Duration Second Time							
Event Priority							
Load shedding							
Start Association Group							
Stop Association Group							
Randomization interval							

Reports to Follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

Please refer to DCP List Set command class (section 4.29.3) for detailed description of the fields.

4.30.3 DCP Event Status Get

The DCP Event Status Get Command used query the status of a specific DCP event in the DCP list.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_MONITOR							
Command = DCP_EVENT_STATUS_GET							
Timestamp -Year 1							
Timestamp -Year 2							
Timestamp -Month							
Timestamp -Day							
Timestamp -Hour Local Time							
Timestamp -Minute Local Time							
Timestamp -Second Local Time							

Please refer to DCP List Set command class (section 4.29.3) for detailed description of the fields.

4.30.4 DCP Event Status Report

The DCP Event Status Report Command used to provide the status of a specific DCP event in the DCP list.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DCP_MONITOR							
Command = DCP_EVENT_STATUS_GET							
Timestamp -Year 1							
Timestamp -Year 2							
Timestamp -Month							
Timestamp -Day							
Timestamp -Hour Local Time							
Timestamp -Minute Local Time							
Timestamp -Second Local Time							
Event status							

Please refer to DCP List report command class (section 4.30.2) for description of fields

CONFIDENTIAL

Event Status (8 bit)

The field contains the status of the event.

Event status	Description
0x00	Reserved
0x01	Event Started
0x02	Event Completed
0x03	Event Rejected by the user
0x04	Event not Applicable
0x05-0xFF	Reserved

CONFIDENTIAL

4.31   Device Reset Locally Command Class, version 1

The Device Reset Locally Command Class is used to notify central controllers that a Z-Wave device is being removed from a network by the use of the API call ZW_SetDefault. In Z-Wave it is no longer REQUIRED to have portable controllers and hence this command class has become mandatory for all Z-Wave devices. The Device Reset Locally Command Class is mandatory as it is the only way to notify other nodes that a device is reset.

4.31.1   Device Reset Locally Notification Command

The Device Reset Locally Notification Command is used to announce that a Z-Wave device will be set to default by calling API call ZW_SetDefault and thereby initializing protocol data in non-volatile memory. In addition, application MUST also initialize application data in non-volatile memory to the starting point when included into a new network. The non-volatile memory initialization is executed by either a push button locally on the Z-Wave device or a serial API command to the Z-Wave device in question. In case a lifeline is configured in group #1 using Association Group Information Command Class then Z-Wave device MUST try to send a Device Reset Locally Notification Command before calling API call ZW_SetDefault etc. It MUST be possible to reset a device repeatedly and skipping Device Reset Locally Notification Command in case node is not part of a network.

In case only selected nodes are reset the static controller receiving a Device Reset Locally Notification Command SHOULD afterwards call API call ZW_RemoveFailedNode for source node in command to get network topology updated accordingly. In addition, associations to the reset node MAY exist in other nodes, which will require appropriate re-configuration of the nodes in question to avoid addressing the reset node going forward.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DEVICE_RESET_LOCALLY							
Command = DEVICE_RESET_LOCALLY_NOTIFICATION							

4.32   Door Lock Command Class, version 1

The Door Lock Command Class used to secure/unsecure a lock type as well as setting the configuration of an advanced Z-Wave™ door lock device.

4.32.1   Door Lock Operation Set Command

The Door Lock Operation Set Command used to set the operation mode of the lock in a Z-Wave™ enabled door lock device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_OPERATION_SET							
Door Lock Mode							

Door Lock Mode (8 bits)

Door Lock Mode will set the door lock device in unsecured or secured mode as well as other peripheral settings.

Hexadecimal	Description
0x00	Door Unsecured [1]
0x01	Door Unsecured with timeout [2]
0x10	Door Unsecured for inside Door Handles [1]
0x11	Door Unsecured for inside Door Handles with timeout [2]
0x20	Door Unsecured for outside Door Handles [1]
0x21	Door Unsecured for outside Door Handles with timeout [2]
0xFF	Door Secured

1) Constant mode. Door will be unsecured until set back to secured mode by Command.
2) Timeout mode. Fallback to secured mode after timeout has expired (set by Door Lock Configuration Set).

4.32.2   Door Lock Operation Get Command

The Door Lock Operation Get Command is used to request the state of the door lock device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_OPERATION_GET							

4.32.3 Door Lock Operation Report

The Door Lock Operation Report Command can be used by a door lock device to send a report either unsolicited or requested by the Door Lock Operation Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_OPERATION_REPORT							
Door Lock Mode							
Outside Door Handles Mode				Inside Door Handles Mode			
Door Condition							
Lock Timeout Minutes							
Lock Timeout Seconds							

Door Lock Mode (8 bits)

See Door Lock Operation Set Command description. In case timeout mode is not supported, this field MUST display other valid mode from the Door Lock Operation Set Command e.g. 0x10 instead of 0x11.

Outside/Inside Door Handles Mode (4 bits)

These parameters indicate the activity of the door handles i.e. which handle(s) has opened the door lock.

Bit	Outside Door Handles Mode (4 bits)	Inside Door Handles Mode (4 bits)
0	0 = Handle 1 inactive; 1 = Handle 1 active	0 = Handle 1 inactive; 1 = Handle 1 active
1	0 = Handle 2 inactive; 1 = Handle 2 active	0 = Handle 2 inactive; 1 = Handle 2 active
2	0 = Handle 3 inactive; 1 = Handle 3 active	0 = Handle 3 inactive; 1 = Handle 3 active
3	0 = Handle 4 inactive; 1 = Handle 4 active	0 = Handle 4 inactive; 1 = Handle 4 active

Door Condition (8 bits)

The Door Condition field indicates the hardware status of the door lock device such as bolt and latch states.

Bit	Description
0	0 = Door Open; 1 = Door Closed
1	0 = Bolt Locked; 1 = Bolt Unlocked
2	0 = Latch Open; 1 = Latch Closed
3-7	Reserved

CONFIDENTIAL

Lock Timeout Minutes (8 bits)

If a Lock Timeout has been set in DOOR_LOCK_CONFIG_SET Command and the door lock state has been set to unsecure with timeout by means of DOOR_LOCK_OPERATION_SET, this field SHALL display the remaining minutes the lock is in unsecured state. In case no timeout has been set this field SHALL always display zero. This field SHALL be set to 0xFE when timeout is not supported by the door lock device.

Lock Timeout Seconds (8 bits)

If a Lock Timeout has been set in DOOR_LOCK_CONFIG_SET Command and the door lock state has been set to unsecure with timeout by means of DOOR_LOCK_OPERATION_SET, this field SHALL display the remaining seconds the lock is in unsecured state. In case no timeout has been set this field SHALL always display zero. This field SHALL be set to 0xFE when timeout is not supported by the door lock device.

4.32.4 Door Lock Configuration Set

The Door Lock Configuration Set Command used to set the configuration of the door lock device including operation type and operation timers.

Note: All Z-Wave enabled devices MUST be able to operate based on factory default settings i.e. an end-user SHOULD NOT be forced to setup the configuration parameters for the door lock to operate.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CONFIGURATION_SET							
Operation Type							
Outside Door Handles State				Inside Door Handles State			
Lock Timeout Minutes							
Lock Timeout Seconds							

Operation Type (1 byte)

The Operation Type field can be set to either constant or timed operation. When timed operation is set, the Lock Timer Minutes and Lock Timer Seconds fields MUST be set to valid values.

Hexadecimal	Description
0x01	Constant operation
0x02	Timed operation
0x03 – 0XFF	*Reserved*

CONFIDENTIAL

Outside/Inside Door Handles State (4 bits)

The Door Handles field is to enable or disable the door handlers that are implemented in the door lock device. For example there could be an inside as well as an outside door handler, whereas the inside door handler can be handled by Z-Wave Commands and the outside door handler will only unlock the door when successful authentication has been verified by e.g. a keypad.

Bit	Outside Door Handles State (4 bits)	Inside Door Handles State (4 bits)
0	0 = Handle 1 disabled; 1 = Handle 1 enabled	0 = Handle 1 disabled; 1 = Handle 1 enabled
1	0 = Handle 2 disabled; 1 = Handle 2 enabled	0 = Handle 2 disabled; 1 = Handle 2 enabled
2	0 = Handle 3 disabled; 1 = Handle 3 enabled	0 = Handle 3 disabled; 1 = Handle 3 enabled
3	0 = Handle 4 disabled; 1 = Handle 4 enabled	0 = Handle 4 disabled; 1 = Handle 4 enabled

Lock Timeout Minutes (1 byte)

Number of minutes the lock stays unsecured. If the Operation Type is set to Timed Operation, this field can be set accordingly to the valid values 1-254 decimal. If timeout is not supported for the door lock device then this SHALL be indicated with 0xFE. All other values not mentioned are reserved.

Lock Timeout Seconds (1 byte)

Number of seconds the lock stays unsecured. If the Operation Type is set to Timed Operation, this field can be set accordingly to the valid 1-59 decimal. If timeout is not supported for the door lock device then this SHALL be indicated with 0xFE. All other values not mentioned are reserved.

4.32.5 Door Lock Configuration Get Command

The Door Lock Configuration Get Command is used to request the configuration parameter of the door lock device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CONFIGURATION_GET							

CONFIDENTIAL

4.32.6 Door Lock Configuration Report Command

The Door Lock Configuration Report Command can be used by a door lock device send a report requested by the Door Lock Configuration Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK							
Command = DOOR_LOCK_CONFIGURATION_REPORT							
Operation Type							
Outside Door Handles State				Inside Door Handles State			
Lock Timeout Minutes							
Lock Timeout Seconds							

See parameter description for DOOR_LOCK_CONFIGURATION_SET.

4.33 Door Lock Logging Command Class, version 1

This Door Lock Logging Command Class provides an audit trail in an access control application. Each time an event takes place at the door lock, the system logs the user's ID, date, time etc.

4.33.1 Door Lock Logging Records Supported Get Command

The Door Lock Logging Records Supported Get Command is used to request the number of records that the audit trail supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK_LOGGING							
Command = DOOR_LOCK_LOGGING_RECORDS_SUPPORTED_GET							

4.33.2 Door Lock Logging Records Supported Report Command

The Door Lock Logging Records Supported Report Command is used to report the maximum number of reports the audit trail supports. The Door Lock Logging Records Supported Report Command can be requested by the Door Lock Logging Records Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK_LOGGING							
Command = DOOR_LOCK_LOGGING_RECORDS_SUPPORTED_REPORT							
Max records stored							

Max records stored (8 bit)

The number of records the audit trail supports stored in a queue.

4.33.3 Door Lock Logging Record Get Command

The Door Lock Logging Record Get Command is used to request the audit trail.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK_LOGGING							
Command = RECORD_GET							
Record number							

Record number (8 bit)

The Record number field indicates the record to be requested.

CONFIDENTIAL

A value 0 – *Max records stored* is acceptable with a value of 0 being the most recent entry. When requesting with a value of 0, the report will contain the record number so the latest record is known.

4.33.4   Door Lock Logging Record Report Command

The Door Lock Logging Record Report Command returns records from the audit trail. The Door Lock Logging Record Report Command can be send unsolicited or as a result of receiving a Door Lock Logging Record Get Command. To provide flexibility the user associated with the record can be identified by one of two methods. The user identifier field or the user code entered MUST be filled into the report when needed as dictated by the access status field.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_DOOR_LOCK_LOGGING							
Command = RECORD_REPORT							
Record number							
Timestamp - Year 1							
Timestamp - Year 2							
Timestamp – Month							
Timestamp – Day							
Record status			Timestamp - Hour Local Time				
Timestamp - Minute Local Time							
Timestamp - Second Local Time							
Event Type							
User Identifier							
User Code Length							
USER_CODE1							
..							
USER_CODEn							

Record number (8 bit)

Record number requested (1- 255).

Timestamp - Year 1..2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Timestamp - Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Timestamp - Day (8 bit)

Specify the day of the month between 01 and 31.

Record status (3-bits)

The record status field is used to indicate whether legal data is stored in the record.

Record State	Description
0	Requested record is empty.
1	Requested record holds legal data.

Timestamp - Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Timestamp - Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Timestamp - Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.
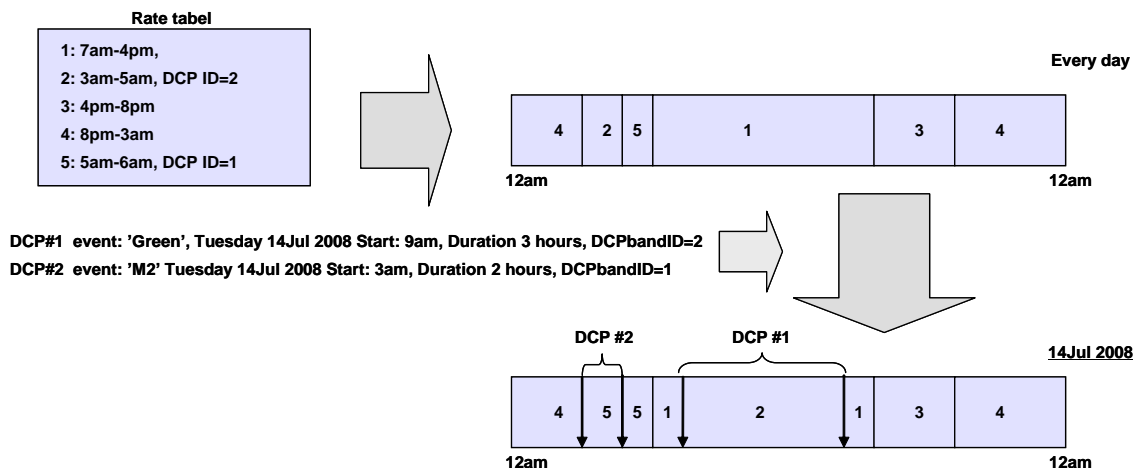
Note: If RTC (Real Time Clock) all values in Time Stamp record SHOULD be set to 0. Most recent Record SHALL be stored under Record Number 1.

Event Type (8 bit)

Following Events are supported:

Event type	Description
1	Lock Command: Keypad access code verified lock command
2	Unlock Command: Keypad access code verified unlock command
3	Lock Command: Keypad lock button pressed
4	Unlock command: Keypad unlock button pressed
5	Lock Command: Keypad access code out of schedule
6	Unlock Command: Keypad access code out of schedule
7	Keypad illegal access code entered
8	Key or latch operation locked (manual)
9	Key or latch operation unlocked (manual)
10	Auto lock operation
11	Auto unlock operation
12	Lock Command: Z-Wave access code verified
13	Unlock Command: Z-Wave access code verified
14	Lock Command: Z-Wave (no code)
15	Unlock Command: Z-Wave (no code)
16	Lock Command: Z-Wave access code out of schedule
17	Unlock Command Z-Wave access code out of schedule
18	Z-Wave illegal access code entered
19	Key or latch operation locked (manual)
20	Key or latch operation unlocked (manual)
21	Lock secured
22	Lock unsecured
23	User code added
24	User code deleted
25	All user codes deleted
26	Master code changed
27	User code changed
28	Lock reset
29	Configuration changed
30	Low battery
31	New Battery installed

Not all events types are supported and it is up to manufacturer to decide which one are to be supported.

CONFIDENTIAL

User Identifier (8 bits)
The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. A User Identifier of 0 is acceptable when the record does not need to identify a user or if the User Code is provided in this report.

User Code Length (8 bit)

The User Code Length field indicates the number of bytes used to hold the User Code. A length of 0 is acceptable when the record does not need to identify a User Code or when the User Identifier field is non-zero.

USER_CODE1…USER_CODEn

These fields contain the user code. Minimum code length is 4 and maximum 10 ASCII digits. For further details about the user code, refer to the User Code Command Class.

4.34 Energy Production Command Class, version 1

The Energy Production Command Class used to retrieve various production data from the device.

4.34.1 Energy Production Get Command

The Energy production Get Command is used to request various production data from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENERGY_PRODUCTION							
Command = ENERGY_PRODUCTION_GET							
Parameter Number							

Parameter Number (8 bits)

The parameter number specifies the kind of production data to retrieve. Currently the following parameter numbers are defined:

Parameter Number	Description
0x00	Instant energy production
0x01	Total energy production
0x02	Energy production today
0x03	Total production time

This list MAY evolve in the future. In case a parameter number is not supported then it SHOULD be ignored.

CONFIDENTIAL

4.34.2 Energy Production Report Command

The Energy Production Report used to retrieve various production data from the device. The Energy Production Report Command can be send requested by the Energy Production Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ENERGY_PRODUCTION							
Command = ENERGY_PRODUCTION_REPORT							
Parameter Number							
Precision			Scale		Size		
Value 1							
Value 2							
…							
Value N							

Parameter Number (8 bits)

Refer to description under the Energy Production Get Command.

Precision (3 bits)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bits)

The scale field indicates the scale used for the specified parameter number:

Parameter Number	Scale	Description
0x00	0x00	W
0x01	0x00	Wh
0x02	0x00	Wh
0x03	0x00	Seconds
	0x01	Hours

Size (3 bits)

The size field indicates the number of bytes used for the value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

Value 1 ... Value N (variable)

The value is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal	Signed 2 bytes decimal value	Hexadecimal
127	0x7F	32767	0x7FFF
25	0x19	1025	0x0401
2	0x02	2	0x0002
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-2	0xFE	-2	0xFFFE
-25	0xE7	-1025	0xFBFF
-128	0x80	-32768	0x8000

CONFIDENTIAL

4.35   Firmware Update Meta Data Command Class, version 1

The Firmware Update Meta Data Command Class used to update a device in the Z-Wave network with new firmware via the 40kbps RF communication link. It is RECOMMENDED to enable the firmware update by some kind of physical authentication (e.g. activation of a pushbutton) on the device in question to avoid unintentional firmware updates.

The Firmware Update Meta Data Command Class is actually not updating the firmware in the device, but only used to store the firmware data temporary in memory e.g. the external EEPROM. The device will now calculate the checksum of the downloaded firmware data and compare it with the checksum received. In case the checksum is successfully verified the application will jump to a bootloader taking care of erasing the Flash memory and transferring the firmware data to the Flash memory. Refer to [2] regarding how to erase/write to Flash. The bootloader MUST be able to reside in Flash during this process and is therefore not altered when the firmware is updated. When the bootloader is finished, it will reset the module and the newly downloaded firmware is then executed. The bootloader MUST be located at the top of the memory map but the start address will be vendor specific depending on the complexity of the bootloader. Bootloader MAY support recovery of protocol/application data when updating firmware. Notice that protocol/application data can change location in the memory map depending on the library/application used. The Version Command Class can be used to verify that the intended firmware version is installed. Device SHOULD beside Version Command Class also support Manufacturer Specific Command Class to enable other devices to select the correct firmware data for the device in question.

A bootloader solution can be avoided in case the Z-Wave Single Chip has an appropriate interface to a host processor. In a two chip scenario the host processor can be used as temporary storage of the firmware data received. Afterwards can the host processor program the Flash on the Z-Wave Single Chip via the SPI interface and thereby avoiding a bootloader on the Z-Wave Single Chip.

The API call ZW_SendDataMeta MUST be used when streaming data to ensure that this traffic doesn't prevent control data from getting through in the network, especially important for 9.6kbps nodes because they can't detect 40kbps RF communication. Refer to [1] regarding a detailed description of the API call ZW_SendDataMeta.

Notice that it is not allowed to update firmware in a ZW0102 based module because it only supports 9.6kbps.

4.35.1   Firmware Meta Data Get Command

The Firmware Meta Data Get Command is used to request the current firmware in the device. This call is typically used to check whether or not the firmware needs to be updated.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD								
Command = FIRMWARE_MD_GET								

CONFIDENTIAL

4.35.2 Firmware Meta Data Report Command

The Firmware Meta Data Report Command used to return status for the current firmware in the device. The Firmware Meta Data Report Command can only be sent requested by the Firmware Meta Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_MD							
Command = FIRMWARE_MD_REPORT							
Manufacturer ID 1							
Manufacturer ID 2							
Firmware ID 1							
Firmware ID 2							
Checksum 1							
Checksum 2							

Manufacturer ID (16 bits)

The Manufacturer ID is a unique ID identifying the manufacturer of the device. The Manufacturer ID is assigned by Sigma Designs upon request. Refer to 0 for a list of assigned Manufacturer ID's. The first byte is the most significant byte.

Firmware ID (16 bits)

The Firmware ID is a unique identification of the firmware file when combined with the Manufacturer ID. Return zero as Firmware ID, in case no value is stored. The manufacturer assigns the Firmware ID. The first byte is the most significant byte.

Checksum (16 bits)

The checksum field used to ensure consistency of the firmware data currently downloaded. Return zero as checksum, in case no value is stored. The first byte is the most significant byte. It is RECOMMENDED to use a checksum algorithm that implements a CRC-CCITT using initialization value equal to 0x1D0F and 0x1021 (normal representation) as the poly. Refer to Appendix C with respect to CRC_CCITT source code.

CONFIDENTIAL

4.35.3 Firmware Update Meta Data Request Get Command

The Firmware Update Meta Data Request Get Command is used to request a firmware update. It is RECOMMENDED to send the Command in conjunction with some kind of physical authentication on the device to be updated.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD							
Command = FIRMWARE_UPDATE_MD_REQUEST_GET							
Manufacturer ID 1							
Manufacturer ID 2							
Firmware ID 1							
Firmware ID 2							
Checksum 1							
Checksum 2							

Manufacturer ID (16 bits)

The Manufacturer ID is a unique ID identifying the manufacturer of the device. The Manufacturer ID is assigned by Sigma Designs upon request. Refer to 0 for a list of assigned Manufacturer ID's. The first byte is the most significant byte.

Firmware ID (16 bits)

The Firmware ID is a unique identification of the firmware file when combined with the Manufacturer ID. The manufacturer assigns the Firmware ID. The first byte is the most significant byte. Bootloader will check that a valid firmware file is available based on the Manufacturer ID and Firmware ID.

Checksum (16 bits)

The checksum field used to ensure consistency of the firmware data to be downloaded. It is RECOMMENDED to use a checksum algorithm that implements a CRC-CCITT using initialization value equal to 0x1D0F and 0x1021 (normal representation) as the poly. Refer to Appendix C with respect to CRC_CCITT source code.

4.35.4 Firmware Update Meta Data Request Report Command

The Firmware Update Meta Data Request Report Command used to return status for the requested firmware update. The Firmware Update Meta Data Request Report Command is returned to the node ID which issued the Firmware Update Meta Data Request Get Command. The Firmware Update Meta Data Request Report Command can only be sent requested by the Firmware Update Meta Data Request Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD							
Command = FIRMWARE_UPDATE_MD_REQUEST_REPORT							
Status							

Status (8 bits)

The Status identifier can return the following values:

Status	Description
0x00	Invalid combination of Manufacturer ID and Firmware ID. The device to be firmware updated will not start to request the firmware data.
0x01	Requires some kind of physical authentication (e.g. activation of a pushbutton) on the device to be updated. The device to be firmware updated will not start to request the firmware data.
0xFF	Valid combination of Manufacturer ID and Firmware ID. The device to be firmware updated will start to request the firmware data from the node ID specified in the Firmware Update Meta Data Request Get Command.

4.35.5   Firmware Update Meta Data Get Command

The Firmware Update Meta Data Get Command is used to request the Firmware Update Meta Data Report Command in case a valid firmware is available. The Firmware Update Meta Data Get Command used as handshake to avoid buffer overflow in the receiving device e.g. when storing the firmware data in the external EEPROM. The Firmware Update Meta Data Get Command will OPTIONALly be able to request multiple Firmware Update Meta Data Report Commands to improve throughput. Any missing reports can also be requested individually.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD							
Command = FIRMWARE_UPDATE_MD_GET							
Number of Reports							
0	Report number 1						
Report number 2							

Number of Reports (8 bits)

Number of Firmware Update Meta Data Report Commands to be received without requesting each Firmware Update Meta Data Report Command by the Firmware Update Meta Data Get Command.

Report number 1 .. 2 (15 bits)

The Report number field indicates the Firmware Update Meta Data Report Command to be requested. The report number values MUST be a sequence starting from 1. The first byte (Report number 1) is the most significant byte.

CONFIDENTIAL

4.35.6 Firmware Update Meta Data Report Command

The Firmware Update Meta Data Report Command used to retrieve firmware data from a device. The Firmware Update Meta Data Report Command(s) can only be sent requested by the Firmware Update Meta Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD							
Command = FIRMWARE_UPDATE_MD_REPORT							
Last	Report number 1						
Report number 2							
Data 1							
…							
Data N							

Last (1 bit)

The Last field indicates if the requested Firmware Update Meta Data Report Command is the last one or the transfer is not complete. The field is equal to 0x01 when the last report is transmitted otherwise the field has the value 0x00. This field makes it possible to start requesting Firmware Update Meta Data Report Commands without knowing the total number of Firmware Update Meta Data Report Commands to be transferred.

Report number 1 .. 2 (15 bits)

Firmware Update Meta Data Report Command number received. This allows the receiving device to check if all requested reports are received. The report number values MUST be a sequence starting from 1. The first byte (Report number 1) is the most significant byte.

The report number can be used to calculate the destination offset of the data by the formula:

Offset = (report number − 1) x 44

The reports are typically transferred as a sequence starting from 1. The device MUST respond to any valid request for any report number to deal with reports that did not arrive properly.

Data 1 .. Data N (variable)

The Data fields contain the requested binary firmware data starting from address 0x0000. Each frame MUST contain 44 bytes to be able to calculate the offset. This result in a frame size equal to 48 bytes due to the payload limitations with respect to a routed single cast frame using 4 hops [1]. The last frame can be reduced in size according to the remaining binary firmware data to be transferred. The number of data fields transmitted in the last frame can be determined from the length field in the frame.

CONFIDENTIAL

4.35.7 Firmware Update Meta Data Status Report Command

The Firmware Update Meta Data Status Report Command used to return the firmware update status. The Firmware Update Meta Data Status Report Command is returned when the firmware update is completed or abandoned by the device receiving the firmware. It is not allowed to send a Firmware Update Meta Data Get Command after the Firmware Update Meta Data Status Report.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD							
Command = FIRMWARE_UPDATE_MD_STATUS_REPORT							
Status							

Status (8 bits)

The Status identifier can return the following values:

Status	Description
0x00	Module was unable to receive the requested firmware data without checksum error. Number of retries and request sequence of missing frames are implementation specific.
0x01	Module was unable to receive the requested firmware data. Number of retries and request sequence of missing frames are implementation specific.
0xFF	Firmware update written successfully to EEPROM. Module will now power cycle itself to install the new firmware.

4.35.8 Detailed description of frame flow

A device (Node A) wants to check the current firmware version loaded in another device (Node B). To obtain the firmware version Node A MUST request it as shown on the figure below:
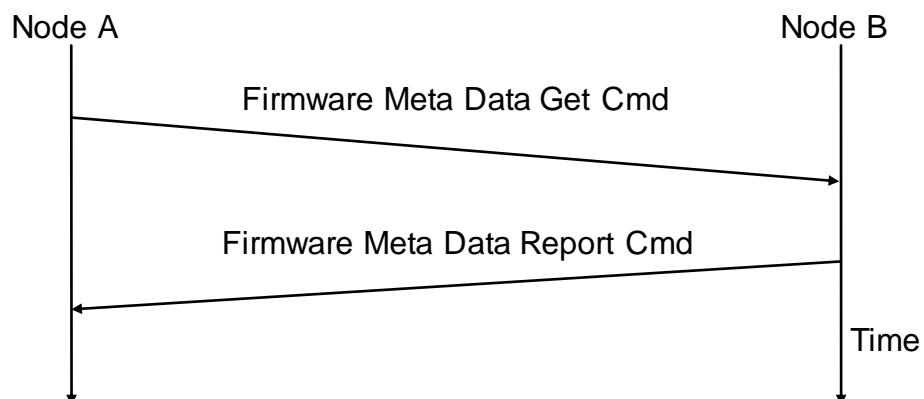
Figure 9, Requesting firmware version in a device

CONFIDENTIAL

To make a firmware update of Node B MUST Node A request Node B to start the process. To avoid unintentional firmware updates is it RECOMMENDED that some kind of physical authentication on the device to be updated is exercised. Node B returns a report to notify Node A whether or not the firmware update request is accepted. In case the firmware update request is accepted by Node B it will start to retrieve the firmware data. The complete frame flow is shown below:
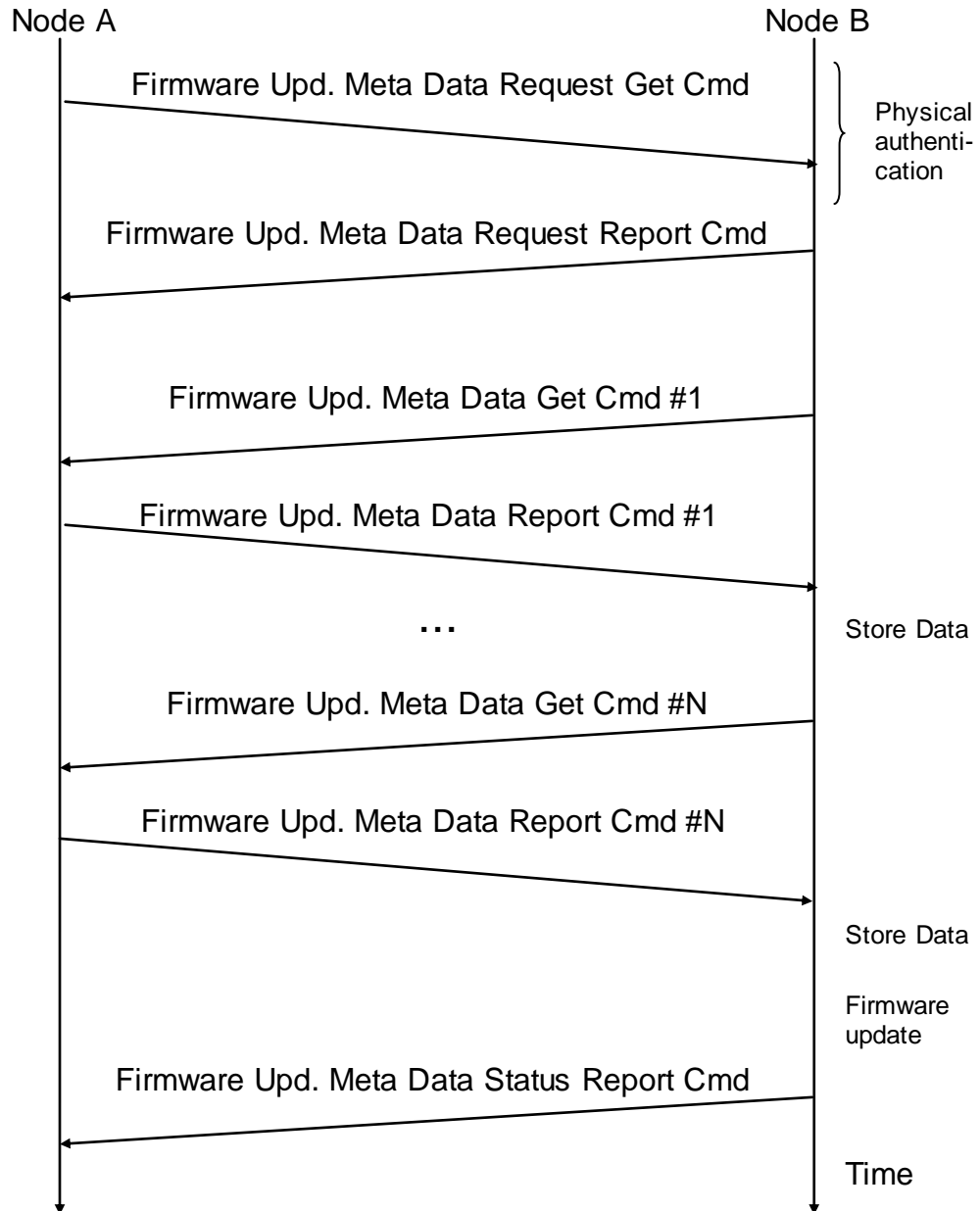
Figure 10, Firmware update of a device

4.36 Firmware Update Meta Data Command Class, version 2

The Firmware Update Meta Data Report is enhanced in Firmware Update Meta Data Command Class, version 2. The commands not mentioned here will remain the same as in version 1.

CONFIDENTIAL

4.36.1 Firmware Update Meta Data Report Command

The Firmware Update Meta Data Report command used to retrieve firmware data from a device. The Firmware Update Meta Data Report command(s) can only be sent as a result of receiving a Firmware Update Meta Data Get command.

7	6	5	4	3	2	1	0

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_FIRMWARE_UPDATE_MD							
Command = FIRMWARE_UPDATE_MD_REPORT							
Last	Report number 1						
Report number 2							
Data 1							
…							
Data N							
Checksum 1							
Checksum 2							

Last (1 bit)

The Last field indicates if the requested Firmware Update Meta Data Report command is the last one or the transfer is not complete. The field is equal to 0x01 when the last report is transmitted otherwise the field has the value 0x00. This field makes it possible to start requesting Firmware Update Meta Data Report commands without knowing the total number of Firmware Update Meta Data Report commands to be transferred.

Report number 1 .. 2 (15 bits)

Firmware Update Meta Data Report command number received. This allows the receiving device to check if all requested reports are received. The report number values MUST be a sequence starting from 1. The first byte (Report number 1) is the most significant byte.

The report number can be used to calculate the destination offset of the data by the formula:

Offset = (report number − 1) x 42

The reports are typically transferred as a sequence starting from 1. The device MUST respond to any valid request for any report number to deal with reports that did not arrive properly.

Data 1 .. Data N (variable)

The Data fields contain the requested binary firmware data starting from address 0x0000. Each frame MUST contain 42 bytes to be able to calculate the offset. This result in a frame size equal to 48 bytes due to the payload limitations with respect to a routed single cast frame using 4 hops [1]. The last frame can be reduced in size according to the remaining binary firmware data to be transferred. The number of data fields transmitted in the last frame can be determined from the length field in the frame.

CONFIDENTIAL

Checksum (16 bits)

The checksum field used to ensure consistency of the command class identifier, command identifier, report number and firmware data downloaded. The checksum is derived from the bytes starting with the command class identifier and until the last data field (Data N). The first byte is the most significant byte. The checksum algorithm is implementation specific but MUST be the same as use in the Firmware Update Meta Data Command Class, version 1. It is RECOMMENDED to use a checksum algorithm that implements a CRC-CCITT using initialization value equal to 0x1D0F and 0x1021 (normal representation) as the poly. Refer to Appendix C with respect to CRC_CCITT source code.

CONFIDENTIAL

4.37   Geographic Location Command Class, version 1

This Geographic Location Command Class used to read latitude and longitude from a device in a Z-Wave network. The latitude and longitude can also be set according to the geographic location in question. Date and geographic location can be used to calculate sunrise and sunset for e.g. automatic lighting control.

4.37.1   Geographic Location Set Command

The Geographic Location Get Command used to set latitude and longitude.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GEOGRAPHIC_LOCATION							
Command = GEOGRAPHIC_LOCATION_SET							
Longitude Degrees							
Long. Sign	Longitude Minutes						
Latitude Degrees							
Lat. Sign	Latitude Minutes						

Longitude (16 bits)

The longitude determines one's location on the earth's surface, East or West of the Greenwich Meridian. The Greenwich Meridian is located at the Greenwich observatory, in Greenwich, England to be the geographic point for where East and West meet. Therefore, Greenwich Meridian is indicated as 0° longitude. Longitude values for points East of the Meridian are always positive, while points West of the Meridian are always negative. Valid ranges are for degrees (from -180 to 180) and minutes (0-59). Other values will be interpreted as 0.

Latitude (16 bits)

The latitude determines one's location on the earth's surface, North or South of the Equator. Latitude is measured between -90° South, and +90° North of the Equator point (0°). Valid ranges are for degrees (from -90 to 90) and minutes (0-59). Other values will be interpreted as 0.

4.37.2   Geographic Location Get Command

The Geographic Location Get Command is used to request latitude and longitude from a device in a Z-Wave network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GEOGRAPHIC_LOCATION							
Command = GEOGRAPHIC_LOCATION_GET							

CONFIDENTIAL

4.37.3 Geographic Location Report Command

The Geographic Location Report Command returns latitude and longitude from a device in a Z-Wave network. The Geographic Location Report Command can be send requested by the Geographic Location Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GEOGRAPHIC_LOCATION							
Command = GEOGRAPHIC_LOCATION_REPORT							
Longitude Degrees							
Long. Sign	Longitude Minutes						
Latitude Degrees							
Lat. Sign	Latitude Minutes						

Refer to description under the Geographic Location Set Command.

CONFIDENTIAL

4.38 Grouping Name Command Class, version 1

The Grouping Name Command Class used to transfer name of groupings (as defined by the grouping identifier in the Association Command Class).

4.38.1 Grouping Name Set Command

The Grouping Name Set Command used to set a grouping Identifier name.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GROUPING_NAME							
Command = GROUPING_NAME_SET							
Grouping identifier							
Reserved					Char. Presentation		
Grouping Name 1							
Grouping Name 2							
…							
Grouping Name x							

Grouping Identifier (8 bits)

The field specifies the Grouping ID.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Char. Presentation (3 bits)

The char presentation identifier can be set to the following values:

Char. Presentation	Description
0	Using standard ASCII codes, see **Appendix B** (values 128-255 are ignored)
1	Using standard and OEM Extended ASCII codes, see **Appendix B**
2	Unicode UTF-16

Note: Devices supporting Unicode UTF-16 characters can have strings of a maximum of 8 characters because each character is described by a 2 byte long decimal representation. The first byte is the most significant byte. I.e. if there is one Unicode character in the set frame the char 1 will be MSB and char 2 will be LSB of the Unicode character.

This list MAY evolve in the future. Undefined values of the character presentation identifier MUST be ignored.

Grouping Name 1-x (variable)

Grouping name using specified character representation.

The Grouping name can have a maximum of 16 characters and a minimum of 0 characters. The number of character fields transmitted can be determined from the frame length. If a frame with more than 16 characters is received only the first 16 characters MUST be accept. The remaining characters MUST be ignored.

4.38.2 Grouping Name Get Command

The Grouping Name Get Command is used to request a grouping Identifier name.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GROUPING_NAME							
Command = GROUPING_NAME_GET							
Grouping identifier							

Grouping Identifier (8 bits)

The field specifies the grouping identifier.

CONFIDENTIAL

4.38.3 Group Name Report Command

The Grouping Name Report Command used to report the grouping Identifier name.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_GROUPING_NAME							
Command = GROUPING_NAME_REPORT							
Grouping identifier							
Reserved					Char. Presentation		
Grouping Name 1							
Grouping Name 2							
…							
Grouping Name x							

Grouping Identifier (8 bits)

The field specifies the grouping identifier.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Char. Presentation (3 bits)

Refer to description under the Grouping Name Set Command

Grouping Name 1-x (variable)

The Grouping Name fields contain the assigned group name. The Group Name can have a maximum of 16 characters and a minimum of 0 characters.

CONFIDENTIAL

4.39 Hail Command Class, version 1

The Hail Command Class used by applications to hail other devices in the Z-Wave network. The usage of the Hail Command Class is application specific.

4.39.1 Hail Command

Application can send unsolicited Hail Command to other devices in a Z-Wave network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HAIL							
Command = HAIL							

CONFIDENTIAL

4.40 HRV Status Command Class, version 1

The residential Heat Recovery Ventilation (HRV) Status Command Class used to read out a number of parameters in the ventilation system.

4.40.1 HRV Status Get Command

The residential HRV Status Get Command is used to request specific parameters from the ventilation system.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_STATUS							
Command = HRV_STATUS_GET							
Status Parameter							

Status Parameter (8 bits)

The status parameter used to indicate which status parameter that is requested.

Status Parameter	Value
Outdoor Air temperature	0
Supply Air (to room) temperature	1
Exhaust Air (from room) temperature	2
Discharge Air temperature	3
Room temperature	4
Relative Humidity in room	5
Remaining filter life	6

4.40.2 HRV Status Report Command

The residential HRV Status Report Command used to report a specific status parameter in response to a HRV Status Get.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_STATUS							
Command = HRV_STATUS_REPORT							
Status Parameter							
Precision			Scale		Size		
Value 1							
....							
Value n							

CONFIDENTIAL

Status Parameter (8 bits)

The status parameter used to indicate which status parameter that is reported. Refer to 4.40.1 HRV Status Get for possible values.

Precision (3 bits)

The precision field describes what the precision of the setpoint value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bits)

The scale field indicates the scale used the list of possible scales are given below:

Status Parameter	Scale	Value
Outdoor Air temperature	Celsius (C)	0
	Fahrenheit (F)	1
Supply Air (to room) temperature	Celsius (C)	0
	Fahrenheit (F)	1
Exhaust Air (from room) temperature	Celsius (C)	0
	Fahrenheit (F)	1
Discharge Air temperature	Celsius (C)	0
	Fahrenheit (F)	1
Room temperature	Celsius (C)	0
	Fahrenheit (F)	1
Relative Humidity in room	Percentage (%)	0
Remaining filter life	Percentage (%)	0

CONFIDENTIAL

Size (3 bits)

The size field indicates the number of bytes used for the sensor value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

Value 1 … Value n

The value is a signed field. The value can be 1, 2 or 4 bytes in size. This first byte is the most significant byte. The table below show examples of signed decimal value with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal		Signed 2 bytes decimal value	Hexadecimal
127	0x7F		32767	0x7FFF
25	0x19		1025	0x0401
2	0x02		2	0x0002
1	0x01		1	0x0001
0	0x00		0	0x0000
-1	0xFF		-1	0xFFFF
-2	0xFE		-2	0xFFFE
-25	0xE7		-1025	0xFBFF
-128	0x80		-32768	0x8000

4.40.3   HRV Status Supported Get Command

The HRV Status Supported Get Command is used to request a bitmap of the supported status parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_STATUS							
Command = HRV_STATUS_SUPPORTED_GET							

4.40.4   HRV Status Supported Report Command

The HRV Status Supported Report Command used to report a bitmap indicating the supported status parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_STATUS							
Command = HRV_STATUS_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

CONFIDENTIAL

Bit Mask 1 ... Bit Mask N (N * Byte)

The Bit Mask fields describe the supported status parameters from the ventilation system. Bit 0 in the Bit Mask 1 field used to indicate if the status parameter "Outdoor Air Temperature" is supported, 0 indicating not supported and 1 indicating supported. Bit 1 in the Bit Mask 1 field used to indicate if the status parameter "Supply Air Temperature" is supported and so forth. All available status parameters are given in Section 4.40.1 HRV Status Get.

It is only necessary to send the Bit Mask fields 1 and up to the one indicating the last support status parameter. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.41 HRV Control Command Class, version 1

The Heat Recovery Ventilation (HRV) Control Command Class is introducing control of Heat Recovery Ventilation systems via the Z-Wave interface.

4.41.1 HRV Mode Set

The HRV Mode Set Command used to set the desired mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_SET							
Reserved			Mode				

Mode (5 bits)

The mode identifier can be set to the following values:

Mode	Name	Description
0	Off	The HRV system is in the off state, frost protection can occur.
1	Demand / Automatic	The HRV system is controlled based on sensor input.
2	Schedule	The HRV system is controlled based on predefined input from the factory and/or user/installer.
3	Energy Savings Mode	The HRV system will be put into a reduced heat / ventilation mode.
4	Manual	The HRV system is in manual mode. The command HRV Manual Control Set can be used to manually control the device.

This list MAY evolve in the future. If a mode is not supported then it MUST be ignored.

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.41.2 HRV Mode Get Command

The HRV Mode Get Command is used to request the current mode from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_GET							

CONFIDENTIAL

4.41.3   HRV Mode Report Command

The HRV Mode Report Command used to report the mode from the device. It can be send either unsolicited or requested by the Mode Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_REPORT							
Reserved			Mode				

Mode (8 bits)

Refer to description under the HRV Mode Set command.

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.41.4   HRV Bypass Set Command

The HRV Bypass Set Command used to set the bypass mode when the ventilation system is set to manual mode. If the system is not in manual mode while receiving this command it MUST be ignored.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_BYPASS_SET							
Bypass							

Bypass (8 bits)

The value can be either 0x00 (close) or 0xFF (open).

Furthermore, it can take a percentage value between 1 to 99 (0x01 - 0x63) if the ventilation system supports modulated bypass, the percentage value will represent the aperture of the bypass. If the system does not support the full range of aperture steps, the supported values MUST be mapped linearly over the entire scale.

If ventilation system does support modulated bypass the values from 1 to 99 MUST be interpreted as fully open.

The value 254 (0xFE) used to set the bypass into automatic mode.

4.41.5   HRV Bypass Get Command

The HRV Bypass Get Command is used to request the current bypass setting.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_BYPASS _GET							

4.41.6   HRV Bypass Report Command

The HRV Bypass Report command used to report the current bypass parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_BYPASS_REPORT							
Bypass							

Bypass (8 bits)

See description under Section 4.41.4 HRV Bypass Set.

4.41.7   HRV Ventilation Rate Set Command

The HRV Ventilation Rate Set Command used to set the ventilation rate when the ventilation system is set to manual mode. If the system is not in manual mode while receiving this command it MUST be ignored.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_VENTILATION_RATE_SET							
Ventilation Rate							

Ventilation Rate (8 bits)

The value can be either 0x00 (off) or 0xFF (on). Furthermore, it can take a percentage value between 1 to 99 (0x01 - 0x63). A ventilation system is not REQUIRED to support all possible steps between 1 and 99.

4.41.8   HRV Ventilation Rate Get Command

The HRV Ventilation Rate Get Command is used to request the current ventilation rate setting.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_ VENTILATION_RATE _GET							

4.41.9   HRV Ventilation Rate Report Command

The HRV Ventilation Rate Report Command used to report the current ventilation rate setting.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_ VENTILATION_RATE _REPORT							
Ventilation Rate							

Ventilation Rate (8 bits)

See description under Section 4.41.7 HRV Ventilation Rate Set.

4.41.10   HRV Mode Supported Get Command

The HRV Mode Supported Get Command is used to request the supported modes from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_SUPPORTED_GET							

4.41.11   HRV Mode Supported Report Command

The HRV Mode Supported Report Command used to report the supported modes from the device. It can be send either unsolicited or requested by the Mode Supported Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_HRV_CONTROL							
Command = HRV_CONTROL_MODE_SUPPORTED_REPORT							
Reserved				Manual Control Supported			
Bit Mask 1							
…							
Bit Mask N							

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

Manual Control Supported (3 bits)

The manual control supported bits describes the supported manual control modes of the ventilation system. The mode is supported if the bit is 1; if the bit is 0 then the mode is not supported.

The bits are mapped to the following controls:

Bit	Name
0	Bypass Open / Close
1	Bypass Auto
2	Modulated Bypass
3	Ventilation Rate

E.g. a ventilation system supporting only open and close would report Manual Control Supported = 0x01.

Bit Mask 1 … Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported modes by the device. The bit 0 in Bit Mask 1 field used to indicate whether Mode = 0 (Off) is supported or not. The mode is supported if the bit is 1; if the bit is 0 then the mode is not supported. The bit 1 in Bit Mask 1 field used by Mode = 1 (Demand / Automatic) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported mode. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

All available modes are given in Section 4.41.1 HRV Mode Set.

4.42 Indicator Command Class, version 1

The Indicator Command Class used to show the actual state, level etc. on a device.

4.42.1 Indicator Set Command

The Indicator Set Command can be used to set a device on or off (enable or disable).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_INDICATOR							
Command = INDICATOR_SET							
Value							

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). Furthermore it can take values from 1 to 99 (0x01 – 0x63). In case the indicator does not have the capability to show the range from 1 to 99 it is interpreted as 0xFF (on/enable).

CONFIDENTIAL

4.42.2   Indicator Get Command

The Indicator Get Command can be used to get the state of the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_INDICATOR							
Command = INDICATOR_GET							

4.42.3   Indicator Report Command

The Indicator Report Command can be send unsolicited or requested by the Indicator Get Command.
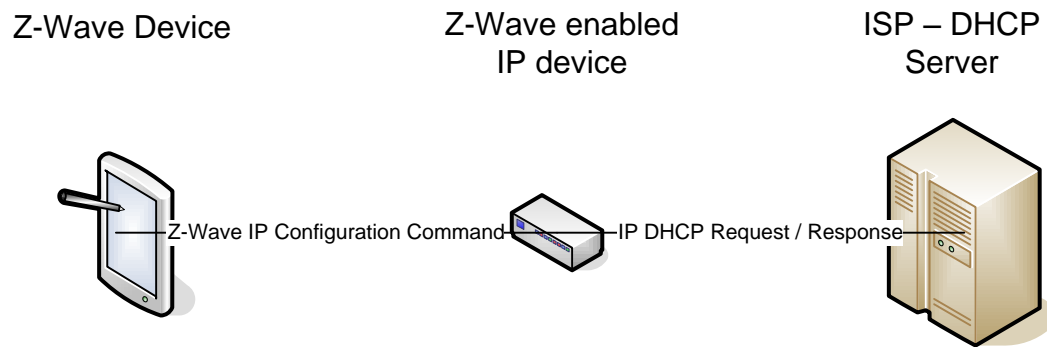
7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_INDICATOR							
Command = INDICATOR_REPORT							
Value							

Value (8 bits)

Refer to description under the Indicator Set Command.

CONFIDENTIAL

4.43   IP Configuration Command Class, version 1

The IP Configuration Command Class used to configure network identifiers for IPV4 devices. The intended use of the command class is illustrated in the figure below.

Figure 11, Configuration of network identifiers for IPV4 devices

In the figure the Z-Wave Remote to the left, sends an IP Configuration Command to the Z-Wave enabled IP device, telling it to acquire its configuration using DHCP. The Z-Wave enabled IP device will now perform a standard DHCP IP request to the DHCP server over an IP based network.

Another example might be where the Z-Wave Remote statically configures the Z-Wave enabled IP device with fixed IP, subnet, DNS etc. by sending an IP Configuration Command.

Note that this class is only intended for IPV4 and not IPV6 support.

4.43.1 IP Configuration Set Command

The IP Configuration Set Command used to configure IPV4 settings in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IP_CONFIGURATION							
Command = IP_CONFIGURATION_SET							
Reserved						Auto IP	Auto DNS
IP Address 1							
IP Address 2							
IP Address 3							
IP Address 4							
Subnet Mask 1							
Subnet Mask 2							
Subnet Mask 3							
Subnet Mask 4							
Gateway 1							
Gateway 2							
Gateway 3							
Gateway 4							
DNS1 1							
DNS1 2							
DNS1 3							
DNS1 4							
DNS2 1							
DNS2 2							
DNS2 3							
DNS2 4							

Reserved (6 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Auto IP (1 bit)

If Auto IP bit is set, the following fields are ignored: IP Address, Subnet Mask, and Gateway. And are allocated by DHCP or BOOTP instead.

Auto DNS (1 bit)

The Auto DNS if set indicates to ignore DNS1 and DNS2 and allocate DNS by DHCP instead. Note that some devices might not support Auto DNS without Auto IP set.

IP Address 1…4 (32 bits)

The IP Address indicates the static IP address of the device itself. The first byte is the most significant byte.

Subnet mask 1…4 (32 bits)

The Subnet Mask determines the portion of the IP address that represents the subnet. The first byte is the most significant byte.

Gateway 1…4 (32 bits)

The Gateway indicates the default gateway that serves as an access point to another network. The first byte is the most significant byte.

DNS1 1…4 (32 bits)

The DNS1 allows the use of domain name system (DNS) server names instead of using numerical IP addresses for management packet routing. In case the device will not need DNS, and SHOULD NOT query it from DHCP then leave field as all zeroes. The first byte is the most significant byte.

DNS2 1…4 (32 bits)

The DNS2 provides a secondary DNS server name. In case only one DNS server is available or the device will not need DNS then leave field as all zeroes. The first byte is the most significant byte.

4.43.2 IP Configuration Get Command

The IP Configuration Get Command is used to request the IPV4 settings in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IP_CONFIGURATION							
Command = IP_CONFIGURATION_GET							

4.43.3 IP Configuration Report Command

The IP Configuration Report Command used to return IPV4 settings in a device. The IP Configuration Report Command can only be sent requested by the IP Configuration Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IP_CONFIGURATION							
Command = IP_CONFIGURATION_REPORT							
Reserved						Auto IP	Auto DNS
IP Address1							
IP Address2							
IP Address3							
IP Address4							
Subnet Mask1							
Subnet Mask2							
Subnet Mask3							
Subnet Mask4							
Gateway1							
Gateway2							
Gateway3							
Gateway4							
DNS11							
DNS12							
DNS13							
DNS14							
DNS21							
DNS22							
DNS23							
DNS24							
LeaseTime1							
LeaseTime2							
LeaseTime3							
LeaseTime4							

Refer to explanation of parameters in IP Configuration Set Command description.

Lease Time 1…4 (32 bits)

The lease time specifies the time the IP address has been granted, if Auto IP is being used (in seconds). This field is OPTIONAL; if the device does not know its lease period it SHOULD return 0 for the lease time fields.

4.43.4 IP Configuration DHCP Release Command

The IP Configuration DHCP Release Command used to release the DHCP lease.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IP_CONFIGURATION							
Command = IP_CONFIGURATION_RELEASE							

4.43.5 IP Configuration DHCP Renew Command

The IP Configuration DHCP Renew Command used to force the renewal of the DHCP lease.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_IP_CONFIGURATION							
Command = IP_CONFIGURATION_RENEW							

4.44 Language Command Class, version 1

The Language Command Class used to specify the language settings on a device.

4.44.1 Language Set Command

The Language Set Command used to transfer data to a device. The device uses the default language in case the selected language is not supported.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LANGUAGE							
Command = LANGUAGE_SET							
Language 1							
Language 2							
Language 3							
Country 1							
Country 2							

Language 1 ... 3

The code definition of the languages can be found in ISO 639-2:1998 'Codes for the representation of names of languages – Part 2: Alpha-3 code'. In the table below are some examples of the alpha-3 codes listed:

Language	Language 1	Language 2	Language 3
English	e	n	g
Flemish; Dutch	d	u	t
French	f	r	e
German	g	e	r
Italian	i	t	a
Polish	p	o	l
Russian	r	u	s
Walloon	w	l	n

Country 1 ... 2

The code definition of the countries can be found in ISO 3166-1 'Country Codes: Alpha-2 codes'. The country is OPTIONAL and only defined in case it's necessary to differ geographical. The number of data fields transmitted can be determined from the length field in the frame. In the table below are some examples of the alpha-2 codes listed:

Language	Country 1	Country 2
Belgium	B	E
Italy	I	T
Netherlands	N	L
Poland	P	L
United Kingdom	G	B
United States	U	S

4.44.2 Language Get Command

The Language Get Command is used to request the current language setting in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LANGUAGE							
Command = LANGUAGE_GET							

CONFIDENTIAL

4.44.3 Language Report Command

The Language Report returns the current language setting in a device. The Language Report is obtained by the Language Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LANGUAGE							
Command = LANGUAGE_REPORT							
Language 1							
Language 2							
Language 3							
Country 1							
Country 2							

Refer to explanation under the Language Set Command.

CONFIDENTIAL

4.45   Lock Command Class, version 1

> **It´s RECOMMENDED using Door Lock Command Class instead for new devices.**

The Lock Command Class used to lock and unlock a "lock" type device, e.g. a door or window lock

4.45.1   Lock Set Command

The Lock Set Command used to set the lock state in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LOCK							
Command = LOCK_SET							
Lock State							

Lock State (8 bits)

The lock state field used to set the lock state of the device. The value 0 indicates that the device is unlocked. The value 1 indicates that the device is locked.

4.45.2   Lock Get Command

The Lock Get Command is used to request the lock state from a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LOCK							
Command = LOCK_GET							

4.45.3   Lock Report Command

The Lock Report Command used to report the lock state of a device. The Lock Report Command can be send unsolicited or requested by the Lock Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_LOCK							
Command = LOCK_REPORT							
Lock State							

Lock State (8 bits)

CONFIDENTIAL

The Lock state field used to report the lock state of the device. The value 0 indicates that the device is unlocked. The value 1 indicates that the device is locked.

CONFIDENTIAL

4.46   Manufacturer Proprietary Command Class, version 1

The Manufacturer Proprietary Command Class used to transfer data between devices in the Z-Wave network. The data content MUST be vendor specific and MUST be non-value added with respect to the Home Automation application in general. An example could be data used to diagnose the hardware in a device.

Note: Do not use the Manufacturer Proprietary Command Class without written approval from Sigma Designs.

4.46.1   Manufacturer Proprietary Command

The Manufacturer Proprietary Command used to transfer data between devices in the Z-Wave network. The Command contains a manufacturer specific identifier to allow the receiving device to check if this Command can be interpreted. The vendor is responsible for establishing a Command structure to differ between the set of Commands supported.

Meta Data: In case more than one frame MUST be transferred sequentially then the API call ZW_SendDataMeta can be used to ensure that control data gets through in the Z-Wave network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MANUFACTURER_PROPRIETARY							
Manufacturer ID 1							
Manufacturer ID 2							
Data 1							
…							
Data N							

Manufacturer ID (16 bits)

The Manufacturer ID is a unique ID identifying the manufacturer of the device. The Manufacturer ID is assigned by Sigma Designs upon request. See 0 for a list of assigned ID's. The first byte (Manufacturer ID 1) is the most significant byte.

Data 1 .. Data N (variable)

The data fields can be used for data transfer etc. The number of data fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.47   Manufacturer Specific Command Class, version 1

Use this Command Class to advertise manufacturer specific information use the Manufacturer Specific Command Class.

NOTICE: Sigma Designs upon request MUST assign the Manufacturer ID.

4.47.1   Manufacturer Specific Info Get Command

A device will use the Manufacturer Specific Info Get Command to get the manufacturer information from another device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MANUFACTURER_SPECIFIC							
Command = MANUFACTURER_SPECIFIC_GET							

CONFIDENTIAL

4.47.2 Manufacturer Specific Info Report Command

A device requires the Manufacturer Specific Info Report to obtain the manufacturer specific information in another device. The Manufacturer Specific Info Report Command can be send unsolicited or requested by the Manufacturer Specific Info Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MANUFACTURER_SPECIFIC							
Command = MANUFACTURER_SPECIFIC_REPORT							
Manufacturer ID 1							
Manufacturer ID 2							
Product Type ID 1							
Product Type ID 2							
Product ID 1							
Product ID 2							

Manufacturer ID (16 bits)

The Manufacturer ID is a unique ID identifying the manufacturer of the device. Sigma Designs upon request MUST assign the Manufacturer ID. See Appendix A for a list of assigned ID's. The first byte is the most significant byte.

Product Type ID (16 bits)

The manufacturer assigns the Product Type ID. The first byte is the most significant byte.

Product ID (16 bits)

The manufacturer assigns the Product ID. The first byte is the most significant byte.

4.48 Manufacturer Specific Command Class, version 2

Manufacturer Specific Command Class, version 2 is improved with a new set of commands to communicate e.g. the serial number of the product in either a supported format.

Commands not mentioned here remains unchanged as specified for Manufacturer Specific Command Class, Version 1.

4.48.1 Device Specific Get Command

This command is used to retrieve a Device Specific Report Command from a Z-Wave product.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MANUFACTURER_SPECIFIC							
Command = DEVICE_SPECIFIC_GET							
Reserved					Device ID Type		

Device ID Type (3 bits)

This field contains values for the Device ID Type. New values can be added upon request to Sigma Designs.

Device ID Type	Value
Return OEM <u>factory default</u> Device ID Type	0
Serial Number	1
Reserved	2-7

4.48.2 Device Specific Report Command

The Device Specific Report Command MUST be transmitted as a response to the Device Specific Get Command. This command MUST NOT be transmitted unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MANUFACTURER_SPECIFIC							
Command = DEVICE_SPECIFIC_REPORT							
Reserved					Device ID Type		
Device ID Data Format			Device ID Data Length Indicator				
Device ID Data 1							
Device ID Data 2							
...							
Device ID Data n							

CONFIDENTIAL

Device ID Data Format (3 bits)

This command field defines the format used for Device ID Data. New values can be added upon request to Sigma Designs.

Device ID Data Format	Value	Description
UTF-8	0	The Device ID Data is in UTF-8 format and MUST be displayed as is.
Binary	1	The Device ID Data is in plain binary format and MUST be displayed as hexadecimal values e.g. 0x30, 0x31, 0x32, 0x33 MUST be displayed as h'30313233.
Reserved	2-7	Reserved for future use.

Device ID Type (3 bits)

This field contains values for the Device ID Type. See Device Specific Get Command for details.

Device ID Data Length Indicator (5 bits)

This field contains the length of the Device ID Data fields.

Device ID Data (variable)

Data fields for Device ID. "Device ID Data Format" defines data format and "Device ID Data Length Indicator" defines length.

Note: Manufacturer ID and Device ID combined MUST be globally unique.

4.49   Meter Command Class, version 1

The Meter Command Class defines the Commands necessary to read accumulated values in physical units from a water meter or metering device (gas, electric etc.) and thereby enabling automatic meter reading capabilities.

Automatic meter reading (AMR), is the technology of automatically collecting data from water meter or energy metering devices and transferring that data to a central database for billing and/or analyzing.

4.49.1   Meter Get Command

The Meter Get Command is used to request the accumulated consumption in physical units from a metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET							

4.49.2   Meter Report Command

The Meter Report Command can be used by a metering device to send a report containing the accumulated consumption in physical units either unsolicited or requested by the Meter Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_REPORT							
Meter Type							
Precision			Scale		Size		
Meter Value 1							
Meter Value 2							
..							
Meter Value n							

CONFIDENTIAL

Meter Type (8 bits)

Meter Type specifies what type of metering device this Command originates from. Refer to the table below with respect to defined metering devices. New metering device types/values can be requested from Sigma Designs.

Sensor Type	Value
Electric meter	0x01
Gas meter	0x02
Water meter	0x03
	0x04
	0x05
	0x06

Precision (3 bits)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bits)

The Scale used to indicate what unit the metering device uses. Refer to the table below with respect to defined scales for the relevant metering devices. New scales/values can be requested from Sigma Designs.

Meter Type	Scale	Value
Electric meter	kWh	0x00
		0x01
		0x02
		0x03
Gas meter	Cubic meters	0x00
		0x01
		0x02
		0x03
Water meter	Cubic meters	0x00
	Cubic feet	0x01
	US gallons	0x02
		0x03

Size (3 bits)

The size field indicates the number of bytes that used for the meter value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

CONFIDENTIAL

Meter Value (variable)

The meter value is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal	Signed 2 bytes decimal value	Hexadecimal
127	0x7F	32767	0x7FFF
25	0x19	1025	0x0401
2	0x02	2	0x0002
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-2	0xFE	-2	0xFFFE
-25	0xE7	-1025	0xFBFF
-128	0x80	-32768	0x8000

Notice: The metering device receiving the Meter Report MUST always show the value even though the Metering Device Type and/or Scale are not supported.

CONFIDENTIAL

4.50 Meter Command Class, version 2

The Meter Command Class is intended for Z-Wave enabled devices capable of reporting energy measurements in addition to any main functionality or features e.g. an appliance module reporting the current consumption of the connected load. This command class is not intended for residential utility submetering such as a water meter counting total consumption. Utility meters MUST refer to the Advanced Energy Control framework.

Meter Command Class (Version 2) is improved with the following functionalities:

- Commands to interview the device for supported Meter types

- 'Previously accumulated consumption' to allow for easy calculation of consumption since last measurement.

- Reset of accumulated consumption

- Add capability to communicate current consumption (W)

- 'Scale' entry for pulse meters along with W, kVAh and Cubic ft.

The commands not mentioned here will remain the same as specified for Meter Command Class (Version 1).

4.50.1 Meter Supported Get Command

The Meter Supported Get Command is used to request the supported scales in a sub meter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_SUPPORTED_GET							

4.50.2 Meter Supported Report Command

The Meter Supported Report Command used to report the supported scales in a sub meter as a reply to the Meter Supported Get Command. The Meter Supported Report Command MUST NOT be sent unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_SUPPORTED_REPORT							
Meter Reset	Reserved		Meter Type				
Reserved				Scale Supported			

CONFIDENTIAL

Meter Reset (1 bit)

The Meter Reset field set to "1" indicates support for the Meter Reset Command.

Meter Type (5 bits)

See Meter Type defined in Meter Report Command.

Scale Supported (4 bits)

Meter Type	Scale	Value	Accumulate / Instant measurement
Electric meter	kWh	Bit 0 asserted	Accumulated
	kVAh	Bit 1 asserted	Accumulated
	W	Bit 2 asserted	Instant
	Pulse count	Bit 3 asserted	Accumulated
Gas meter	Cubic meters	Bit 0 asserted	Accumulated
	Cubic feet	Bit 1 asserted	Accumulated
	Reserved	Bit 2 asserted	n/a
	Pulse count	Bit 3 asserted	Accumulated
Water meter	Cubic meters	Bit 0 asserted	Accumulated
	Cubic feet	Bit 1 asserted	Accumulated
	US gallons	Bit 2 asserted	Accumulated
	Pulse count	Bit 3 asserted	Accumulated

4.50.3   Meter Reset Command

The Meter Reset Command used to reset ALL accumulated values stored in the meter device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_RESET							

4.50.4   Meter Get Command

The Meter Get Command is used to request a device supported measurement in physical units from a metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET							
Reserved			Scale		Reserved		

Scale (2 bits)

See description of Meter Report Command.

Note! A device receiving a Meter Get Command containing a non-supported Scale MUST ignore the command.

4.50.4.1   Backwards compatibility

When devices supporting Meter Command Class (Version 1) receives a Meter Get Command of Version 2 it MUST report its implemented scale.

When devices supporting Meter Command Class (Version 2) receives a Meter Get Command of Version 1 it MUST report its default scale. It is up to the manufacture to define which scale is the default scale and describe this in the product manual.

CONFIDENTIAL

4.50.5 Meter Report Command

The Meter Report Command MUST be transmitted as a response to the Meter Get Command and in addition it can also be transmitted unsolicited in case the OEM application deems this necessary. The OEM application MUST however ensure not to transmit unsolicited report commands using a broadcast frame, in order to avoid unintentional transmission of an unsolicited command to a destination not ratifying support for it.

The Meter Report Command used by the metering device to report the instant or accumulated consumption in physical units.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_REPORT							
Reserved	Rate Type		Meter Type				
Precision			Scale		Size		
Meter Value 1							
..							
Meter Value n							
Delta Time 1							
Delta Time 2							
Previous Meter Value 1							
..							
Previous Meter Value n							

Rate Type (2 bits)

Rate Type specifies if it is import or export values to be read. Setting the Rate Type to Import on the Meter Report Command is an indication that the Meter Value is a consumed measurement. In contrary when the Rate Type is set to Export the indication of the Meter Value is a produced measurement.

Rate Type	Value
Reserved	0x00
Import (consumed)	0x01
Export (produced)	0x02
Reserved	0x03

Meter Type (5 bits)

Meter Type specifies what type of metering device this command originates from. Refer to the table below with respect to defined metering devices. New metering device types/values can be requested from Sigma Designs.

Meter Type	Value
Reserved	0x00
Electric meter	0x01
Gas meter	0x02
Water meter	0x03
Reserved	0x04-0x1F

Precision (3 bits)

The Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 1 MUST be interpreted as 102.5 and if precision is 3 then the interpreted value is 1.025.

Scale (2 bits)

The Scale field used to indicate what unit the metering device uses. Refer to the table below with respect to defined scales for the relevant metering devices. New scales/values can be requested from Sigma Designs.

Meter Type	Scale	Value
Electric meter	kWh	0x00
	kVAh	0x01
	W	0x02
	Pulse count	0x03
Gas meter	Cubic meters	0x00
	Cubic feet	0x01
	Reserved	0x02
	Pulse count	0x03
Water meter	Cubic meters	0x00
	Cubic feet	0x01
	US gallons	0x02
	Pulse count	0x03

Size (3 bits)

The Size field indicates the number of bytes that used for the Meter Value and the Previous Meter Value. This field can take values from 1 (001b), 2 (010b) or 4 (100b) meaning both the Meter Value and the Previous Meter Value can be 1, 2 or 4 bytes long respectively.

Meter Value (variable) / Previous Meter Value (variable)

The Meter Value and Previous Meter Value are signed fields and can be 1, 2 or 4 bytes long defined by the Size field i.e. if the Size field is 2 then both Meter Value and Previous Meter Value are 2 bytes long. The first byte is the most significant byte.

The table below shows signed decimal values together with the hexadecimal equivalents.

Signed 1 byte		Signed 2 bytes		Signed 4 bytes	
Dec	**Hex**	**Dec**	**Hex**	**Dec**	**Hex**
127	0x7F	32767	0x7FFF	2147483647	0x7FFFFFFF
..	..	..	..	..	..
63	0x3F	16383	0x3FFF	1073741823	0x3FFFFFFF
..	..	..	..	..	..
1	0x01	1	0x0001	1	0x00000001
0	0x00	0	0x0000	0	0x00000000
-1	0xFF	-1	0xFFFF	-1	0xFFFFFFFF
..	..	..	..	..	..
-63	0xC1	-16383	0xC001	-1073741823	0xC0000001
..	..	..	..	..	..
-128	0x80	-32768	0x8000	-2147483648	0x80000000

Delta Time (16 bits)

The Delta Time field will report the elapsed time in seconds between the 'Meter Value' and the 'Previous Meter Value' measurements. Valid values are 0 to 65536 seconds.

Delta Time	Value
No Previous Meter Value field included in the Meter Report	0x0000
1 sec. between Meter Value and Previous Meter Value	0x0001
..	..
65534 sec. between Meter Value and Previous Meter Value	0xFFFE
Unknown Delta Time between Meter Value and Previous Meter Value.	0xFFFF

CONFIDENTIAL

4.50.5.1 Examples

To retrieve accumulated power consumption from a device supporting Meter Command Class (Version 2), the Meter Get Command MUST indicate what scale is requested.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x01 (METER_GET)							
0			0 (kWh)			0	

As a response to the above Meter Get Command an example of accumulated power consumption can be reported with the below Meter Report indicating a current measurement of 12065.298 kWh and 10 minutes ago the previous measurement was 12060.678 kWh.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x02 (METER_REPORT)							
0	2 (export)		0 (Electric meter)				
3 (3 decimals)			0 (kWh)		4 (4 bytes)		
0x00 (Meter Value 1)							
0xB8 (Meter Value 2)							
0x1A (Meter Value 3)							
0x12 (Meter Value 4)							
0x02 (Delta Time 1)							
0x58 (Delta Time 2)							
0x00 (Previous Meter Value 1)							
0xB8 (Previous Meter Value 2)							
0x08 (Previous Meter Value 3)							
0x06 (Previous Meter Value 4)							

To retrieve instant power consumption from a device supporting Meter Command Class (Version 2), the Meter Get Command MUST indicate what scale is requested.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x01 (METER_GET)							
0			2 (W)			0	

CONFIDENTIAL

An example of instant power consumption can be reported with the below Meter Report indicating a current measurement of 39.99 W.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x02 (METER_REPORT)							
0	2 (export)		0 (Electric meter)				
2 (2 decimals)			2 (W)		2 (2 bytes)		
0x0F (Meter Value 1)							
0x9F (Meter Value 2)							
0x00 (Delta Time 1)							
0x00 (Delta Time 2)							

4.51 Meter Command Class, version 3

The Meter Command Class is intended for Z-Wave enabled devices capable of reporting energy measurements in addition to any main functionality or features e.g. an appliance module reporting the current consumption of the connected load. This command class is not intended for residential utility submetering such as a water meter counting total consumption. Utility meters MUST refer to the Advanced Energy Control framework [3].

Meter Command Class (Version 3) is improved with the following functionalities:

- 'Scale' entry has been enlarged to support 8 scales

The commands not mentioned here will remain the same as specified for Meter Command Class (Version 2).

4.51.1 Meter Supported Report Command

The Meter Supported Report Command is used to report the supported scales in a sub meter as a reply to the Meter Supported Get Command. The Meter Supported Report Command MUST NOT be sent unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_SUPPORTED_REPORT							
Meter Reset	Reserved		Meter Type				
Scale Supported							

Meter Reset (1 bit)

The Meter Reset field set to "1" indicates support for the Meter Reset Command.

Meter Type (5 bit)

See Meter Type defined in Meter Report Command.

Scale Supported (8 bit)

Meter Type	Scale	Value	Accumulate / Instant measurement
Electric meter	kWh	Bit 0 asserted	Accumulated
	kVAh	Bit 1 asserted	Accumulated
	W	Bit 2 asserted	Instant
	Pulse count	Bit 3 asserted	Accumulated
	V	Bit 4 asserted	Instant
	A	Bit 5 asserted	Instant

	Power factor	Bit 6 asserted	Instant
	reserved	Bit 7 asserted	reserved
Gas meter	Cubic meters	Bit 0 asserted	Accumulated
	Cubic feet	Bit 1 asserted	Accumulated
	Reserved	Bit 2 asserted	n/a
	Pulse count	Bit 3 asserted	Accumulated
	Reserved	Bit 4-7 asserted	reserved
Water meter	Cubic meters	Bit 0 asserted	Accumulated
	Cubic feet	Bit 1 asserted	Accumulated
	US gallons	Bit 2 asserted	Accumulated
	Pulse count	Bit 3 asserted	Accumulated
	Reserved	Bit 4-7 asserted	reserved

4.51.2   Meter Get Command

The Meter Get Command is used to request a device supported measurement in physical units from a metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_GET							
Reserved		Scale			Reserved		

Scale (3 bit)

See description of Meter Report Command.

Note! A device receiving a Meter Get Command containing a non-supported Scale MUST ignore the command.

4.51.2.1   Backwards compatibility

When devices supporting Meter Command Class (Version 1) receives a Meter Get Command of Version 2 or 3 it MUST report its implemented scale.

When devices supporting Meter Command Class (Version 2 or 3) receives a Meter Get Command of Version 1 it MUST report its default scale. It is up to the manufacture to define which scale is the default scale and describe this in the product manual.

4.52 Meter Report Command

The Meter Report Command MUST be transmitted as a response to the Meter Get Command and in addition it can also be transmitted unsolicited in case the OEM application deems this necessary. The OEM application MUST however ensure not to transmit unsolicited report commands using a broadcast frame, in order to avoid unintentional transmission of an unsolicited command to a destination not ratifying support for it.

The Meter Report Command is used by the metering device to report the instant or accumulated consumption in physical units.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER							
Command = METER_REPORT							
Scale (2)	Rate Type		Meter Type				
Precision			Scale (1:0)		Size		
Meter Value 1							
..							
Meter Value n							
Delta Time 1							
Delta Time 2							
Previous Meter Value 1							
..							
Previous Meter Value n							

Rate Type (2 bit)

Rate Type specifies if it is import or export values to be read. Setting the Rate Type to Import on the Meter Report Command is an indication that the Meter Value is a consumed measurement. In contrary when the Rate Type is set to Export the indication of the Meter Value is a produced measurement.

Rate Type	Value
Reserved	0x00
Import (consumed)	0x01
Export (produced)	0x02
Reserved	0x03

CONFIDENTIAL

Meter Type (5 bit)

Meter Type specifies what type of metering device this command originates from. Refer to the table below with respect to defined metering devices. New metering device types/values can be requested from Sigma Designs.

Meter Type	Value
Reserved	0x00
Electric meter	0x01
Gas meter	0x02
Water meter	0x03
Reserved	0x04-0x1F

Precision (3 bit)

The Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 1 MUST be interpreted as 102.5 and if precision is 3 then the interpreted value is 1.025.

Scale (3 bit)

The Scale field is used to indicate what unit the metering device uses. Refer to the table below with respect to defined scales for the relevant metering devices. New scales/values can be requested from Sigma Designs.

Meter Type	Scale	Value
Electric meter	kWh	0x00
	kVAh	0x01
	W	0x02
	Pulse count	0x03
	V	0x04
	A	0x05
	Power Factor	0x06
	Reserved	0x07
Gas meter	Cubic meters	0x00
	Cubic feet	0x01
	Reserved	0x02
	Pulse count	0x03
	Reserved	0x04- 0x07
Water meter	Cubic meters	0x00
	Cubic feet	0x01
	US gallons	0x02

CONFIDENTIAL

| | Pulse count | 0x03 |
| | Reserved | 0x04-0x07 |

Size (3 bit)

The Size field indicates the number of bytes that is used for the Meter Value and the Previous Meter Value. This field can take values from 1 (001b), 2 (010b) or 4 (100b) meaning both the Meter Value and the Previous Meter Value can be 1, 2 or 4 bytes long respectively.

Meter Value (variable) / Previous Meter Value (variable)

The Meter Value and Previous Meter Value are signed fields and can be 1, 2 or 4 bytes long defined by the Size field i.e. if the Size field is 2 then both Meter Value and Previous Meter Value are 2 bytes long. The first byte is the most significant byte.

The table below shows signed decimal values together with the hexadecimal equivalents.

Signed 1 byte		Signed 2 bytes		Signed 4 bytes	
Dec	**Hex**	**Dec**	**Hex**	**Dec**	**Hex**
127	0x7F	32767	0x7FFF	2147483647	0x7FFFFFFF
..	..	..	..	..	..
63	0x3F	16383	0x3FFF	1073741823	0x3FFFFFFF
..	..	..	..	..	..
1	0x01	1	0x0001	1	0x00000001
0	0x00	0	0x0000	0	0x00000000
-1	0xFF	-1	0xFFFF	-1	0xFFFFFFFF
..	..	..	..	..	..
-63	0xC1	-16383	0xC001	-1073741823	0xC0000001
..	..	..	..	..	..
-128	0x80	-32768	0x8000	-2147483648	0x80000000

Delta Time (16 bit)

The Delta Time field will report the elapsed time in seconds between the 'Meter Value' and the 'Previous Meter Value' measurements. Valid values are 0 to 65536 seconds.

Delta Time	Value
No Previous Meter Value field included in the Meter Report	0x0000
1 sec. between Meter Value and Previous Meter Value	0x0001
..	..
65534 sec. between Meter Value and Previous Meter Value	0xFFFE
Unknown Delta Time between Meter Value and Previous Meter Value.	0xFFFF

4.52.1   Examples of Meter Report Commands

To retrieve Instant Voltage from a device supporting Meter Command Class (Version 3), the Meter Get Command MUST indicate what scale is requested.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x01 (METER_GET)							
0			4 (V)			0	

As a response to the above Meter Get Command an example of instant Voltage can be reported with the below Meter Report indicating a current measurement of 108.5 V and 10 minutes ago the previous measurement was 109.1V.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x02 (METER_REPORT)							
1 (V)	2 (export)		0 (Electric meter)				
1 (1 decimals)			0 (V)		4 (4 bytes)		
0x00 (Meter Value 1)							
0x00 (Meter Value 2)							
0x04 (Meter Value 3)							
0x3D (Meter Value 4)							
0x02 (Delta Time 1)							
0x58 (Delta Time 2)							
0x00 (Previous Meter Value 1)							
0x00 (Previous Meter Value 2)							
0x04 (Previous Meter Value 3)							
0x43 (Previous Meter Value 4)							

CONFIDENTIAL

To retrieve instant power consumption from a device supporting Meter Command Class (Version 2), the Meter Get Command MUST indicate what scale is requested.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x01 (METER_GET)							
0			2 (W)		0		

An example of instant power consumption can be reported with the below Meter Report indicating a current measurement of 39.99 W.

7	6	5	4	3	2	1	0
0x32 (COMMAND_CLASS_METER)							
0x02 (METER_REPORT)							
0	2 (export)		0 (Electric meter)				
2 (2 decimals)			2 (W)		2 (2 bytes)		
0x0F (Meter Value 1)							
0x9F (Meter Value 2)							
0x00 (Delta Time 1)							
0x00 (Delta Time 2)							

4.53   Meter Table Configuration Command Class, version 1

The Meter Table Configuration Command Class defines the Commands necessary to configure the fundamental properties of the meter.

The Meter Table configuration commands are separated from the Meter Table monitoring commands in the Meter Table Monitor Command Class, allowing the classes to be OPTIONAL supported at different Z-Wave security levels. (E.g. Meter table monitoring commands could be supported in any device, while enabling a strict and certificate based security solution for the Meter Table Configuration Command class). Please refer to the Hybrid Security Command class for more details regarding Z-Wave security levels.

4.53.1   Meter Table Point Adm Number Set Command

The Meter Table Point Adm Number Set Command is used to set the Meter Point Administration Number in the metering device. The Meter Point Administration Number is used to identify the customer.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_CONFIG							
Command = METER_TBL_TABLE_POINT_ADM_NO_SET							
Reserved			Number of Meter Point Adm Number Characters				
Meter Point Adm Number Character 1							
…							
Meter Point Adm Number Character N							

Reserved (3 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Number of Meter Point Adm Number Characters (5 bit)

Number of characters in the meter point administration number(1…32).

Meter Point Adm Number Character 1 .. Meter Point Adm Number Character N (Number of Meter Point Adm Number Characters * 8 bit)

The Meter Point Adm Number character fields hold the string identifying the customer. The character presentation uses standard ASCII codes (values 128-255 are ignored).

CONFIDENTIAL

4.54   Meter Table Monitor Command Class, version 1

The Meter Table Monitor Command Class defines the Commands necessary to read historical and accumulated values in physical units from a water meter or other metering device (gas, electric etc.) and thereby enabling automatic meter reading capabilities

4.54.1   Meter Table Point Adm. Number Get Command

The Meter Table Point Adm. Number Get Command is used to request the Meter Point Administration Number to identify customer.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_POINT_ADM_NO_GET							

4.54.2   Meter Table Point Adm. Number Report Command

The Meter Table Point Adm. Number Report Command reports parameters used for identification of customer and metering device and MUST be requested by the Meter Table Point Adm. Number Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_POINT_ADM_NO_REPORT							
Reserved			Number of Meter Point Adm. Number Characters				
Meter Point Adm. Number Character 1							
…							
Meter Point Adm. Number Character N							

Refer to description of fields under the Meter Table Point Adm. Number SET Command (section 4.53.1).

4.54.3   Meter Table ID Get Command

The Meter Table ID Get Command is used to request the parameters used for identification of customer and metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_ID_GET							

4.54.4 Meter Table ID Report Command

The Meter Table ID Report Command reports parameters used for identification of customer and metering device and MUST be requested by the Meter Table ID Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_ID_REPORT							
Reserved			Number of Meter ID Characters				
Meter ID Character 1							
…							
Meter ID Character N							

Reserved (3 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Number of Meter ID Characters (5 bit)

Number of characters defining the meter ID (1..32).

Meter ID Character 1 .. Meter ID Character N (Number of Meter ID Characters * 8 bit)

The Meter ID character fields hold the string identifying the individual metering device. The character presentation uses standard ASCII codes (values 128-255 are ignored). In addition one can use the Manufacturer Specific Command Class in conjunction as product identification.

4.54.5 Meter Table Capability Get Command

The Meter Table Capability Get Command is used to request the capabilities of a metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_CAPABILITY_GET							

CONFIDENTIAL

4.54.6 Meter Table Capability Report Command

The Meter Table Capability Report Command can be used by a metering device to send a report containing the meter table capabilities requested by the Meter Table Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_REPORT							
Rate Type		Meter Type					
Reserved				Pay Meter			
Dataset Supported 1							
Dataset Supported 2							
Dataset Supported 3							
Dataset History Supported 1							
Dataset History Supported 2							
Dataset History Supported 3							
Data History Supported 1							
Data History Supported 2							
Data History Supported 3							

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

Meter Type (6 bit)

Meter Type specifies the type of metering device the command originates. Meter Type defined as the *Meter Type* variable; refer to section 3.3.5 for a definition of the variable.

Reserved (4 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Pay Meter (4 bit)

Pay Meter specifies the way settling of account is done. Refer to the table below with respect to defined payment meters. Request new Pay Meter types/values from Sigma Designs.

Pay Meter	Value
Reserved	0x00
Creditmeter	0x01
Prepayment meter	0x02
Prepayment meter with debt recovery	0x03
Reserved	0x04-0x07

Dataset Supported 1, 2, 3/ Dataset History Supported 1, 2, 3 (24 bit)

Dataset Supported specifies which parameters, are available to be requested from the metering device. Dataset Supported Parameters defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

Dataset History Supported specifies the type of data/values, which can be requested through the Meter Table Historical data Get command (see 4.54.14) from the metering device. Dataset History Supported Parameters defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

Data History Supported (24 bit)

Data History Supported specifies number of possible entries in the buffer holding the historical values. Historical data cannot be retrieved when Data History Supported is equal to 0.

4.54.7 Meter Table Status Supported Get Command

The Meter Table Status Supported Get Command is used to request the supported operating status event parameters and logging depth of the metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_SUPPORTED_GET							

4.54.8 Meter Table Status Supported Report Command

The Meter Table Status Report Command is used to report the supported operation status' and logging depth of these in the meter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_SUPPORTED_REPORT							
Supported Operating Status 1							
Supported Operating Status 2							
Supported Operating Status 3							
Status Event Log Depth							

Supported Operating Status 1, 2, 3 (24 bit)

This command parameter is defined in a bitmap format and holds values and combinations of the meters operating status. If no "Operating Status Event" is supported by the application all fields of this parameter MUST be set to '0' as an indication of the meter is at "Normal/Idle" operating status. Supported Operating Status supports the following combinations of operating status:

Operating Status	Bit Map	Operating Status Event Identifier	Description
1	Bit 0	0x00	Disconnected The meter has been disconnected.
1	Bit 1	0x01	Leak detection A leak has been detected.
1	Bit 2	0x02	Power failure All phases / meter without power.
1	Bit 3	0x03	Power failure – Phase 1 Power failure on phase 1 detected.
1	Bit 4	0x04	Power failure – Phase 2 Power failure on phase 2 detected.
1	Bit 5	0x05	Power failure – Phase 3 Power failure on phase 3 detected.
1	Bit 6	0x06	Voltage Quality Voltage quality not within limits.
1	Bit 7	0x07	Tamper detection Tamper switch has been activated.
2	Bit 0	0x08	Magnetic tampering Magnetic tempering detected.
2	Bit 1	0x09	Meter Clock Set The clock has been set in the meter.
2	Bit 2	0x0A	Meter Clock Adjusted The clock has been adjusted in the meter.

2	Bit 3	0x0B	<u>Out of Credit</u> The meter is out of credit.
2	Bit 4	0x0C	<u>Emergency credit</u> The meter is running on emergency credit.
2	Bit 5	0x0D	<u>Transformer Ratio Changed</u> "Current transformation ratio" in meter changed.
2	Bit 6	0x0E	<u>Temperature Sensor 1 – Out of range</u> Temperature Sensor 1 outside specifications.
2	Bit 7	0x0F	<u>Temperature Sensor 2 – Out of range</u> Temperature Sensor 2 outside specifications.
3	Bit 0	0x10	<u>Temperature Sensor 3 – Out of range</u> Temperature Sensor 3 outside specifications.
3	Bit 1	0x11	<u>Burst Detection Heat</u> Burst detected in the heating network causing potential flooding.
3	Bit 2	0x12	<u>Error – Check Meter</u> Any other error condition not mentioned above.
3	Bit 4 – 7	0x13 – 0x1F	<u>Reserved</u> Values reserved for future expansion.

Status Event Log Depth (8 bit)

The *status event log depth* indicates the supported depth of the event log. If the meter only supports reporting the current status the status event log depth MUST be set to 0.

4.54.9 Meter Table Status Depth Get Command

The Meter Table Status Depth Get Command is used to request the current operating status of the metering device or to request a number of the latest status event from the logs.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR								
Command = METER_TBL _STATUS_DEPTH_GET								
Status Event Log Depth								

Status Event Log Depth (8 bit)

The *status event log depth* indicates the number of latest recorded events that SHOULD be returned in the corresponding report. If the log depth is set to 0 the meter will only return the current status; if the log depth is set to 0xFF the meter SHOULD return the entire status log.

CONFIDENTIAL

4.54.10 Meter Table Status Date Get Command

The Meter Table Status Date Get Command is used to request a number of status events recorded in a certain time interval. If the meter does not support a status event log history it MUST return the current state of the meter (see Meter Table Status Report Command).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_DATE_GET							
Maximum Reports							
Start - Year 1							
Start - Year 2							
Start - Month							
Start - Day							
Start - Hour Local Time							
Start - Minute Local Time							
Start - Second Local Time							
Stop - Year 1							
Stop - Year 2							
Stop - Month							
Stop - Day							
Stop - Hour Local Time							
Stop - Minute Local Time							
Stop - Second Local Time							

Maximum Reports (8 bit)

The maximum reports parameter is used to indicate the maximum number of reports to return based on the get. Reports are always returned with the most recently recorded log entry first. If set to 0x00 the meter will return all reports based on the request.

Start / Stop - Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Start / Stop - Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Start / Stop - Day (8 bit)

Specify the day of the month between 01 and 31.

Start / Stop - Hour Local Time (8 bit)

CONFIDENTIAL

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start / Stop - Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Start / Stop - Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

4.54.11 Meter Table Status Report Command

The Meter Table Status Report Command returns the current status of the meter and a given number of entries in the status event log depending on the parameters in the Meter Table Status Depth Get Command or Meter Table Status Date Get Command received.

This command can also be sent unsolicited – in this case only the current status will be transmitted.

CONFIDENTIAL

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS _REPORT							
Reports to follow							
Current Operating Status 1							
Current Operating Status 2							
Current Operating Status 3							
Event 1 - Type	Reserved		Event 1 - Operating Status Event ID				
Event 1 - Year 1							
Event 1 - Year 2							
Event 1 - Month							
Event 1 - Day							
Event 1 - Hour Local Time							
Event 1 - Minute Local Time							
Event 1 - Second Local Time							
…							
Event n -Type	Reserved		Event n - Operating Status Event ID				
Event n - Year 1							
Event n - Year 2							
Event n - Month							
Event n - Day							
Event n - Hour Local Time							
Event n -Minute Local Time							
Event n - Second Local Time							

Reports to follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

Current Operating Status 1, 2, 3 (24 bit)

This command parameter is defined in a bitmap format and holds values and combinations of the meters operating status. If no "Operating Status Event" has been registered for the period requested by the Meter Table Status Date Get Command, all fields of this parameter MUST be set to '0' AND no "Event" parameters are to be included in the Meter Table Status Report Command i.e. the command parameters ends after "Current Operating Status" . Refer to "Supported Operating Status" in section 4.54.8 for details.

Event n – Type (1 bit)

CONFIDENTIAL

The parameter reports type of event number *n* in the event log. The event type can be either 0 indicating that the entered the state described by the operating event status ID at the time reported, or 1 indicating that the meter left the state described by the operating event status ID.

Event n – Operating Status Event ID

The operating status event identifier of event number *n* in the event log. The identifiers available are given in the table in section 4.54.8.

Event n - Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Event n - Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Event n - Day (8 bit)

Specify the day of the month between 01 and 31.

Event n - Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Event n - Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Event n - Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

4.54.12 Meter Table Current Data Get Command

The Meter Table Current Data Get Command is used to request a number of time stamped values (current) in physical units according to the dataset mask.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_CURRENT_DATA_GET							
Dataset Requested 1							
Dataset Requested 2							
Dataset Requested 3							

CONFIDENTIAL

Dataset Requested 1, 2, 3 (24 bit)

The Dataset Requested is used to indicate which parameters are requested from the meter. The Dataset Requested parameter defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

4.54.13 Meter Table Current Data Report Command

The Meter Table Current Data Report Command used to report a number of time stamped values (current) in physical units in the device based on a dataset selection. The values MUST be returned in the order they are given in the Dataset bytes from less significant bit to most significant bit.

The Meter Table Data Report Command can be sent unsolicited or requested by the Meter Table Current Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_CURRENT_DATA_REPORT							
Reports to Follow							
Reserved						Rate Type	
Dataset 1							
Dataset 2							
Dataset 3							
Year 1							
Year 2							
Month							
Day							
Hour Local Time							
Minute Local Time							
Second Local Time							
Current Meter Precision 1			Current Meter Scale 1				
Current Value 1,1							
Current Value 1,2							
Current Value 1,3							
Current Value 1,4							
…							
Current Meter Precision N			Current Meter Scale N				
Current Value N,1							
Current Value N,2							
Current Value N,3							
Current Value N,4							

CONFIDENTIAL

Reports to follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

Reserved (6 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

Dataset 1, 2, 3 (24 bit)

The dataset parameter indicates which data is included in the report. Dataset parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte. A year equal to 0x0000 indicates that an accumulated value is not determined yet.

Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Day (8 bit)

Specify the day of the month between 01 and 31.

Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

Meter Precision (3 bit)

The Meter Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

CONFIDENTIAL

Meter Scale (5 bit)

The Meter Scale used to indicate the scale (unit) of the reported parameter.

Meter Type	Meter Scale	Value
Electric meter	kWh	0x00
	kVARh	0x01
	%	0x02
	Pulse count	0x03
	kVAR	0x04
	Voltage (V)	0x05
	Amperes (A)	0x06
	kW	0x07
	Ratio	0x08
	Reserved	0x09- 0x1F
Gas and water meter	Cubic meter	0x00
	Cubic feet	0x01
	US gallon	0x02
	Pulse count	0x03
	IMP gallon	0x04
	Liter	0x05
	kPa	0x06
	Centum cubic feet	0x07
	Cubic meter per hour	0x08
	Liter per hour	0x09
	kWh	0x0A
	MWh	0x0B
	KW	0x0C
	Hours	0x0D
	Reserved	0x0E-0x1F

Heating and Cooling Meter	Cubic meter (m³)	0x00
	Metric Ton (tonne) (t)	0x01
	Cubic meter per hour (m³/h)	0x02
	Liter per hour (l/h)	0x03
	kW	0x04
	MW	0x05
	kWh	0x06
	MWh	0x07
	Giga Joule (GJ)	0x08
	Giga Calorie (Gcal)	0x09
	Celsius (C°)	0x0A
	Fahrenheit (°F)	0x0B
	Hours	0x0C
	Reserved	0x0D-0x1F

Current Value (32 bit)

The Current Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

NOTICE: The device receiving the Meter Table Current Data Report MUST always show the value even though the scale is not supported.

CONFIDENTIAL

4.54.14 Meter Table Historical Data Get Command

The Meter Table Historical Data Get Command is used to request a number of time stamped values (historical) in physical units according to rate type, dataset mask and time interval.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_HISTORICAL_DATA_GET							
Maximum Reports							
Historical Dataset Requested 1							
Historical Dataset Requested 2							
Historical Dataset Requested 3							
Start Year 1							
Start Year 2							
Start Month							
Start Day							
Start Hour Local Time							
Start Minute Local Time							
Start Second Local Time							
Stop Year 1							
Stop Year 2							
Stop Month							
Stop Day							
Stop Hour Local Time							
Stop Minute Local Time							
Stop Second Local Time							

Maximum Reports (8 bit)

The maximum reports parameter is used to indicate the maximum number of reports to return based on the get. Reports are always returned with the most recently recorded value first. If set to 0x00 the meter will return all reports based on the request.

Dataset History 1, 2, 3 (24 bit)

The Historical Dataset Requested parameter is use to indicate which parameters are requested from the meter. Historical Dataset Requested parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

Start/Stop Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

CONFIDENTIAL

Start/Stop Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Start/Stop Day (8 bit)

Specify the day of the month between 01 and 31.

Start/Stop Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start/Stop Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Start/Stop Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

CONFIDENTIAL

4.54.15 Meter Table Historical Data Report Command

The Meter Table Historical Data Report Command used to report a number of time stamped values (historical) in physical units in the device. The values MUST be returned in the order they are given in the Dataset bytes from less significant bit to most significant bit.

The Meter Table Data Report Command can be sent unsolicited or requested by the Meter Table Historical Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_HISTORICAL_DATA_REPORT							
Reports to Follow							
Reserved						Rate Type	
Dataset 1							
Dataset 2							
Dataset 3							
Historical Year 1							
Historical Year 2							
Historical Month							
Historical Day							
Historical Hour Local Time							
Historical Minute Local Time							
Historical Second Local Time							
Historical Precision 1			Historical Scale 1				
Historical Value 1,1							
Historical Value 1,2							
Historical Value 1,3							
Historical Value 1,4							
…							
Historical Precision N			Historical Scale N				
Historical Value N,1							
Historical Value N,2							
Historical Value N,3							
Historical Value N,4							

Reports to Follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

CONFIDENTIAL

Reserved (6 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

Dataset 1, 2, 3 (24 bit)

This command parameter is defined in a bitmap format and holds values and combinations of the meters dataset. If no historical data has been registered for the period requested by the Meter Table Historical Data Get Command, all fields of this parameter MUST be set to '0' AND no "Historical" parameters are to be included in the Meter Table Historical Data Report Command i.e. the command parameters ends after "Dataset". Refer to "Meter Dataset" in section 0 for details.

Historical Year 1, 2 (16 bit)

Specify for the dataset the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Historical Month (8 bit)

Specify for the dataset the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that an accumulated value is not determined yet.

Historical Day (8 bit)

Specify for the dataset the day of the month between 01 and 31.

Historical Hour Local Time (8 bit)

Specify for the dataset the number of complete hours that have passed since midnight (00-23) in local time.

Historical Minute Local Time (8 bit)

Specify for the dataset the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Historical Second Local Time (8 bit)

Specify for the dataset the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

Historical Precision (3 bit)

The Historical Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Historical Scale (5 bit)

The Historical Scale used to indicate the scale (unit) of the reported parameter. The Historical Scale parameter is of the variable type *Meter Scale*; refer to Section 4.54.13 for a definition of the variable.

CONFIDENTIAL

Historical Value

The Historical Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

NOTICE: The device receiving the Meter Table Historical Data Report MUST always show the value even though the Scale is not supported.

CONFIDENTIAL

4.55   Meter Table Monitor Command Class, version 2

The Meter Table Monitor Command Class defines the Commands necessary to read historical and accumulated values in physical units from a water meter or other metering device (gas, electric etc.) or electric sub-metering device and thereby enabling automatic meter reading capabilities

4.55.1   Meter Table Point Adm. Number Get Command

The Meter Table Point Adm. Number Get Command is used to request the Meter Point Administration Number to identify customer.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_POINT_ADM_NO_GET							

4.55.2   Meter Table Point Adm. Number Report Command

The Meter Table Point Adm. Number Report Command reports parameters used for identification of customer and metering device and must be requested by the Meter Table Point Adm. Number Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_POINT_ADM_NO_REPORT							
Reserved			Number of Meter Point Adm. Number Characters				
Meter Point Adm. Number Character 1							
…							
Meter Point Adm. Number Character N							

Refer to description of fields under the Meter Table Point Adm. Number SET Command in Meter Table Configuration Command Class, version 1.

If Meter Table Point Adm, Number is not Set using the Meter Table Point Adm, Number SET Command in Meter Table Configuration Command Class, version 1 the Reported Number of Meter Point Adm, Number must be 0x00 and no Meter Table Point Adm, Numbers must be reported.

CONFIDENTIAL

4.55.3   Meter Table ID Get Command

The Meter Table ID Get Command is used to request the parameters used for identification of customer and metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_ID_GET							

4.55.4   Meter Table ID Report Command

The Meter Table ID Report Command reports parameters used for identification of customer and metering device and must be requested by the Meter Table ID Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_ID_REPORT							
Reserved			Number of Meter ID Characters				
Meter ID Character 1							
…							
Meter ID Character N							

Reserved (3 bit)

The reserved field is for future use. The implementation shall zero these fields and shall make no assumptions on the values of these fields nor perform processing based on their content.

Number of Meter ID Characters (5 bit)

Number of characters defining the meter ID (1..32).

Meter ID Character 1 .. Meter ID Character N (Number of Meter ID Characters * 8 bit)

The Meter ID character fields hold the string identifying the individual metering device. The character presentation uses standard ASCII codes (values 128-255 are ignored). In addition one can use the Manufacturer Specific Command Class in conjunction as product identification.

Note: If the Meter Table ID is not used Number of Meter ID Characters must be reported as 0x00 and no Meter ID Characters must be reported.

CONFIDENTIAL

4.55.5 Meter Table Capability Get Command

The Meter Table Capability Get Command is used to request the capabilities of a metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_TABLE_CAPABILITY_GET							

4.55.6 Meter Table Capability Report Command

The Meter Table Capability Report Command can be used by a metering device to send a report containing the meter table capabilities requested by the Meter Table Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_REPORT							
Rate Type		Meter Type					
Reserved				Pay Meter			
Dataset Supported 1							
Dataset Supported 2							
Dataset Supported 3							
Dataset History Supported 1							
Dataset History Supported 2							
Dataset History Supported 3							
Data History Supported 1							
Data History Supported 2							
Data History Supported 3							

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

If the Meter Type is Submeter the Rate Type must be Import to indicate consumption.

Meter Type (6 bit)

Meter Type specifies the type of metering device the command originates. Meter Type defined as the *Meter Type* variable; refer to section 3.3.5 for a definition of the variable.

Reserved (4 bit)

The reserved field is for future use. The implementation shall zero these fields and shall make no assumptions on the values of these fields nor perform processing based on their content.

Pay Meter (4 bit)

Pay Meter specifies the way settling of account is done. Request new Pay Meter types/values from Sigma Designs.

Pay Meter	Value
Reserved	0x00
Creditmeter	0x01
Prepayment meter	0x02
Prepayment meter with debt recovery	0x03
Reserved	0x04-0x07

Dataset Supported 1, 2, 3/ Dataset History Supported 1, 2, 3 (24 bit)

Dataset Supported specifies which parameters, are available to be requested from the metering device. Dataset Supported Parameters defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

Dataset History Supported specifies the type of data/values, which can be requested through the Meter Table Historical data Get command (see 4.55.14) from the metering device. Dataset History Supported Parameters defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

Data History Supported (24 bit)

Data History Supported specifies number of possible entries in the buffer holding the historical values. Historical data cannot be retrieved when Data History Supported is equal to 0.

4.55.7   Meter Table Status Supported Get Command

The Meter Table Status Supported Get Command is used to request the supported operating status event parameters and logging depth of the metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_SUPPORTED_GET							

CONFIDENTIAL

4.55.8   Meter Table Status Supported Report Command

The Meter Table Status Report Command is used to report the supported operation status' and logging depth of these in the meter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_SUPPORTED_REPORT							
Supported Operating Status 1							
Supported Operating Status 2							
Supported Operating Status 3							
Status Event Log Depth							

Supported Operating Status (1–4 Bytes)

This command parameter is defined in a bitmap format and holds values and combinations of the meters operating status. If no "Operating Status Event" is supported by the application all fields of this parameter must be set to '0' as an indication of the meter is at "Normal/Idle" operating status. Supported Operating Status supports the following combinations of operating status:

Status Event Log Depth (8 bit)

The *status event log depth* indicates the supported depth of the event log. If the meter only supports reporting the current status the status event log depth must be set to 0.

4.55.9   Meter Table Status Depth Get Command

The Meter Table Status Depth Get Command is used to request the current operating status of the metering device or to request a number of the latest status event from the logs.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_DEPTH_GET							
Status Event Log Depth							

Status Event Log Depth (8 bit)

The *status event log depth* indicates the number of latest recorded events that should be returned in the corresponding report. If the log depth is set to 0 the meter will only return the current status; if the log depth is set to 0xFF the meter should return the entire status log.

CONFIDENTIAL

4.55.10   Meter Table Status Date Get Command

The Meter Table Status Date Get Command is used to request a number of status events recorded in a certain time interval. If the meter does not support a status event log history it must return the current state of the meter (see section 4.55.11)

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS_DATE_GET							
Maximum Reports							
Start - Year 1							
Start - Year 2							
Start - Month							
Start - Day							
Start - Hour Local Time							
Start - Minute Local Time							
Start - Second Local Time							
Stop - Year 1							
Stop - Year 2							
Stop - Month							
Stop - Day							
Stop - Hour Local Time							
Stop - Minute Local Time							
Stop - Second Local Time							

Maximum Reports (8 bit)

The maximum reports parameter is used to indicate the maximum number of reports to return based on the get. Reports are always returned with the most recently recorded log entry first. If set to 0x00 the meter will return all reports based on the request.

Start / Stop - Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Start / Stop - Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Start / Stop - Day (8 bit)

Specify the day of the month between 01 and 31.

Start / Stop - Hour Local Time (8 bit)

CONFIDENTIAL

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start / Stop - Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Start / Stop - Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

4.55.11 Meter Table Status Report Command

The Meter Table Status Report Command returns the current status of the meter and a given number of entries in the status event log depending on the parameters in the Meter Table Status Depth Get Command or Meter Table Status Date Get Command received.

This command can also be sent unsolicited – in this case only the current status will be transmitted.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL _STATUS _REPORT							
Reports to follow							
Current Operating Status 1							
Current Operating Status 2							
Current Operating Status 3							
Event 1 - Type	Reserved		Event 1 - Operating Status Event ID				
Event 1 - Year 1							
Event 1 - Year 2							
Event 1 - Month							
Event 1 - Day							
Event 1 - Hour Local Time							
Event 1 - Minute Local Time							
Event 1 - Second Local Time							
…							
Event n -Type	Reserved		Event n - Operating Status Event ID				
Event n - Year 1							
Event n - Year 2							
Event n - Month							
Event n - Day							
Event n - Hour Local Time							
Event n -Minute Local Time							
Event n - Second Local Time							

Reports to follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

CONFIDENTIAL

Current Operating Status 1, 2, 3 (24 bit)

This command parameter is defined in a bitmap format and holds values and combinations of the meters operating status. If no "Operating Status Event" has been registered for the period requested by the Meter Table Status Date Get Command, all fields of this parameter must be set to '0' AND no "Event" parameters are to be included in the Meter Table Status Report Command i.e. the command parameters ends after "Current Operating Status" . Refer to "Supported Operating Status" in section 4.54.8 for details.

Event n – Type (1 bit)

The parameter reports type of event number *n* in the event log. The event type can be either 0 indicating that the entered the state described by the operating event status ID at the time reported, or 1 indicating that the meter left the state described by the operating event status ID.

Event n – Operating Status Event ID

The operating status event identifier of event number *n* in the event log. The identifiers available are given in the table in section 4.54.8.

Event n - Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Event n - Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Event n - Day (8 bit)

Specify the day of the month between 01 and 31.

Event n - Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Event n - Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Event n - Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

4.55.12   Meter Table Current Data Get Command

The Meter Table Current Data Get Command is used to request a number of time stamped values (current) in physical units according to the dataset mask.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_CURRENT_DATA_GET							
Dataset Requested 1							
Dataset Requested 2							
Dataset Requested 3							

Dataset Requested 1, 2, 3 (24 bit)

The Dataset Requested is used to indicate which parameters are requested from the meter. The Dataset Requested parameter defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

CONFIDENTIAL

4.55.13   Meter Table Current Data Report Command

The Meter Table Current Data Report Command used to report a number of time stamped values (current) in physical units in the device based on a dataset selection. The values must be returned in the order they are given in the Dataset bytes from less significant bit to most significant bit.

The Meter Table Data Report Command can be sent unsolicited or requested by the Meter Table Current Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_CURRENT_DATA_REPORT							
Reports to Follow							
Operating Status Indication	Reserved					Rate Type	
Dataset 1							
Dataset 2							
Dataset 3							
Year 1							
Year 2							
Month							
Day							
Hour Local Time							
Minute Local Time							
Second Local Time							
Current Meter Precision 1			Current Meter Scale 1				
Current Value 1,1							
Current Value 1,2							
Current Value 1,3							
Current Value 1,4							
…							
Current Meter Precision N			Current Meter Scale N				
Current Value N,1							
Current Value N,2							
Current Value N,3							
Current Value N,4							

CONFIDENTIAL

Reports to follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

Operating Status Indication (1 bit)

The Operating Status Indication bit is used to indicate that the reported meter data is measured while the meter is in a operating status different from Normal e.g. accuracy warning or clock not accurate.

Setting the bit to 1 means that the meter is operating in a status different from Normal. Setting the bit to 0 means that the meter is operating in Normal mode.

Reserved (5 bit)

The reserved field is for future use. The implementation shall zero these fields and shall make no assumptions on the values of these fields nor perform processing based on their content.

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

If the Meter Type is Submeter the Rate Type must be Import to indicate consumption.

Dataset 1, 2, 3 (24 bit)

The dataset parameter indicates which data is included in the report. Dataset parameters defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte. A year equal to 0x0000 indicates that an accumulated value is not determined yet.

Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Day (8 bit)

Specify the day of the month between 01 and 31.

Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

CONFIDENTIAL

Meter Precision (3 bit)

The Meter Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Meter Scale (5 bit)

The Meter Scale used to indicate the scale (unit) of the reported parameter. Meter Scale variables; refer to section **Error! Reference source not found.** for a definition of the variable.

Current Value (32 bit)

The Current Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

NOTICE: The device receiving the Meter Table Current Data Report must always show the value even though the scale is not supported.

CONFIDENTIAL

4.55.14   Meter Table Historical Data Get Command

The Meter Table Historical Data Get Command is used to request a number of time stamped values (historical) in physical units according to rate type, dataset mask and time interval.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR								
Command = METER_TBL_HISTORICAL_DATA_GET								
Maximum Reports								
Historical Dataset Requested 1								
Historical Dataset Requested 2								
Historical Dataset Requested 3								
Start Year 1								
Start Year 2								
Start Month								
Start Day								
Start Hour Local Time								
Start Minute Local Time								
Start Second Local Time								
Stop Year 1								
Stop Year 2								
Stop Month								
Stop Day								
Stop Hour Local Time								
Stop Minute Local Time								
Stop Second Local Time								

Maximum Reports (8 bit)

The maximum reports parameter is used to indicate the maximum number of reports to return based on the get. Reports are always returned with the most recently recorded value first. If set to 0x00 the meter will return all reports based on the request.

Dataset History 1, 2, 3 (24 bit)

The Historical Dataset Requested parameter is use to indicate which parameters are requested from the meter. Historical Dataset Requested parameters defined as the *Meter Dataset* variable; refer to section 3.3.2 for a definition of the variable.

Start/Stop Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

CONFIDENTIAL

Start/Stop Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Start/Stop Day (8 bit)

Specify the day of the month between 01 and 31.

Start/Stop Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start/Stop Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Start/Stop Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

CONFIDENTIAL

4.55.15 Meter Table Historical Data Report Command

The Meter Table Historical Data Report Command used to report a number of time stamped values (historical) in physical units in the device. The values must be returned in the order they are given in the Dataset bytes from less significant bit to most significant bit.

The Meter Table Data Report Command can be sent unsolicited or requested by the Meter Table Historical Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_MONITOR							
Command = METER_TBL_HISTORICAL_DATA_REPORT							
Reports to Follow							
Operating Status Indication	Reserved					Rate Type	
Dataset 1							
Dataset 2							
Dataset 3							
Historical Year 1							
Historical Year 2							
Historical Month							
Historical Day							
Historical Hour Local Time							
Historical Minute Local Time							
Historical Second Local Time							
Historical Precision 1		Historical Scale 1					
Historical Value 1,1							
Historical Value 1,2							
Historical Value 1,3							
Historical Value 1,4							
…							
Historical Precision N		Historical Scale N					
Historical Value N,1							
Historical Value N,2							
Historical Value N,3							
Historical Value N,4							

CONFIDENTIAL

Reports to Follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

Operating Status Indication (1 bit)

The Operating Status Indication bit is used to indicate that the reported meter data is measured while the meter is in a operating status different from Normal e.g. accuracy warning or clock not accurate.

Setting the bit to 1 means that the meter is operating in a status different from Normal. Setting the bit to 0 means that the meter is operating in Normal mode.

Reserved (6 bit)

The reserved field is for future use. The implementation shall zero these fields and shall make no assumptions on the values of these fields nor perform processing based on their content.

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

If the Meter Type is Submeter the Rate Type must be Import to indicate consumption.

Dataset 1, 2, 3 (24 bit)

This command parameter is defined in a bitmap format and holds values and combinations of the meters dataset. If no historical data has been registered for the period requested by the Meter Table Historical Data Get Command, all fields of this parameter must be set to '0' AND no "Historical" parameters are to be included in the Meter Table Historical Data Report Command i.e. the command parameters ends after "Dataset". Refer to "Meter Dataset" in section 3.3.2 for details.

Historical Year 1, 2 (16 bit)

Specify for the dataset the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Historical Month (8 bit)

Specify for the dataset the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that an accumulated value is not determined yet.

Historical Day (8 bit)

Specify for the dataset the day of the month between 01 and 31.

Historical Hour Local Time (8 bit)

Specify for the dataset the number of complete hours that have passed since midnight (00-23) in local time.

Historical Minute Local Time (8 bit)

Specify for the dataset the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Historical Second Local Time (8 bit)

Specify for the dataset the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

Historical Precision (3 bit)

The Historical Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Historical Scale (5 bit)

The Historical Scale used to indicate the scale (unit) of the reported parameter. The Historical Scale parameter is of the variable type *Meter Scale*; refer to Section 4.54.13 for a definition of the variable.

Historical Value

The Historical Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

NOTICE: The device receiving the Meter Table Historical Data Report must always show the value even though the Scale is not supported.

CONFIDENTIAL

4.56   Meter Table Push Configuration Command Class version 1

The Meter Table Push Configuration Command Class is used to configure the meter to send a Current Data Report at a given interval. The meter can be setup to return different dataset at different intervals using both the primary and secondary push commands.

CONFIDENTIAL

4.56.1   Meter Table Push Configuration Set Command

The Meter Table Push Configuration Set Command is used to request the meter to send a Current Data Report at a given interval. The meter can be setup to return different dataset at different intervals using both the primary and secondary push commands.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_PUSH							
Command = METER_TBL_PUSH_CONFIGURATION_SET							
Reserved			P/S	Operating Status Push Mode			
Push Dataset 1							
Push Dataset 2							
Push Dataset 3							
Interval Months							
Interval Days							
Interval Hours							
Interval Minutes							
Push Node ID							

Operating Status Push Mode (4 bit)

The Operating Status Push Mode is use to configure if the Meter Table Status Report Command (please refer to 4.54.11) participates in the Push Functionality

Operating Status Push Mode Identifier	Description
0x00	Operating Status push disabled
0x01	Operating Status push based on Interval
0x02	Operating Status push based on Status Change
0x03	Operating Status push Based on Interval AND status change
0x04-0x0F	Reserved

P/S (1 bit)

P/S	Description
0x00	Primary push configuration
0x01	Secondary push configuration

Push Dataset 1, 2, 3 (24 bit)

The Push Dataset parameter is use to indicate which parameters are requested to be pushed from the meter. The Push Dataset parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

Interval Months (8 bit)

Specify the number of months between pushing the push dataset.

Interval Day (8 bit)

Specify the number of days between pushing the push dataset.

Interval Hours (8 bit)

Specify the number of hours between pushing the push dataset.

Interval Minute (8 bit)

Specify the number of minutes between pushing the push dataset.

Push Node ID (8 bit)

Specify the node ID of the node to receive push dataset in the given interval.

4.56.2   Meter Table Push Configuration Get Command

The Meter Table Push Configuration Get Command is used to request the meters push configuration

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_PUSH							
Command = METER_TBL_PUSH_CONFIGURATION_GET							

CONFIDENTIAL

4.56.3   Meter Table Push Configuration Report Command

The Meter Table Push Configuration Report Command is used report the current Push Configuration

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_TBL_PUSH
Command Class = COMMAND_CLASS_METER_TBL_PUSH							
Command = METER_TBL_PUSH_CONFIGURATION_REPORT							
Reserved			P/S	Operating Status Push Mode			
Push Dataset 1							
Push Dataset 2							
Push Dataset 3							
Interval Months							
Interval Days							
Interval Hours							
Interval Minutes							
Push Node ID							

Please refer to Meter Table Push Configuration Set Command (section 4.56.1) for detailed description of the fields.

CONFIDENTIAL

4.57 Move To Position Window Covering Command Class, version 1

The Move To Position Window Covering Command Class used to move the window covering to a given position and request current position.

4.57.1 Move To Position Set Command

The Move To Position Set Command used to move the window covering to a given position. The speed of the movement is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MTP_WINDOW_COVERING							
Command = MOVE_TO_POSITION_SET							
Value							

Value (8 bits)

The value can be either 0x00 (close) or 0xFF (open). Furthermore it can take values from 1 to 99 (0x01 – 0x63). The unit of the value is percentage.

4.57.2 Move To Position Get Command

The Move To Position Get Command is used to request the actual position of a drape, shade, blind etc.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MTP_WINDOW_COVERING							
Command = MOVE_TO_POSITION_GET							

4.57.3 Move To Position Report Command

The Move To Position Report Command is sent requested by the Move To Position Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MTP_WINDOW_COVERING							
Command = MOVE_TO_POSITION_REPORT							
Value							

Value (8 bits)

Refer to explanation under the Move To Position Set Command.

CONFIDENTIAL

4.58 Multi Command Command Class, version 1

This Multi Command Command Class used to encapsulate multiple Commands in one Command. The purpose for this command class is to limit the number of transmissions to reduce network traffic and improve latency when a number of Commands MUST be executed sequentially in a device. This command class can also be useful for e.g. battery operated devices in order to extend battery lifetime.

A device that support the Multi Command Command Class MAY receive a combination of encapsulated and normal non-encapsulated requests and the correct response SHOULD be as follows:

   d) If the request is sent encapsulated, the response MUST be returned encapsulated also.

   e) If the request is sent as a normal frame, i.e. non-encapsulated, the response MUST also be sent non-encapsulated. Only if c) below is observed the response can be send encapsulated.

   f) Before any setting, request, or response is send encapsulated to any other device, the sending device MUST ensure, that the destination will be able to understand the encapsulated Command.

For a device sending set or request Commands, this could for instance be through the device application software requesting the NIF from the destination and ensuring that the Multi Command Command Class is listed as "supported".

For a device responding, this is automatically ensured in situation a) above. However, in situation b) above, or where a device is configured to send responses (e.g. reports) to a 3rd device, for instance via the Association Command Class, then the responding device MUST ensure the destination device understands the Multi Command encapsulation as described above.

For clarification it SHOULD be emphasized that a part of implementing the Multi Command Command Class as "controller" the device MUST also be able to decode the encapsulated responses that MAY come back as a result of sending encapsulated request (e.g. get Command to retrieve a report).

A device that supports the Multi Command Command Class MUST be able to receive and interpret the encapsulated form of all the command classes that it list in the NIF and supports in the normal non-encapsulated form, except of course the Multi Command Command Class itself.

4.58.1 Multi Command Encapsulated Command

The Multi Command Encapsulated Command used to contain multiple Commands. The encapsulated Commands MUST be executed in the order they are received. In case get Commands in a Multi Command Encapsulated Command are received by a device the reports MUST be replied in a Multi Command Encapsulated Command in the same order as the gets were received. Be aware of the payload limitations with respect to a routed single cast frame [1].

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CMD							
Command = MULTI_CMD_ENCAP							
Number of Commands							
Command Length 1							
Command Class 1							
Command 1							
Data 1,1							
Data 1,2							
…							
Data 1,N							
…							
Command Length X							
Command Class X							
Command X							
Data X,1							
Data X,2							
…							
Data X,N							

Number of Commands (8 bits)

This field indicates the number of encapsulated Commands.

Command Length (8 bits)

The Command Length field indicates the number of bytes used by the following encapsulated Command including Command Class identifiers, Command identifiers and the data.

Command Class (8 bits)

The Command Class field indicates the command class identifier of the encapsulated Command.

Command (8 bits)

The Command field indicates the Command identifier of the encapsulated Command.

CONFIDENTIAL

Data 1-N (variable)

The Data fields in the encapsulated Command of the respective command class and Command identifiers.

4.58.2 Example

This example shows how a battery operated device can be instructed to go to sleep immediately after an internal parameter have been changed by one frame using the Multi Command Encapsulated Command.

7	6	5	4	3	2	1	0
COMMAND_CLASS_MULTI_CMD							
MULTI_CMD_ENCAP							
Number of Commands = 0x02							
Length = 0x05							
COMMAND_CLASS_CONFIGURATION							
CONFIGURATION_SET							
Parameter Number = 0x01							
Size = 0x01							
Configuration Value = 0x07							
Length = 0x02							
COMMAND_CLASS_WAKE_UP							
WAKE_UP_NO_MORE_INFORMATION							

In this example the Multi Command Encapsulated Command contains two encapsulated Commands. First the internal parameter no. 1 is changed to 0x07 by the Configuration Set Command and afterwards the Wake Up No More Information Command instructs the battery operated device to go to sleep immediately after, thereby minimizing the power consumption.

CONFIDENTIAL

4.59 Multi Channel Association Command Class, version 2

This Multi Channel Association Command Class used to enable bindings to nodes comprising of a number of end points. The command class can handle nodes with and without end points.

The Association command class, version 1 is a subset of the Multi Channel Association command class and the data intersection of the two command classes MUST be mirrored on application level allowing access to the same data via both command classes. A device supporting the Multi Channel Association command class MUST therefore support the Association command class, version 2 to fulfill the above requirement.

NOTE: **A device supporting the Multi Channel Association command class MUST support the Association command class, version 2.**

NOTE: **The Association Report will only return node IDs for nodes without end points specified. To obtain associations configured for nodes with and without end points request the Multi Channel Association Report command.**

CONFIDENTIAL

4.59.1   Multi Channel Association Set Command

The Multi Channel Association Set Command used to add nodes with/without end points to a given grouping identifier. The node receiving the set command SHOULD add the nodes received to the nodes already associated by this grouping until the grouping is full.

Note: Remember that routing slaves also MUST have assigned return routes by a controller using the API call ZW_AssignReturnRoute [1] to all the associated nodes. Delete all return routes by the API call ZW_DeleteReturnRoute before assignment. This is of course necessary for all the associated nodes out of direct range but SHOULD always be done default to all the associated nodes to create a reliable and robust network. It's OPTIONAL whether the return routes are assigned before or after the Multi Channel Association Set command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION							
Command = MULTI_CHANNEL_ASSOCIATION_SET							
Grouping identifier							
Node ID1							
Node ID2							
...							
Node ID n							
Marker = MULTI_CHANNEL_ASSOCIATION_SET_MARKER							
Multi Channel Node ID1							
Bit address	End Point						
Multi Channel Node ID2							
Bit address	End Point						
…							
Multi Channel Node IDn							
Bit address	End Point						

Grouping identifier (8 bits)

This grouping identifier used to instruct how nodes are grouped together. The grouping identifier values MUST be a sequence starting from 1. This field can be ignored in case the node only supports one grouping.

NodeID1 .. NodeIDn

These fields contain a list of node IDs that SHOULD be associated with the grouping.

Marker (8 bits)

This marker identifier used to separate between nodes without and with end points attached. This field can be omitted in case no Multi Channel node follows.

Multi Channel NodeID1 .. Multi Channel NodeIDn

These fields contain a list of node IDs with End Points attached that SHOULD be associated with the grouping.

Bit address (1 bit)

This bit is set to 1 if the end point(s) is given in a bit mask which makes it possible to address end points in parallel.

Note: Only the first 7 end points are bit addressable.

This bit is set to 0 if the end point is addressed individually.

End Point (7 bits)

The end point(s) that SHOULD receive the command. This field MUST be interpreted based in the "Bit address" value:

Bit address equals 1: Bit 0 is End Point 1, bit 1 is End Point 2 … bit 6 is End Point 7

Bit address equals 0: End Point addresses an individual End Point of the device. Valid values are 1 to 127.

Note: The same node id can occur several times in the Multi Channel Nodes ID list, if associations to several end points are desired.

4.59.2 Multi Channel Association Get Command

The Multi Channel Association Get Command is used to request the current association configuration of a given grouping identifier. The node receiving this command SHOULD answer with Multi Channel Association Report command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION							
Command = MULTI_CHANNEL_ASSOCIATION_GET							
Grouping Identifier							

Grouping identifier (8 bits)

This grouping identifier used to identify how nodes are grouped together. The grouping identifier values MUST be a sequence starting from 1. This field can be ignored in case the node only supports one grouping.

CONFIDENTIAL

4.59.3 Multi Channel Association Report Command

The Multi Channel Association Report Command SHOULD be used to report all nodes associated with the matching grouping identifier. Be aware that it's only possible to get information about how many groupings a given node are associated to, but not the total number of groupings. Therefore the Grouping Identifiers SHOULD be allocated with care starting from 0x01 to avoid unnecessary overhead in finding the groupings with associations. A remote with up to 6 different groupings there are controlled by 6 buttons numbered 1...6 could use the same numbers as grouping identifiers. The Multi Channel Association Report command can be send unsolicited or requested by a Multi Channel Association Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION								
Command = MULTI_CHANNEL_ASSOCIATION_REPORT								
Grouping Identifier								
Max Nodes Supported								
Reports to Follow								
NodeID1								
Node ID2								
…								
Node IDn								
Marker = MULTI_CHANNEL_ASSOCIATION_REPORT_MARKER								
Multi Channel Node ID1								
Bit address	End Point							
Multi Channel Node ID2								
Bit address	End Point							
…								
Multi Channel Node IDn								
Bit address	End Point							

Grouping identifier (8 bits)

Refer to description under the Multi Channel Association Set command.

Max Nodes Supported (8 bits)

Maximum number of nodes grouping identifier above supports.

CONFIDENTIAL

Reports to Follow (8 bits)

This value indicates how many report frames left before transferring the entire list of node IDs associated with the given grouping identifier

NodeID1 .. NodeIDn

Refer to description under the Multi Channel Association Set command.

Marker (8 bits)

Refer to description under the Multi Channel Association Set command.

Multi Channel NodeID1 .. Multi Channel NodeIDn

Refer to description under the Multi Channel Association Set command.

Bit address (1 bit)

Refer to description under the Multi Channel Association Set command.

End Point (7 bits)

Refer to description under the Multi Channel Association Set command.

CONFIDENTIAL

4.59.4 Multi Channel Association Remove Command

The Multi Channel Association Remove Command used to remove nodes from a given grouping.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION							
Command = MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier							
Node ID1							
Node ID2							
..							
Node IDn							
Marker = MULTI_CHANNEL_ASSOCIATION_REMOVE_MARKER							
Multi Channel Node ID1							
Bit address	End Point						
Multi Channel Node ID2							
Bit address	End Point						
…							
Multi Channel Node IDn							
Bit address	End Point						

Grouping identifier

This grouping identifier used to determine from which grouping the supplied node ID's SHOULD be removed. If no group ID or group ID zero is supplied all groups SHOULD be cleared.

NodeID1 .. NodeIDn

These fields contain a list of node ID's that SHOULD be removed from the specified grouping identifier. In case no node ID's are supplied the whole grouping SHOULD be cleared.

Marker (8 bits)

Refer to description under the Multi Channel Association Set command.

Multi Channel NodeID1 .. Multi Channel NodeIDn

Refer to description under the Multi Channel Association Set command.

Bit address (1 bit)

Refer to description under the Multi Channel Association Set command.

End Point (7 bits)

Refer to description under the Multi Channel Association Set command.

CONFIDENTIAL

4.59.4.1   Examples

Remove specified Node ID from Group 3

<u>Get association settings for Group 3</u>

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_GET							
Grouping identifier = 3							

<u>Current association settings for Group 3</u>

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 16							
End Point = 3							
Multi Channel Node ID = 17							
End Point = 2							

<u>Remove Node ID = 12 from Group 3</u>

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 3							
Node ID = 12							

<u>New association settings for Group 3</u>

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 16							
End Point = 3							
Multi Channel Node ID = 17							
End Point = 2							

CONFIDENTIAL

Remove specified End Point from Group 3

Get association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_GET							
Grouping identifier = 3							

Current association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 16							
End Point = 3							
Multi Channel Node ID = 17							
End Point = 2							

Remove Multi Channel Node ID = 16,3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 3							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 3							

New association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 17							
End Point = 2							

CONFIDENTIAL

Remove bit addressable End Points from Group 3

Get association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_GET							
Grouping identifier = 3							

Current association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 134 (b1000 0110)							
Multi Channel Node ID = 17							
End Point = 2							

Remove Multi Channel Node ID = 16,2 and 16,3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 3							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 134 (b1000 0110)							

Note! The Multi Channel Node MUST report end points as bit addressable in order to be removed in the same manners.

New association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 17							
End Point = 2							

CONFIDENTIAL

Remove specified Node ID and End Point from Group 3

Get association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_GET							
Grouping identifier = 3							

Current association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 16							
End Point = 3							
Multi Channel Node ID = 17							
End Point = 2							

Remove Node ID = 12 & Multi Channel Node ID = 16,3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 3							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 3							

New association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 17							
End Point = 2							

CONFIDENTIAL

Remove all associations from Group 3

Get association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_GET							
Grouping identifier = 3							

Current association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							
Node ID = 11							
Node ID = 12							
MULTI_CHANNEL_ ASSOCIATION_REPORT_MARKER							
Multi Channel Node ID = 16							
End Point = 2							
Multi Channel Node ID = 16							
End Point = 3							
Multi Channel Node ID = 17							
End Point = 2							

Remove all Node IDs from Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 3							

New association settings for Group 3

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REPORT							
Grouping Identifier = 3							
Max Nodes Supported = 5							
Reports to Follow = 0							

Remove specific Node ID from all Groups

7	6	5	4	3	2	1	0
COMMAND_CLASS_ MULTI_CHANNEL_ASSOCIATION							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 0							
Node ID = 12							

CONFIDENTIAL

Remove specific Multi Channel Node ID from all Groups

7	6	5	4	3	2	1	0
COMMAND_CLASS_ **MULTI_CHANNEL_ASSOCIATION**							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 0							
MULTI_CHANNEL_ **ASSOCIATION_REPORT_MARKER**							
Multi Channel Node ID = 16							
End Point = 3							

Remove all associations from all Groups

7	6	5	4	3	2	1	0
COMMAND_CLASS_ **MULTI_CHANNEL_ASSOCIATION**							
MULTI_CHANNEL_ASSOCIATION_REMOVE							

or

7	6	5	4	3	2	1	0
COMMAND_CLASS_ **MULTI_CHANNEL_ASSOCIATION**							
MULTI_CHANNEL_ASSOCIATION_REMOVE							
Grouping identifier = 0							

4.59.5 Multi Channel Association Supported Groupings Get Command

The Multi Channel Association Supported Groupings Get Command is used to request the number of groupings that this node supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION							
Command = MULTI_CHANNEL_ASSOCIATION_GROUPINGS_GET							

CONFIDENTIAL

4.59.6 Multi Channel Association Supported Groupings Report Command

The Multi Channel Association Supported Groupings Report Command used to report the maximum number of groupings the given node supports. The Multi Channel Association Supported Groupings Report command can be send unsolicited or requested by a Multi Channel Association Supported Groupings Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL_ASSOCIATION							
Command = MULTI_CHANNEL_ASSOCIATION_GROUPINGS_REPORT							
Supported Groupings							

Supported Groupings (8 bits)

The number of groupings this node supports.

CONFIDENTIAL

4.60   Multi Channel Command Class, version 3

This section contains commands that can be used to control a Multi Channel Command Class.

The Multi Channel command class used to control one or more end points in a given device that supports this command class. It is REQUIRED that an application using this command class sets the Optional Functionality bit in the NIF.

Multi Channel devices can have from 1 to 127 end points that can be controlled individually. The first 7 end points can be controlled together in any combination. This command class defines how the capabilities of the end points are communicated in a Z-Wave network and how the end points are addressed when other nodes needs to communicate with them.

It is up to the application developer to define how a Multi Channel device reacts to un-encapsulated commands. The reaction MUST however be within these guidelines:

- Un-encapsulated commands MUST trigger an action and/or response that accurately reflect the basic functionality of the Multi Channel device in question[2]. This MUST be enforced to preserve backwards compatibility with simple controllers not able to address individual end points.

- Un-encapsulated commands MUST NOT limit the functionality, or enable non-compliant behavior by any End Point in the device.

- For simplicity un-encapsulated command MUST be forwarded to End Point 1 in the Multi Channel device. Therefore End Point capabilities of End Point 1 MUST be reported in the standard Z-Wave NIF. **Note:** End point 1 can not be a dynamic end point (more details see 4.60.2).

Announce only Multi Channel Command Class support on the highest NIF level to avoid recursion problems.

4.60.1   Multi Channel End Point Get Command

The Multi Channel End Point Get Command used to get the number of end points embedded in a single node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_END_POINT_GET							

---

[2] *E.g. for a power strip with 5 outlets designed as a Multi Channel device, an un-encapsulated Basic Off command could turn off all outlets in the power strip, and a un-encapsulated Basic On could turn On all outlets.*

CONFIDENTIAL

4.60.2 Multi Channel End Point Report Command

The Multi Channel End Point Report Command reports the number of end points embedded in a single node. The Multi Channel End Point Report command can only be send as a result of receiving a Multi Channel End Point Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_END_POINT_REPORT							
Dyna-mic	Iden-tical	Res					
Res	End Points						

Res (6 bits) and (1 bit)

These bits are reserved bit. MUST be set to 0 by devices sending this frame and MUST be ignored by devices receiving this frame.

End Points (7 bits)

Number of end points embedded in the node. The maximum number of end points is 127.

Identical (1 bit)

This bit is set to 1 if all the end points in the node has the same generic and specific command class and supports the same OPTIONAL command classes.

Dynamic (1 bit)

This field is set to 1 if the device has a dynamic number of end points. When the dynamic bit is set the number of end points in the device can change over time.

Warning: Care SHOULD be taken when communicating with dynamic end points as the transmitter cannot be entirely sure the specific end point exists.

Note: Implementation of a device supporting dynamic end points MUST follow the guide lines given in 4.60.8.2. Future definitions MAY be defined, always refer to the newest version of the command class specification.

CONFIDENTIAL

4.60.3   Multi Channel Capability Get Command

The Multi Channel Capability Get Command used to get the capabilities of the end points in a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_CAPABILITY_ GET							
Res	End Point						

Res (1 bit)

This bit is reserved bit. MUST be set to 0 by devices sending this frame and MUST be ignored by devices receiving this frame.

End Point (7 bits)

End Point number to get capabilities from.

4.60.4   Multi Channel Capability Report Command

The Multi Channel Capability Report Command reports the generic and specific device class of the end point and the supported command classes of the end point. The Multi Channel Capability Report command can be send unsolicited or requested by a Multi Channel Command Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_CAPABILITY_REPORT							
Dyna-mic	End Point						
Generic Device Class							
Specific Device Class							
Command Class 1							
Command Class 2							
…							
Command Class N							

End Point (7 bits)

The End point field indicates what end point number the report frame is referring to.

Dynamic (1 bit)

This field is set to one if this end point is a dynamic end point. When this bit is set in an end point it can not be assumed that it will reply to commands send to it because it could be gone again when a command is send to it.

Note: End point 1 can not be dynamic

CONFIDENTIAL

Generic Device class (8 bits)

The generic device class of the specified end point.

Specific Device class (8 bits)

The specific device class of the specified end point.

Command Class 1 .. Command Class N (N*8 bits)

Command classes supported or controlled by the device in question. The number of fields transmitted can be determined from the length field in the frame.

Note: For memory reasons it is not REQUIRED for a controlling device to save the capabilities of each end point. However controlling devices SHOULD be able to control with at least the basic command class for each end point.

4.60.5　Multi Channel End Point Find Command

The Multi Channel End Point Find Command used to find end points in a device with a given set of generic and specific device class.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_END_POINT_FIND							
Generic Device Class							
Specific Device Class							

Generic Device Class (8 bits)

The generic device class that SHOULD be found in the Multi Channel device.

Specific Device Class (8 bits)

The specific device class that SHOULD be found in the Multi Channel device. If 0xFF is specified in this field then all devices with the specified generic device class will be returned.

CONFIDENTIAL

4.60.6   Multi Channel End Point Find Report Command

The Multi Channel End Point Find Report Command used to reply to a Multi Channel End Point Find command. This command can only be send as a response to a Multi Channel End Point Find command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_END_POINT_FIND_REPORT							
Reports to Follow							
Generic Device Class							
Specific Device Class							
Res	End Point 1						
…							
Res	End Point n						

Reports to Follow (8 bits)

This value indicates how many report frames left before transferring the entire list of end points matching the given generic and specific device class.

Generic Device Class (8 bits)

The generic device class that SHOULD be found in the Multi Channel device.

Specific Device Class (8 bits)

The specific device class that SHOULD be found in the Multi Channel device. If 0xFF is specified in this field then all devices with the specified generic device class will be returned.

Res (1 bit)

This bit is reserved bit. MUST be set to 0 by devices sending this frame and MUST be ignored by devices receiving this frame.

End Point 1 … n (7 bits)

The end point(s) that matches the generic and specific device class send in the get command.

Note: If the receiver of Multi Channel End Point Find Command had no support for the queried Generic Device Class and/or Specific Device Class, it will return a Multi Channel End Point Report Command with End Point 1 = 0.

Example: Device not supporting Generic Device Class equal to Multilevel Switch and Specific Device Class equal to Motor Control Class C returns:

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_END_POINT_FIND_REPORT							
Reports to Follow = 0							
Generic Device Class = Multilevel Switch							
Specific Device Class = Motor Control Class C							
Res=0	End Point 1 = 0						

CONFIDENTIAL

4.60.7 Multi Channel Command Encapsulation Command

The Multi Channel Command Encapsulation Command used to encapsulate commands send to a Multi Channel device so it can address one or several end points in a Z-Wave node. Any command that the end point reports that it supports can be encapsulated using this command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_MULTI_CHANNEL							
Command = MULTI_CHANNEL_CMD_ENCAP							
Res	Source End Point						
Bit address	Destination End Point						
Command Class							
Command							
Parameter 1							
...							
Parameter N							

Res (1 bit)

This bit is reserved bit. MUST be set to 0 by devices sending this frame and MUST be ignored by devices receiving this frame.

Source End Point (7 bits)

This field indicates the end point from where the command was send. Valid Source End Points are 1 to 127. The Source End Point MUST be used as destination in the case the encapsulated frame contains a request to the destination.

If the sending device does not support multiple channels the Source End Point MUST be set to 0. Please note that in this case a response to a GET command will be send encapsulated back to End Point 0.

Bit address (1 bit)

This bit is set to 1 if the end point(s) is given in a bit mask which makes it possible to address end points in parallel.

This bit is set to 0 if the end point is addressed individually.

Note: Only the first 7 end points are bit addressable.

Note: If the encapsulated command is a request (requiring a reply from the destination) it is not allowed to bit address the frame. This is prohibited to decrease implementation complexity of devices supporting this command class.

CONFIDENTIAL

Destination End Point (7 bits)

The end point(s) that SHOULD receive the command. This field MUST be interpreted based in the "Bit address" value:

Bit address equals 1: Bit 0 is End Point 1, bit 1 is End Point 2 … bit 6 is End Point 7

Bit address equals 0: End Point addresses an individual End Point of the device. Valid values are 1 to 127.

Command Class (8 bits)

Command class identifier of embedded command class.

Command (8 bits)

Command identifier of embedded command class.

Parameter 1 .. Parameter N (variable)

Parameters attached to embedded command class. The number of fields transmitted can be determined from the length field in the frame.

Note: All command classes and commands can be encapsulated; however care SHOULD be taken when encapsulating management command such as configuration and association command classes. These will only affect the end point addressed and not the device in whole.

Note: If the command encapsulated in this frame is a request, the reply MUST be encapsulated in a Multi Channel Command Encapsulation frame.

Note: It is allowed to encapsulate this command into another type of encapsulation, and to encapsulate another type of encapsulation into this command. BUT it is not allowed to have multiple encapsulations of the same type e.g. Multi Channel Command Encapsulations or Multi Command Encapsulations nested into each other.

4.60.8 Implementation Recommendations

4.60.8.1 Supporting Devices

A device supporting Multi Channel command class MUST "answer as asked". There are two possible ways of sending a report to a request:

A.) Non-encapsulated frames
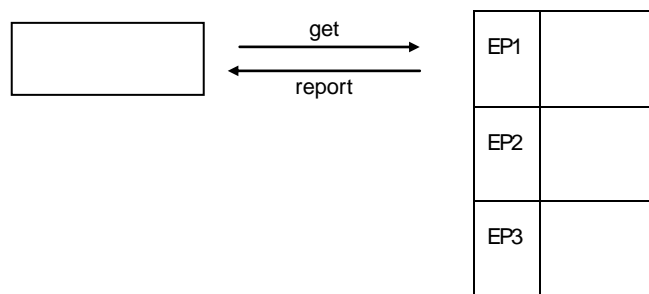
B.) Multi Channel Command Encapsulation

CONFIDENTIAL
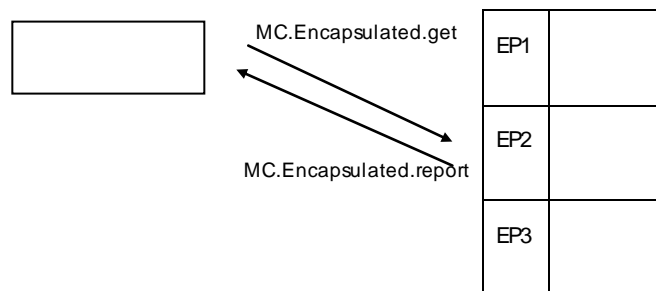
4.60.8.2   Supporting Dynamic End Points

When implementing a device supporting dynamic End Points the following guide lines MUST be followed for adding and removing End Points.

Adding an End Point

When adding a new End Point it MUST be numbered in succession. If the next consecutive number is 128, the device MUST search from the start of the list for the first empty entry.

Examples adding end points

The following examples show how End Points are inserted into the list of End Points.

<u>Adding an End Point</u>

End Point list before   End Point list after

1	Occupied
2	Occupied
3	Free
…	
126	Free
127	Free

1	Occupied
2	Occupied
3	Occupied
…	
126	Free
127	Free

<u>Adding an End Point</u>

End Point list before   End Point list after

1	Occupied
2	Free
3	Occupied
…	
126	Occupied
127	Free

1	Occupied
2	Free
3	Occupied
…	
126	Occupied
127	Occupied

<u>Adding an End Point</u>

End Point list before   End Point list after

1	Occupied
2	Free
3	Occupied
…	
126	Occupied
127	Occupied

1	Occupied
2	Occupied
3	Occupied
…	
126	Occupied
127	Occupied

Removing an End Point

When removing an End Point, the End Point MUST leave an empty spot in the End Point list, thus making sure there is no change in the numbering of the other End Points supported by the device.

Example removing end point

The following example show how End Points are removed from the list of End Points.

Remove End Point #2

End Point list before

1	Occupied
2	Occupied
3	Occupied
…	
126	Free
127	Free

End Point list after

1	Occupied
2	Free
3	Occupied
…	
126	Free
127	Free

CONFIDENTIAL

4.61 Multilevel Sensor Command Class, version 1-4

This section contains Commands that can be used to control a multilevel sensor.

4.61.1 Multilevel Sensor Get Command

The Multilevel Sensor Get Command is used to request the level of a multilevel sensor. The Multilevel Sensor Get Command versions 1-3 have the same layout.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_GET							

4.61.2 Multilevel Sensor Report Command

This Command can be used by a multilevel sensor to send a report either unsolicited or requested by the Multilevel Sensor Get Command. The Multilevel Sensor Report Command version 2 and 3 are extensions with respect to Sensor Types and associated Scales.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_REPORT							
Sensor Type							
Precision			Scale		Size		
Sensor Value 1							
Sensor Value 2							
..							
Sensor Value n							

Sensor Type (8 bits)

Sensor type specifies what type of sensor this Command originates from. Refer to Multilevel Sensor version 5 with respect to defined sensors. New sensor types/values can be requested from Sigma Designs.

Precision (3 bits)

The precision field describes what the precision of the sensor value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bits)

The Scale used to indicate what unit the sensor uses. Refer to Multilevel Sensor version 5 with respect to defined scales for the relevant sensors. New scales/values can be requested from Sigma Designs.

CONFIDENTIAL

Size (3 bits)

The size field indicates the number of bytes that used for the sensor value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

Sensor Value (variable)

The sensor value is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal	Signed 2 bytes decimal value	Hexadecimal
127	0x7F	32767	0x7FFF
25	0x19	1025	0x0401
2	0x02	2	0x0002
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-2	0xFE	-2	0xFFFE
-25	0xE7	-1025	0xFBFF
-128	0x80	-32768	0x8000

Notice: The device receiving the Multilevel Sensor Report MUST always show the sensor value even though the Sensor Type and/or Scale are not supported.

CONFIDENTIAL

4.62   Multilevel Sensor Command Class, Version 5

The Multilevel Sensor Command Class is used to interview a multilevel sensor for its capabilities and to retrieve the measured values. Version 5 of this command class is extended with the following functionalities:

- A "get-supported" mechanism for the controlling device to interview the multilevel sensor for its supported sensor types and/or scales

- Additional sensor type and scale fields to the Multilevel Sensor Get command to request for a specific sensor report.

- Additional sensor types and/or scales to the list of multilevel sensors

4.62.1   Multilevel Sensor Get Supported Sensor Command

This command is introduced in v5 of the Multilevel Sensor command class and used to retrieve the supported sensor types from the multilevel sensor device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_GET_SENSOR							

4.62.2   Multilevel Sensor Supported Sensor Report Command

This command is introduced in v5 of the Multilevel Sensor command class and MUST be sent as requested by a received Multilevel Sensor Get Supported Sensor command. This command indicates the supported sensor types of the multilevel sensor device in a bit mask format and SHALL NOT be transmitted unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_SENSOR_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask 1 … Bit Mask N

The Bit Mask field describes the supported sensor types by the multilevel sensor device and refers to the Sensor Type table of Multilevel Sensor Report command.

CONFIDENTIAL

First bit (Bit 0) in Bit Mask 1 indicates support for Sensor Type = 0x01 (Temperature #1). If this bit is set to 1, the multilevel sensor device supports Temperature Sensor, if this bit is set to 0, there is no support for Temperature Sensor.

Second bit (Bit 1) in Bit Mask 1 indicates support for Sensor Type = 0x02 (General Purpose #2) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the Bit Mask N indicating the last supported Sensor Type. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

Example:

To indicate sensor type support for Relative Humidity, Luminance and Temperature the Multilevel Sensor Supported Sensor Report command MUST be structured as illustrated below.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_SENSOR_REPORT							
Reports to follow = 0							
Bit Mask 1							
0	0	0	1	0	1	0	1

4.62.3 Multilevel Sensor Get Supported Scale Command

This command is introduced in v5 of the Multilevel Sensor command class and used to retrieve the supported scales of the specific sensor type from the Multilevel Sensor device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_GET_SCALE							
Sensor Type							

Sensor Type (8 bit)

See Multilevel Sensor Report command.

4.62.4 Multilevel Sensor Supported Scale Report Command

This command is introduced in v5 of the Multilevel Sensor command class and MUST be sent as requested by a received Multilevel Sensor Get Supported Scale command. This command indicates the supported scales of the requested sensor type in a bit mask format and SHALL NOT be transmitted unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_SCALE_REPORT							
Sensor Type							
Reserved				Scale Bit Mask			

Sensor Type (8 bit)

See Multilevel Sensor Report command.

Scale Bit Mask (4 bit)

The Scale Bit Mask field describes the supported scales for the requested sensor type by the multilevel sensor device and refers to the Scale table of Multilevel Sensor Report command.

First bit (Bit 0) in Scale Bit Mask indicates support for the first Scale of the requested Sensor Type e.g. if Sensor Type = Voltage and the Bit 0 = 1, then the Volt scale is supported.

Second bit (Bit 1) in Scale Bit Mask indicates support for the second scale and so forth.

Reserved (4 bit)

This field is reserved and MUST be zero.

Example:

To indicate scale support for mV and V of a sensor type = Voltage, the Multilevel Sensor Supported Scale Report command MUST be structured as illustrated below.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_SUPPORTED_SCALE_REPORT							
Sensor Type = Voltage							
Reserved				Scale Bit Mask			
0	0	0	0	0	0	1	1

4.62.5 Multilevel Sensor Get Command

In contrary to v1-4, v5 of Multilevel Sensor command class add fields to the Multilevel Sensor Get Command to request for the specific and supported sensor type and scale for its measured value. Versions prior to v5 do not specify the additional sensor type and scale fields, why the Multilevel Sensor device can only support one type of sensor when implementing solely Multilevel Sensor command class v1-4.

NOTICE: It is the responsibility of the transmitter to ensure the receiver in fact supports the requested Sensor Types and/or Scales. This is done through the interview process using Multilevel Sensor Get Supported Sensor/Scale commands. If a Sensor Type and/or Scale is not supported by the receiver, it will rightfully reply with a manufacture-selected and factory default Multilevel Sensor Report command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_GET							
Sensor Type							
Reserved			Scale		Reserved		

Sensor Type (8 bit) & Scale (2 bit)

See Multilevel Sensor Report command.

Reserved (3 bit + 3 bit)

These fields are reserved and MUST be zero.

4.62.6 Multilevel Sensor Report Command

This command can be used by a multilevel sensor to send a Multilevel Sensor Report Command either unsolicited or requested by the Multilevel Sensor Get Command. The Multilevel Sensor Report Command v5 is extended with additional Sensor Types and the associated Scales.

NOTICE: In case the received Multilevel Sensor Get command contains Sensor Type and/or Scale values not supported by the device, then this device MUST reply with a manufacture-selected and factory default Multilevel Sensor Report command. This behavior MUST also be implemented if the received Multilevel Sensor Get command is from command class versions prior to Multilevel Sensor Command Class v5.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_MULTILEVEL							
Command = SENSOR_MULTILEVEL_REPORT							
Sensor Type							
Precision			Scale		Size		
Sensor Value 1							
..							
Sensor Value n							

CONFIDENTIAL

Sensor Type (8 bit)

Sensor type specifies what type of sensor this command originates from. Refer to the table below with respect to defined sensors. Values not defined below are reserved and MUST NOT be used. New sensor types/values can be requested from Sigma Designs Technology Denmark.

Sensor Type	Value	Sensor Type	Value
Reserved	0x00	Current (v3)	0x10
Air temperature (v1)	0x01	CO_2-level (v3)	0x11
General purpose value (v1)	0x02	Air flow (v3)	0x12
Luminance (v1)	0x03	Tank capacity (v3)	0x13
Power (v2)	0x04	Distance (v3)	0x14
Humidity (v2 + v5)	0x05	Angle Position (v4)	0x15
Velocity (v2)	0x06	Rotation (v5)	0x16
Direction (v2)	0x07	Water temperature (v5)	0x17
Atmospheric pressure (v2)	0x08	Soil temperature (v5)	0x18
Barometric pressure (v2)	0x09	Seismic intensity (v5)	0x19
Solar radiation (v2)	0x0A	Seismic magnitude (v5)	0x1A
Dew point (v2)	0x0B	Ultraviolet (v5)	0x1B
Rain rate (v2)	0x0C	Electrical resistivity (v5)	0x1C
Tide level (v2)	0x0D	Electrical conductivity (v5)	0x1D
Weight (v3)	0x0E	Loudness (v5)	0x1E
Voltage (v3)	0x0F	Moisture (v5)	0x1F

Precision (3 bit)

The precision field describes what the precision of the sensor value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bit)

The Scale is used to indicate what unit the sensor uses. Refer to the table below with respect to defined scales for the relevant sensors. New scales/values can be requested from Sigma Designs Technology Denmark.

Sensor Type	Scale	Value
Air temperature (v1)	Celsius (C)	0x00
	Fahrenheit (F)	0x01
	Reserved	0x02-0x03
General purpose (v1)	Percentage value	0x00
	Dimensionless value	0x01
	Reserved	0x02-0x03

CONFIDENTIAL

Luminance (v1)	Percentage value	0x00
	Lux	0x01
	Reserved	0x02-0x03
Power (v2)	W	0x00
	Btu/h	0x01
	Reserved	0x02-0x03
Humidity (v2 & V5)	Percentage value	0x00
	Absolute humidity (g/m³) – (v5)	0x01
	Reserved	0x02-0x03
Velocity (v2)	m/s	0x00
	Mph	0x01
	Reserved	0x02-0x03
Direction (V2)	0 to 360 degrees. 0 = no wind, 90 = east, 180 = south, 270 = west, and 360 = north	0x00
	Reserved	0x01-0x03
Atmospheric pressure (v2)	kPa (kilopascal)	0x00
	Inches of Mercury	0x01
	Reserved	0x02-0x03
Barometric pressure (v2)	kPa (kilopascal)	0x00
	Inches of Mercury	0x01
	Reserved	0x02-0x03
Solar radiation (v2)	W/m²	0x00
	Reserved	0x01-0x03
Dew point (v2)	Celsius (C)	0x00
	Fahrenheit (F)	0x01
	Reserved	0x02-0x03
Rain rate (v2)	mm/h	0x00
	in/h	0x01
	Reserved	0x02-0x03
Tide level (v2)	m	0x00
	Feet	0x01
	Reserved	0x02-0x03
Weight (v3)	Kg	0x00
	pounds	0x01
	Reserved	0x02-0x03

Voltage (v3)	V	0x00
	mV	0x01
	Reserved	0x02-0x03
Current (v3)	A	0x00
	mA	0x01
	Reserved	0x02-0x03
CO_2-level (v3)	Ppm	0x00
	Reserved	0x01-0x03
Air flow (v3)	m_3/h	0x00
	cfm (cubic feet per minute)	0x01
	Reserved	0x02-0x03
Tank capacity (v3)	l (liter)	0x00
	cbm (cubic meter)	0x01
	gallons	0x02
	Reserved	0x03
Distance (v3)	M	0x00
	Cm	0x01
	Feet	0x02
	Reserved	0x03
Angle Position (v4)	Percentage value	0x00
	Degrees relative to north pole of standing eye view	0x01
	Degrees relative to south pole of standing eye view	0x02
	Reserved	0x03
Rotation (v5)	rpm (revolutions per minute)	0x00
	Hz (Hertz)	0x01
	Reserved	0x02-0x03
Water temperature (v5)	Celsius (C)	0x00
	Fahrenheit (F)	0x01
	Reserved	0x02-0x03
Soil temperature (v5)	Celsius (C)	0x00
	Fahrenheit (F)	0x01
	Reserved	0x02-0x03
Seismic intensity (v5)	Mercalli	0x00
	European Macroseismic	0x01
	Liedu	0x02

	Shindo	0x03
Seismic magnitude (v5)	Local (M_L)	0x00
	Moment (M_W)	0x01
	Surface wave (M_S)	0x02
	Body wave (M_B)	0x03
Ultraviolet (v5)	UV index	0x00
		0x01-0x03
Electrical resistivity (v5)	ohm metre (Ωm)	0x00
		0x01-0x03
Electrical conductivity (v5)	siemens per metre (S·m−1)	0x00
		0x01-0x03
Loudness (v5)	Absolute loudness (dB)	0x00
	A-weighted decibels (dBA)	0x01
		0x02-0x03
Moisture (v5)	Percentage value	0x00
	Volume water content (m^3/m^3)	0x01
	Impedance (kΩ)	0x02
	Water activity (a_w)	0x03

Size (3 bit)

The size field indicates the number of bytes that is used for the sensor value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

Sensor Value (variable)

The sensor value is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal	Signed 2 bytes decimal value	Hexadecimal
127	0x7F	32767	0x7FFF
25	0x19	1025	0x0401
2	0x02	2	0x0002
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-2	0xFE	-2	0xFFFE
-25	0xE7	-1025	0xFBFF
-128	0x80	-32768	0x8000

NOTICE: The device receiving the Multilevel Sensor Report MUST always show the sensor value <u>as is</u> even though the Sensor Type and/or Scale are not supported.

CONFIDENTIAL

4.63   Multilevel Switch Command Class, version 1

This section contains Commands that can be used to control a multilevel switch. These Commands allow applications to turn a multilevel switch on/off, start/stop dimming/operation, jumping/dimming to a specified level, and read the current level.

4.63.1   Multilevel Switch Set Command

The Multilevel Switch Set Command version 1 can be used to set the level in a device that supports the multilevel switch functionality. The speed that the switch increases or decreases the level with is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value							

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). Furthermore it can take values from 1 to 99 (0x01 – 0x63). The unit of the value is implementation specific i.e. percentage. Devices SHALL implement all values.

The value 0xFF (on/enable) will cause the device to change to the level when the device last was turned on. With respect to motor controls both 0x63 and 0xFF MAY be interpreted as fully open.

The values 100…254 (0x64…0xFE) are reserved and SHALL be ignored by the receiving device.

Controlling devices MAY send any of the values 0x00…0x63 and 0xFF to the device. Controlling devices MUST NOT send any of the reserved values to the device.

Devices receiving this Command are free in the mapping of the received values 1…99 (0x01…0x63) to an internal representation of their setting, except that for a mapping to percentages the value 0x63 represents 100%. The mapping of the value received SHOULD be monotonous i.e. a higher value in a set Command SHOULD always result in either a higher or same level.

It is not REQUIRED that a device implements 100 distinct setting levels. If a device implements less than 100 internal setting levels, then the *active* internal setting levels SHOULD be spread equally over the entire range of 1…99 (0x01…0x63).

Example:
RECOMMENDED behavior of a device implementing three light levels:

0	OFF	(inactive / OFF setting)
1…33	1/3 On	(3 *active* settings → spread over 99 values)
34…66	2/3 On	
67…99	FULL ON	

CONFIDENTIAL

4.63.2   Multilevel Switch Get Command

The Multilevel Switch Get Command version 1 can be used to request the status of a multilevel switch.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_GET							

4.63.3   Multilevel Switch Report Command

The Multilevel Switch Report Command version 1 can be send unsolicited or requested by the Multilevel Switch Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_REPORT							
Value							

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). Furthermore it can return values from 1 to 99 (0x01 – 0x63). Devices MUST NOT respond with a Multilevel Switch Report with any other value.

It is neither specified that a device will respond in a Report with exactly the value sent in a Set Command, nor that a device will ever use all permitted values in Reports.

Note:
If a controlling device is implementing the dimming Commands and want to align all the dimmers when dimming is stopped requires extra precaution: The get Command used to enquire the end level from one device MUST be from a Multilevel Switch to avoid any unintentionally jumps in light level.

CONFIDENTIAL

4.63.4   Multilevel Switch Start Level Change Command

The Multilevel Switch Start Level Change Command version 1 can be used to inform a multilevel switch, that it SHOULD start changing the level. The speed that the switch increases or decreases the level with is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Re-ser-ved	Up/Down	Ignore Start Level	Reserved				
Start Level							

Up/Down (1 bit)

If the Up/Down   bit is set to 0 the switch SHALL increase its level. If field is set to 1 the switch SHALL decrease its level.

Ignore Start Level (1 bit)

An Ignore Start Level bit set to 0 indicates to use the start level specified in the Command. An Ignore Start Level bit set to 1 indicates to start from the actual level in the device.

A device that supports Multilevel Switch Commands is not REQUIRED to implement the Ignore Start Level bit. If not implemented, the behavior SHALL be as if the Ignore Start Level bit is 1.

A controlling device is not REQUIRED to implement the Ignore Start Level bit. If not implemented, the Ignore Start Level bit SHALL always be set to 1 by a device sending this Command.

<u>Note:</u>
While many types of devices that can be supported by Multilevel Switch Commands are capable of jumping immediately to a specified start level, some types of devices cannot change their level immediately (e.g. High Intensity Discharge lamps, motor controlled devices). For this reason, it is permitted for devices not to implement the start level as specified above. For dimmers it is RECOMMENDED to implement both values of the Ignore Start Level bit.

Reserved (1+5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Start Level (8 bits)

The start level field contains the initial level for the device to assume when starting to change the level.

CONFIDENTIAL

4.63.5   Multilevel Switch Stop Level Change Command

The Multilevel Switch Stop Level Change Command version 1 can be used to inform a multilevel switch, that it SHOULD stop changing the level.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_STOP_LEVEL_CHANGE							

CONFIDENTIAL

4.64   Multilevel Switch Command Class, version 2

This section contains Commands to control a multilevel switch. These Commands allow applications to turn a multilevel switch on/off, start/stop dimming/operation at a given rate, jumping/dimming to a specified level for a given duration, and read the current level. Version 2 is extended with respect to a parameter specifying the duration/rate of the wanted operation for the Multilevel Switch Set and Multilevel Switch Start Level Change Commands.

4.64.1   Multilevel Switch Set Command

The Multilevel Switch Set Command version 2 can be used to set the level in a device that supports the multilevel switch functionality. The speed that the switch increases or decreases the level with is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SET							
Value							
Dimming Duration							

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). Furthermore, it can take values from 1 to 99 (0x01 – 0x63). The unit of the value is implementation specific i.e. percentage. All values MUST be implemented, but the actual implementation of the values 0x01…0x63 is not defined, except that for a mapping to percentages the value 0x63 represents 100%. The mapping of the value received SHOULD be monotonous i.e. a higher value in a set Command will always result in either a higher or the same setting. If a device implements less than distinct 100 values in its internal setting, then the mapping of the active internal setting SHOULD be spread equally over the entire range of 0x00…0x63.

The value 0xFF (on/enable) will cause the device to jump to the level when the device last was turned on.

All other values are ignored by the receiving device. These values are reserved for future use.

Controlling devices MAY send any of the values 0x00…0x63 and 0xFF to the device. Controlling devices MUST NOT send any of the reserved values 0x64…0xFE to the device.

CONFIDENTIAL

Dimming Duration (8 bits)

The dimming duration can either be instantly or it can be a dimming duration, or a factory default dimming duration. The dimming duration parameter is the only addition to version 1 to create a version 2 Command. The table below shows how to obtain the wanted functionality:

Dimming Duration	Description
0x00	Instantly
0x01-0x7F	Obtain dimming durations from 1 second (0x01) to 127 seconds (0x7F) in 1-second resolution. The dimming duration is defined as the interval between start of dimming and until the specified level is reached.
0x80-0xFE	Specify dimming durations from 1 minute (0x80) to 127 minutes (0xFE) in 1-minute resolution. The dimming duration is defined as the interval between start of dimming and until the specified level is reached.
0xFF	Factory default dimming duration or a rate compatible with a version 1 implementation of the Multilevel Command Class.

It is allowed to make a device supporting version 2 of the Multilevel Switch Command Class which at the same time is backward compatible to a device supporting version 1 of the Multilevel Switch Command Class with respect to the dimming profile. This can be done by implementing a version 1 dimming profile when specifying a dimming duration equal to 0xFF for a device supporting version 2 of the Multilevel Switch Command Class. Remember to default initialize the dimming duration to 0xFF for a device supporting version 2 of the Multilevel Switch Command Class to be backward compatible in case it is controlled by a Multilevel Switch Command Class version 1.

4.64.2 Multilevel Switch Get Command

The Multilevel Switch Get Command version 2 is similar to version 1.

4.64.3 Multilevel Switch Report Command

The Multilevel Switch Report Command version 2 is similar to version 1.

CONFIDENTIAL

4.64.4 Multilevel Switch Start Level Change Command

The Multilevel Switch Start Level Change Command version 2 can be used to inform a multilevel switch, that it SHOULD start changing the level. The speed that the switch increases or decreases the level with is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Re-ser-ved	Up/Down	Ignore Start Level	Reserved				
Start Level							
Dimming Duration							

Up/Down (1 bit)

If the Up/Down bit is set to 0 the switch SHOULD increase the level. If field is set to 1 the switch SHOULD decrease the level.

Ignore Start Level (1 bit)

If the Ignore Start Level bit is set to 0 the switch SHOULD use the start level specified in the Command. If field is set to 1 the switch SHOULD start from the actual level in the device.

Reserved (1 + 5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Start Level (8 bits)

The start level field contains the initial level that the switch SHOULD assume when it start to change the level. Most devices are capable of jumping immediately to the specified start level. But some devices the Multilevel Switch Command Class also was intended for can not change level immediately such as HID (High Intensity Discharge) lamps, motor controlled devices etc. Therefore are Z-Wave enabled devices that are not capable to change level immediately allowed to use the current level as a starting point even though a start level is specified in the Command.

Dimming Duration (8 bits)

Refer to description under the Multilevel Switch Set Command version 1. The only deviation is how the dimming duration is defined. The Multilevel Switch Start Level Change only starts dimming but does not define when to stop. The Multilevel Switch Stop Level Change Command is REQUIRED to stop dimming. This requires that the dimming duration MUST be defined according to a fixed reference. The dimming duration is therefore in this case defined as the interval it takes to dim from level 0 to 99. The Multilevel Switch Start Level Change can now determine the dimming rate to use. The dimming duration parameter is the only addition to version 1 to create a version 2 Command.

4.64.5 Multilevel Switch Stop Level Change Command

The Multilevel Switch Stop Level Change Command version 2 is similar to version 1.

CONFIDENTIAL

4.65   Multilevel Switch Command Class, version 3

Version 3 of the Multilevel Switch Command Class version 3implements the option to retrieve a switch type from the device for the controller to determine the behavior in terms of movement. In addition a general increment and decrement mechanism has been added to the Start Level Change command.

Commands not described in Version 3 stays unchanged from Version 2.

4.65.1   Multilevel Switch Supported Get Command

The Multilevel Switch Supported Get Command used to interview the device for the Switch Type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SUPPORTED_GET							

4.65.2   Multilevel Switch Supported Report Command

The Multilevel Switch Supported Report Command is sent as a response to the Multilevel Switch Supported Get command. This report command contains information about the behavior of the device with regards to the Up/Down and Inc/Dec parameters included in the Multilevel Switch Start Level Change command. This report MUST NOT be sent unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_SUPPORTED_REPORT							
Reserved			Primary Switch Type				
Reserved			Secondary Switch Type				

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Primary / Secondary Switch Type (5 bits)

Note! The Primary Switch Type is addressed by Multilevel Switch Set, Get and Report as well as Start/Stop Level Change (Up/Down) commands. In case the Primary Switch Type is set to 0x00 (Undefined / Not supported) the application MUST respond to Multilevel Switch Get with Multilevel Switch Report (Value = 0xFE) for unknown state/position.

Note! The Secondary Switch Type is addressed only through Multilevel Switch Start/Stop Level Change (Inc/Dec).

For both Switch Types the value describes the movement profile of the device. Refer to the table below with respect to defined Switch Types. New types/values can be requested from Sigma Designs.

Switch Type Value	0x00 (Direction/Endpoint A)	0x63/0xFF (Direction/Endpoint B)
0x00	Undefined / Not supported	
0x01	Off	On
0x02	Down	Up
0x03	Close	Open
0x04	Counter-Clockwise	Clockwise
0x05	Left	Right
0x06	Reverse	Forward
0x07	Pull	Push
0x08-0x1F	Reserved	

4.65.3   Multilevel Switch Start Level Change Command

The Multilevel Switch Start Level Change Command can be used to inform a multilevel switch, that it SHOULD start changing the level. The speed that the switch increases or decreases the level with is implementation specific. In Multilevel Switch Command Class (Version 3) a Step Size field has been added to this command to support general increment or decrement function typically available in a motor controlled device featuring two-dimensional movements.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_MULTILEVEL							
Command = SWITCH_MULTILEVEL_START_LEVEL_CHANGE							
Up/Down		Ignore Start Level	Inc/Dec		Reserved		
Start Level							
Dimming Duration							
Step Size							

Up/Down (2 bits)

If the Up/Down field is set to 0 the switch SHOULD increase the level. If field is set to 1 the switch SHOULD decrease the level. If the field is set to 3 there is no change to the Up/Down level.

Value	Description
0x00	Up
0x01	Down
0x02	Reserved
0x03	No Up/Down motion

Ignore Start Level (1 bit)

This field is unchanged from version 2.

Note! Refer to the device class specification for correct implementation of this field.

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

Start Level (8 bits)

This field is unchanged from version 2.

Note! Refer to the device class specification for correct implementation of this field.

Dimming Duration (8 bits)

This field is unchanged from version 2.

Note! Refer to the device class specification for correct implementation of this field.

Inc/Dec (2 bits)

This field instructs the device to Increment or Decrement the value specified in the Step Size field.

Value	Description
0x00	Increment
0x01	Decrement
0x02	Reserved
0x03	No Inc/Dec

Step Size (8 bits)

The "Step Size" field indicates the percentage of steps an increment or decrement function SHOULD execute. "Step Size" can take values from 0 to 99 (0x00 – 0x63) and 255 (0xFF). All values MUST be implemented, but the actual implementation of the values 0x00…0x63 is a mapping to percentages where the value 0x63 represents 100%. The mapping of the value received SHOULD be monotonous i.e. a higher value in a start level change command will always result in either a higher or the same setting. If a device implements less than distinct 100 values in its internal setting, then the mapping of the active internal setting SHOULD be spread equally over the entire range of 0x00…0x63. The value 255 (0xFF) indicates a fixed step size defined by the OEM.

If the Increment/Decrement field is set to 3, the Step Size field MUST be set to 0.

4.66   Multilevel Toggle Switch Command Class, version 1

Do not use this command class for new devices.

Use instead Multilevel Switch Generic Device Class for such devices.

This section contains Commands that can be used to make a multilevel toggle switch. These Commands allow applications to set and get the level of a multilevel toggle switch.

4.66.1   Multilevel Toggle Switch Set Command

The Multilevel Toggle Switch Set Command can be used to set the level in a device that supports the multilevel switch functionality.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL								
Command = SWITCH_TOGGLE_MULTILEVEL_SET								

4.66.2   Multilevel Toggle Switch Get Command

The Multilevel Toggle Switch Get Command can be used to request the state of the load controlled by the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL								
Command = SWITCH_TOGGLE_MULTILEVEL_GET								

4.66.3 Multilevel Toggle Switch Report Command

The Multilevel Toggle Switch Report Command can be send unsolicited or requested by the Multilevel Toggle Switch Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL							
Command = SWITCH_TOGGLE_MULTILEVEL_REPORT							
Value							

Value (8 bits)

The value can be either 0x00 (off/disable) or 0xFF (on/enable). Furthermore it can take values from 1 to 99 (0x01 – 0x63).

CONFIDENTIAL

4.66.4 Multilevel Toggle Switch Start Level Change Command

The Multilevel Toggle Switch Start Level Change Command can be used to inform a multilevel toggle switch, that it SHOULD start changing the level. The speed that the switch increases or decreases the level with is implementation specific.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL							
Command = SWITCH_TOGGLE_MULTILEVEL_START_LEVEL_CHANGE							
Roll Over	Reser-ved	Ignore Start Level	Reserved				
Start Level							

Roll Over (1 bit)

If the roll over bit is set to 0 then the switch SHOULD stop when reaching the max or min level. If the roll over bit is set to 1 then the switch SHOULD continually increase and decrease the level until otherwise instructed.

Ignore Start Level (1 bit)

If the Ignore Start Level bit is set to 0 the switch SHOULD use the start level specified in the Command. If field is set to 1 the switch SHOULD start from the actual level in the device.

Reserved (1 + 5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Start Level (8 bits)

The start level field contains the initial level that the switch SHOULD assume when it start to change the level.

4.66.5 Multilevel Toggle Switch Stop Level Change Command

The Multilevel Toggle Switch Stop Level Change Command can be used to inform a multilevel toggle switch, that it SHOULD stop changing the level.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SWITCH_TOGGLE_MULTILEVEL							
Command = SWITCH_TOGGLE_MULTILEVEL_STOP_LEVEL_CHANGE							

CONFIDENTIAL

4.67 Z-Wave Network Management Command Classes

The Z-Wave Network Management commands are organized as follows.

All Network Management Command Class MUST be sent securely when used on the Z-Wave PAN, using at least Z-Wave Security Command Class Version 1. When used on the LAN side other means of security SHOULD be used.

The commands defined in the following sections MAY span more than the payload bytes available in classic Z-Wave frames, and MUST be fragmented using the Transport Service Command Class. When using the Network Management Command Classes, there is no maximum frame size. Instead all frames exceeding the maximum size MUST be split using Transport Service. This also applies to the Security Command Class. This means that the built-in Security Command Class fragmentation is not used, and the encrypted payload size MAY increase beyond the normal Z-Wave frame size.

When using IP transport, the IP UDP limit of 1280 bytes MUST be respected.

There is a risk that a remote host tries to use commands in a controller which has become secondary in the meantime. If a requesting node tries to use a command which is not available in the current configuration the node MUST return a Not Supported Command as defined per the Application Capability Command Class.

All Network management commands MUST be sent as singlecast. A receiving node MUST ignore Network Management commands received via broadcast or multicast.

The following valid combinations of the Primary, Basic and Inclusion Network Management command classes exist:

SIS/Primary/Inclusion Controller:

Network Management Inclusion

Network Management Basic

Network Management Primary

**Slave / Secondary Controller:**

Network Management Basic

CONFIDENTIAL

Command Class	Command	Purpose	Comment
Network Management Proxy	Node List Get	Request node list from Network Management Proxy.	
	Node List Report	Return node list from Network Management Proxy.	If proxy is not a primary controller the node list MAY not be up to date.
	Node Info Cached Get	Request cached node info from Network Management Proxy.	Information is also available for sleeping battery nodes.
	Node Info Cached Report	Return cached node info from Network Management Proxy.	A full Node Information structure is returned
Network Management Basic	Learn Mode Set	Enable learn mode. (requires a live interface)	
	Learn Mode Set Status	Conveys status information during the learn mode process.	
	Node Information Send	Used to make a node identify itself to other nodes	MAY be used for support of push-button association in legacy controller devices.
	Request Network Update	Request updated topology from SIS	
	Request Network Update Status	Conveys status information for the Network Update process	
	Default Set	Reset the controller to the factory default state.	
	Default Set Complete	Reports fail/success of the reset operation	Note that when used on the PAN side a Default Set Complete will not be received
Network Management Inclusion	Node Add	Add nodes to the Z-Wave network.	
	Node Add Status	Conveys status information during the inclusion process.	
	Node Remove	Remove a node from the network	
	Node Remove Status	Conveys status information for the removal process.	
	Failed Node ID Remove	Remove a node ID from the node list	MUST be ignored if the node is not failing
	Failed Node ID Remove Status	Conveys status information for the removal process	

CONFIDENTIAL

	Failed Node ID Replace	Re-use a node ID of a failing node for a new node	MUST be ignored if the node is not failing
	Failed Node ID Replace Status	Conveys status information for the replacement process	
	Node Neighbor Update Request	Get Neighbors from the specified node.	Used to update topology map if a node is moved.
	Node Neighbor Update Status	Report status of the Node Neighbor Update Request call.	
	Return Route Assign	Assign static return routes to a Routing Slave node or Enhanced Slave node.	
	Return Route Assign Complete	Reports fail/success of the Return Route Assign operation	
	Return Route Delete	Delete all static return routes in a Routing Slave node or Enhanced Slave node.	
	Return Route Delete Complete	Reports fail/success of the Return Route Delete operation	
Network Management Primary	Controller Change	Add a controller node to the Z-Wave network and make it primary.	
	Controller Change Status	Conveys status information during the controller change process.	

CONFIDENTIAL

4.67.1   Scope of Network Management

Network management commands MAY be used in a number of scenarios. Three scopes have been identified:
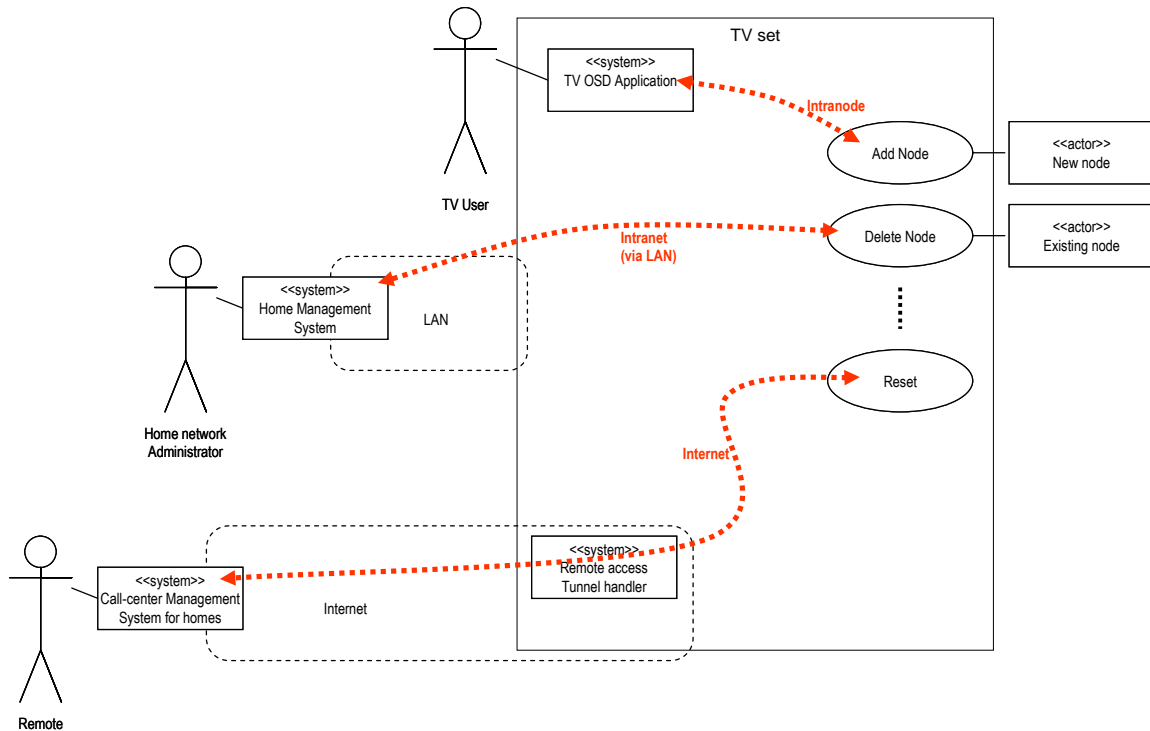
4.67.1.1   Intranode

When used in an intranode configuration, the network management command classes are primarily used for implementation convenience. As an example, a software module of the Z/IP Gateway application MAY be used to provide a standard IP-based interface for other Linux applications inside a set-top box. In this way an application programmer does not have to bother about serial port communication, Telnet command parsing, etc.

4.67.1.2   Intranet (LAN)

Managed building automation systems MAY implement one central network manager controlling a number of geographically distributed Z/IP Gateways via the network management command classes. Each Z/IP Gateway MAY be instructed to perform local inclusion or exclusion of nodes; thus creating a large infrastructure segmented into subnets.

4.67.1.3   Internet (WAN)

The help desk of a service provider MAY provide support from a remote call center via the Internet. This enables the deployment of border routers, remote controls and plug-in modules in consumer environments without relying on the technical interest and/or capabilities of the user.

4.67.2   Security considerations

Network management is a powerful toolbox. From an application level it MUST be ensured that the user does not unintentionally reset the controller or remove nodes.

CONFIDENTIAL

At the same time it MUST be ensured that it is not possible for unauthorized persons to inject malicious commands into the network, e.g. resetting the primary controller to default factory settings.
Therefore classic Z-Wave networks MUST apply S0 network security, as specified by Security Command Class version 1 or better protection of the network management commands defined in this document.
The Transport Service Command Class MUST be implemented as this is REQUIRED for secure communication using long packets.
If the network management commands are carried in IP packets over Z-Wave, a minimum level of security is automatically applied since S0 network security is mandatory for all Z/IP traffic.

When Z-Wave network management commands are carried over IP LAN and WAN media (intranet & internet) the IP traffic SHOULD be protected. A Z/IP Gateway MAY allow a LAN-based IP host to send un-encrypted Network Management commands to the primary controller via the Z/IP Gateway. Support for un-encrypted Network Management commands SHOULD be disabled by default and after a factory reset.

4.67.3   Designing for single-threading and limited transmit buffer

In order to support constrained CPU platforms, the Z-Wave API has been designed for single-threaded operation. A node MUST NOT accept another Network Management command if already processing one such command.

A node MUST return status messages to the node that actually initiated the operation. Another Network Management message arriving during a Network Management operation MUST be ignored by the application layer.

An originating node MUST be robust by design so that it times out waiting for a status message. If not receiving a status message within at least 10 seconds the node SHOULD re-send the Network Management command; using the same sequence number to allow the target node to detect duplicates.

A target node MAY return a "busy" indication. Doing so could however lead to transmit buffer overflows. Care SHOULD be taken to avoid this during implementation.

In general, these command classes rely on basic Z-Wave Acknowledgements; preventing the need for "started" type messages. The Z-Wave Ack does not necessarily indicate that the command is being executed, but that it has been received by the protocol. The sending application MUST wait for the Network Management command callback, or time out.

Upon completion, the target application MAY return status messages signaling "Done" or "Failed". To avoid transmit queue overflows, messages such as LearnModeStatus SHOULD be emitted before and after calling the API function LearnModeSet but never during the execution of the API function.

CONFIDENTIAL

4.67.4   Network Management Proxy Command Class, version 1

The Network Management Proxy Command Class provides functions to access basic network information such as the list of nodes currently included.

4.67.4.1   Node List Get Command

The Node List Get Command is used to request the node list from local storage in a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_PROXY							
Command = COMMAND_NODE_LIST_GET							
Seq. No							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

4.67.4.2   Node List Report Command

The Node List Report Command carries node data for the node range requested with the command Node List Get.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_PROXY							
Command = COMMAND_NODE_LIST_REPORT							
Seq. No							
Status							
Node List Controller ID							
Node List Data 1							
…							
Node List Data 29							

CONFIDENTIAL

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Values of the status byte:

0: Returned latest updated node list
1: Cannot guarantee that returned node list is latest update

Node List Controller ID (1 byte)

The Node List Controller ID is a Node ID pointing at a controller, which keeps latest updated node list. A value of 0 (zero) indicates that Node List Controller ID is unknown.

The Node List Controller SHOULD provide up-to-date information, but the actual freshness of data depends on the network construction. If a portable controller is primary there MAY be no access to the most recent network data. In that case it MAY be REQUIRED that a user manually wakes up the portable controller and initiates a controller replication to an always listening secondary controller.

Note: No explicit Z-Wave route is provided for reaching the Node List Controller. The requesting node MAY use methods such as explorer discovery or Controller Network Update if the node does not already hold a working route to the indicated Node List Controller.

Notice: Node List Controller ID MAY not support Network Management Proxy Command Class.

Node List Data (29 bytes)

This field carries a complete bitmap presenting all included nodes as a set bit ('1') while unused node IDs are presented as a ('0'). The first bit in the bitmap represents node ID 1; the last bit represents node ID 232.

A receiving node MAY use the Node Info Cached Get command to get information on individual node properties.

CONFIDENTIAL

4.67.4.3   Node Info Cached Get Command

The Node Info Cached Get Command is used to request node capabilities for a node cached by another node. The command works as a proxy function provided by the node list controller. The purpose is to preserve the bandwidth of the Z-Wave network and to provide access to properties of sleeping nodes.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_PROXY							
Command = COMMAND_NODE_INFO_CACHED_GET							
Seq. No							
Reserved				Max Age			
NodeID							

A Z/IP client MAY issue the Node Info Cached Get command as an IPv4 broadcast or an IPv6 'all routers' multicast packet. A Z/IP Gateway MUST accept such a packet and return a Node Info Cached Report in response.

A Node Info Cached Report returned by a Z/IP Gateway in response to a multicast message MUST be delayed by a random delay in the range 0..450msec as more than one Z/IP Gateway may be responding. The Z/IP Gateway MUST send the response as a unicast IP packet.
A Z/IP Client MAY time out waiting for Node Info Cached Report commands after 500msec.

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Max Age(4 bits)

The maximum age of the NodeInfo frame, given in 2^n minutes. If the cache entry does not exist or if it is older that the value given in this field, the Z/IP Gateway SHOULD attempt to get a Fresh NodeInfo frame before responding to the Node Info Cached Get command. A value of 15 means infinite, i.e. No Cache Refresh. A value of 0 means force update.

The values 0..15 allow for cache timeouts in the range 1min, 2min, 4min, …, 11days – and infinite.

NodeID (1 byte)

The NodeID of the node for which the destination node is to return cached data. A value of 0 is interpreted as the ID of the queried network management node.

CONFIDENTIAL

4.67.4.4   Node Info Cached Report Command

Node Info Cached Report Command for returning cached node information.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_PROXY							
Command = COMMAND_NODE_INFO_CACHED_REPORT							
Seq. No							
Status				Age			
List.	Z-Wave Protocol Specific Part						
Opt. Func.	Sensor			Z-Wave Protocol Specific Part			
Reserved							
Basic Device Class							
Generic Device Class							
Specific Device Class							
Non-Secure Command Class 1 *)							
...							
Non-Secure Command Class n *)							
Security Scheme 0 MARK 0xF1							
Security Scheme 0 MARK 0x00							
Security Scheme 0 Command Class 1 *)							
…							
Security Scheme 0 Command Class n *)							

*) Command classes MAY be extended $\Rightarrow$ spanning two bytes for one command class

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

CONFIDENTIAL

Status (4 bits)

Status indicating the progress.

Status	Comment
STATUS_OK	The requested NodeID could be found and up-to-date information is returned.
STATUS_NOT_RESPONDING	The requested NodeID could be found but fresh information could not be retrieved.
STATUS_UNKNOWN	The NodeID is unknown.

Table 8, Encoding of Node Info Cached Report::Status parameter

Age (4 bits)

Age of the NodeInfo frame. Time is given in 2^n minutes.

List. (1 bit)

The listening bit is set to 1 if this node is always listening for commands and 0 if the node does not listen for commands.

Opt. Func. (1 bit)

The Optional Functionality bit indicates if true (== '1') the node supports more command classes in addition to the ones covered by the device classes listed in this message. The additional command classes follow the device class fields.

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Basic Device Class (1 byte)

Generic Device Class (1 byte)

Specific Device Class (1 byte)

CONFIDENTIAL

Command Class (n bytes)

Variable-length part of the frame carrying one or more command classes implemented by the actual node. Any command classes listed before a mark is Supported Non-Secure. When a 0xEF is encounter the following command classes are Controlled Non-Secure. If 0xF1->0xFF is encounter they are extended command classes and the following byte MUST be included. 0xFFF1 is a Security Scheme 0 Command Class Mark that indicates that all following command classes are supported using Scheme 0 Security, after a Security Scheme 0 Mark a Command Class Mark and additionally be used to indicate Control for Scheme 0 Security Command Classes.

Command Class ID	Comment
0xF1->0xFF	Extended Command Classes, represented by two bytes – ID is made up of MS byte being 0xF1->0xFF and LS byte 0x00->0xFF
0xEF	Command Class Mark – Anything that follows this mark and until next mark is Controlled and not supported
0xF100	Security Scheme 0 Command Class Mark – Anything that follows this mark and until next mark is supported using Security Scheme 0

CONFIDENTIAL

4.67.5   Network Management Basic Node Command Class, version 1

The Network Management Basic Node Command Class provides functions to get nodes included into a Z-Wave network, enabling nodes to request network updates and resetting itself factory default state.

4.67.5.1   Default Set Command

The Default Set Command sets the Controller back to the factory default state.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_DEFAULT_SET							
Seq. No							

Warning: This function SHOULD be used with care as it could render a network unusable if the primary controller in an existing network is set back to default. If a node is set to default while it is still a member of a network, the node will become a failing nodeID in that network.

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

4.67.5.2   Default Set Complete Command

The Default Set Complete Command Indicates that the Default Set Command execution has been completed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_DEFAULT_SET_COMPLETE							
Seq. No							
Status							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Possible values are DEFAULT_SET_DONE or DEFAULT_SET_BUSY. A controller MUST return DEFAULT_SET_BUSY if it is already engaged in another network management command.

CONFIDENTIAL

4.67.5.3   Learn Mode Set Command

The Learn Mode Set Command is used to allow a node to be added to (or removed from) the network.
When a node is added to the network the node is assigned a valid Home ID and Node ID.
This command allows a controlling application to request the transmission of Node Information frames in
regular intervals until included or removed or until learn mode is disabled again.

NOTE: Learn mode SHOULD only be enabled when necessary, and it SHOULD always be disabled
again as quickly as possible. However to ensure a successful synchronization of the inclusion process
the device SHOULD be able to stay in learn mode in up to 5 seconds.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_LEARN_MODE_SET							
Seq. No							
Reserved							
Mode							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new
message MUST be an increment of the previous message. When receiving response to this request the
node MUST verify that the response carries the same sequence number.

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no
assumptions on the values of these fields nor perform processing based on their content.

Mode (1 byte)

The Mode field controls operation.

Modes	Comment
LEARN_MODE_SET_CLASSIC	Start the learn mode on the controller and accept only being included in direct range
LEARN_MODE_SET_NWI	Start the learn mode on the controller and accept routed inclusion.
LEARN_MODE_SET_DISABLE	Stop the learn mode of the node

Table 9, Encoding of Slave Learn Mode Set ::Mode parameter

4.67.5.3.1   Learn mode in a controller

If the node is a controller type the node receives and stores the node list and routing table for the
network. The controller application receives and stores application information transmitted as part of the
replication.

Note: This function will most likely change the capabilities of the controller.

CONFIDENTIAL

4.67.5.4   Learn Mode Set Status Command

The Learn Mode Set Status Command Used to indicate the progress of the Learn Mode Set command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_LEARN_MODE_SET_STATUS							
Seq. No							
Status							
Reserved							
New Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Status indicating the progress.

Modes	Comment
LEARN_MODE_DONE	The learn process is complete and the controller is now included into the network.
LEARN_MODE_FAILED_TIMEOUT	The learn process failed due to time-out waiting for controller
LEARN_MODE_SECURITY_FAILED	The learn process is complete but security handshaking failed. The node is **not** operating securely.
LEARN_MODE_FAILED	The learn process failed in some general way

Table 10, Encoding of Learn Mode Status::Status parameter

The learn mode status message returns a LEARN_MODE_DONE if inclusion was completely successful. If a node supports security command classes, LEARN_MODE_DONE indicates that the inclusion was completed; including the security handshake. If inclusion is successful but security handshake fails, the status LEARN_MODE_SECURITY_FAILED MUST be returned.

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

New Node ID (1 byte)

The node ID assigned to the new node by another primary controller or inclusion controller.
If the node was removed from the network, the new node ID MUST be zero.

CONFIDENTIAL

4.67.5.5   Node Information Send Command

When receiving this Node Information Send Command , a node MUST send out a Node Information Frame.

No status message is returned for this command.

A management application MAY use this message to make a node identify itself towards a classic Z-Wave remote control during association operations. This message SHOULD NOT be used while learn mode is activated. Instead, periodic Node Information transmissions MAY be enabled along with learn mode; refer to 4.67.5.1.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_NODE_INFORMATION_SEND							
Seq. No							
Reserved							
Destination Node ID							
tx Options							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Destination Node ID (1 byte)

The node ID of the node that is to receive this Node Information frame. The node ID MAY be set to the value NODE_BROADCAST to reach all nodes within direct range. Acknowledgement SHOULD NOT be requested when broadcasting.

tx Options (1 byte)

The tx Options field allows a management application to specify if the Node Information frame is to be sent with special properties. Several flags MAY be combined via a logical OR.

CONFIDENTIAL

Option flags	Comment
NULL	Transmit at normal power level without any transmit options.
TRANSMIT_OPTION_ACK	Request acknowledgment from destination node. Allow routing.
TRANSMIT_OPTION_NO_ROUTE	Send only in direct range
TRANSMIT_OPTION_EXPLORE	Resolve new routes via explorer discovery if existing routes fail
TRANSMIT_OPTION_LOW_POWER	Transmit at low output power level (1/3 of normal RF range)

Table 11, Encoding of Node Information Send::Tx Options

The typical tx option flags used for Node Information Send will be TRANSMIT_OPTION_NO_ROUTE and the nodeID will be the broadcast NodeID.

4.67.5.6   Network Update Request Command

The Network Update Request Command is used to request network topology updates from the SUC/SIS node. All controllers can use this call in case a SUC ID Server (SIS) is available.
Secondary controllers can only use this call when a SUC is present in the network. Routing Slaves can only use this call, when a SUC is present in the network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_NETWORK_UPDATE_REQUEST							
Seq. No							

NOTE: The SUC can only handle one network update at a time, so care SHOULD be taken not to have multiple controllers in the network ask for updates at the same time.

WARNING: This API call will generate a lot of network activity that will use bandwidth and stress the SUC in the network. Therefore, network updates SHOULD be requested as seldom as possible.

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

CONFIDENTIAL

4.67.5.7   Network Update Request Status Command

The Network Update Request Status Command indicates that the Network Update Request command execution has completed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_BASIC							
Command = COMMAND_NETWORK_UPDATE_REQUEST_STATUS							
Seq. No							
Status							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Indicates the status of the Network Update process.

Status values	Comment
SUC_UPDATE_DONE	The update process succeeded
SUC_UPDATE_ABORT	The update process aborted because of an error
SUC_UPDATE_WAIT	The SUC node is busy
SUC_UPDATE_DISABLED	The SUC functionality is disabled
SUC_UPDATE_OVERFLOW	The controller requested an update after more than 64 changes have occurred in the network. The controller has to make a replication.

Table 12, Encoding of Network Update Request Status :: Status parameter

CONFIDENTIAL

4.67.6   Network Management Inclusion Command Class, version 1

The Network Management Inclusion Command Class provides functionality only available in a primary controller. Since this is a dynamic property, there is a risk that a remote host tries to use commands in a controller which has become secondary in the meantime.

If a requesting node tries to use a command which is not supported in the current configuration the node MUST return a "Command Class Not Supported" as defined per the Application Capability Command class.

4.67.6.1   Node Add Command

The Node Add Command is used to add nodes to the network.

The process of adding a node is started by the network management application sending a Node Add command to the Node List Controller (primary controller). The network management application receives a status message indicating if inclusion was successful. If NWI inclusion was used, the calling application MAY re-issue the Node Add command if more nodes are to be included.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_NODE_ADD							
Seq. No							
Reserved							
Mode							
tx Options							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

NOTE: The Node Add state SHOULD be disabled after use to avoid adding other nodes than expected. It is RECOMMENDED to have a timer that disables the Node Add state after a while without any activity.

CONFIDENTIAL

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Mode (1 byte)

Modes	Comment
NODE_ADD_ANY	Add any type of node to the network.
NODE_ADD_STOP	Stop adding nodes to the network. A call with this mode will always result in a status message reporting NODE_ADD_STATUS_FAILED

Table 13, Encoding of Node Add :: Mode parameter

tx Options (1 byte)

The tx Options field allows a controlling node to specify if transmissions MUST use special properties. Several flags MAY be combined via a logical OR.

Option flags	Comment
NULL	Transmit at normal power level without any transmit options.
TRANSMIT_OPTION_EXPLORE	Allow network-wide inclusion
TRANSMIT_OPTION_LOW_POWER	Transmit at low output power level (1/3 of normal RF range)

Table 14, Encoding of Node Add :: Tx Options

The RECOMMENDED use of option flags in combination with the NODE_ADD_ANY option is to set the flag TRANSMIT_OPTION_EXPLORE.

Installer scenarios with a requirement for more confidential transfer of network security keys MAY set the flag TRANSMIT_OPTION_LOW_POWER. This requires that a new node is included in direct range of the primary controller.

CONFIDENTIAL

4.67.6.2 Node Add Status Command

The Node Add Status Command is used to report the result of the Node Add command.

To avoid re-entrance issues and transmit queue overflows, the Node Add Status message SHOULD be issued before or after processing the Node Add message but never during the execution of the function.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_NODE_ADD_STATUS							
Seq. No							
Status							
Reserved							
NewNodeID							
NodeInfoLength							
List.	Capability						
Opt. Func.	Security						
Basic Device Class							
Generic Device Class							
Specific Device Class							
Command Class 1 *)							
...							
Command Class n *)							

*) Command classes MAY be extended ⇒ spanning two bytes for one command class

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

CONFIDENTIAL

Status (1 byte)

Indicates the status of the Node Add process.

Status values	Comment
NODE_ADD_STATUS_DONE	The new node has now been included.
NODE_ADD_STATUS_FAILED	The process failed
NODE_ADD_STATUS_SECURITY_FAILED	Node has been included but the secure inclusion failed.

Table 15, Encoding of Node Add Status :: Status parameter

The Node Add status message returns a NODE_ADD_STATUS_DONE if inclusion was completely successful.
If a node supports security command classes, NODE_ADD_STATUS_DONE indicates that the inclusion was completed; including the security handshake. If inclusion is successful but security handshake fails, the status NODE_ADD_STATUS_SECURITY_FAILED MUST be returned.

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

New Node ID (1 byte)

The nodeID of the newly added node. Only valid if Status is NODE_ADD_STATUS_DONE; else this field MUST be zero.

Node Info Length (1 byte)

The length of the encapsulated Node Information. The length value includes the length field.

Node Info (n bytes)

The node info structure of the newly added node. The length is signaled in Node Info Length.

Basic Device Class (1 byte)

Generic Device Class (1 byte)

Specific Device Class (1 byte)

Command Class (n bytes)

Variable-length part of the frame carrying one or more command classes implemented by the actual node. Any command classes listed before a mark is Supported Non-Secure. When a 0xEF is encounter the following command classes are Controlled Non-Secure. If 0xF1->0xFF is encounter they are extended command classes and the following byte MUST be included. 0xFFF1 is a Security Scheme 0 Command Class Mark that indicates that all following command classes are supported using Scheme 0 Security, after a Security Scheme 0 Mark a Command Class Mark and additionally be used to indicate Control for Scheme 0 Security Command Classes.

Command Class ID	Comment
0xF1->0xFF	Extended Command Classes, represented by two bytes – ID is made up of MS byte being 0xF1->0xFF and LS byte 0x00->0xFF
0xEF	Command Class Mark – Anything that follows this mark and until next mark is Controlled and not supported
0xF100	Security Scheme 0 Command Class Mark – Anything that follows this mark and until next mark is supported using Security Scheme 0

4.67.6.3   Node Remove Command

The Node Remove Command is used to remove any node from the network. The operation MAY only be used in direct range between the controller and the node that is to be deleted.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_NODE_REMOVE							
Seq. No							
Reserved							
Mode							

There is no node ID in this message. The node ID of the node SHOULD be received from the actual node since the node ID MUST be cleared in the deleted node as well as in the primary controller.

NOTE: The Node Remove state SHOULD ALWAYS be disabled after use to avoid adding other nodes than expected. It is RECOMMENDED that Node Remove is sent with NODE_REMOVE_STOP every time a NODE_REMOVE_STATUS_DONE is received, and that the controller also contains a timer that disables the Node Remove state.

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

CONFIDENTIAL

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Mode (1 byte)

Modes	Comment
NODE_REMOVE_ANY	Remove any type of node from the network
NODE_REMOVE_STOP	Stop the delete process

<p align="center">Table 16, Encoding of Node Remove :: Mode parameter</p>

The process of removing a node is started by sending Node Remove. The delete process is complete when the status NODE_REMOVE_STATUS_DONE is returned.

4.67.6.4 Node Remove Status Command

The Node Remove Status Command is used to report progress during the removal of nodes.

To avoid re-entrance issues and tx queue overflows, the Node RemoveStatus message SHOULD be emitted before and after calling the API function RemoveNodeFromNetwork but never during the execution of the API function.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_NODE_REMOVE_STATUS							
Seq. No							
Status							
NodeID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Indicates the status of the node remove process.

Status values	Comment
NODE_REMOVE_STATUS_DONE	The node has now been removed and the controller is ready to continue normal operation again. Removed nodeID is returned.
NODE_REMOVE_STATUS_FAILED	The remove process failed

Table 17, Encoding of Status parameter of Node Remove Status

Node ID (1 byte)

MUST be the node ID that was attempted to be removed.

4.67.6.5   Failed Node Remove Command

The Failed Node Remove Command is used to remove a non-responding node. A non-responding node is put onto the failed node ID list in the controller. In case the node responds again at a later stage then it is removed from the failed node ID list. A node MUST be on the failed node ID list and as an extra precaution also fail to respond before it is removed. Responding nodes MUST NOT be removed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_FAILED_NODE_REMOVE							
Seq. No							
Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Node ID (1 byte)

Node ID to be removed.

CONFIDENTIAL

4.67.6.6   Failed Node Remove Status Command

The Failed Node Remove Status Command is used to indicate the progress of the Remove Failed Node Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_FAILED_NODE_REMOVE_STATUS							
Seq. No							
Status							
Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Status field indicating the result of the operation.

Status values	Comment
DONE	The process was completed successfully.
FAILED_NODE_NOT_FOUND	The requested process failed. The nodeID was not found in the controller list of failing nodes.
FAILED_NODE_REMOVE_FAIL	The requested process failed. Reasons include: * Controller is busy * The node responded to a NOP; thus the node is no longer failing.

Table 18, Encoding of Status parameter of Failed Node ID Remove :: Status

The removal process MAY fail if the requested NodeID responds to requests. The error message FAILED_NODE_REMOVE_FAIL does not indicate why the removal operation failed.
A network management application SHOULD issue a NOP for the requested NodeID to test if the node is actually responding again.

Node ID (1 byte)

Node ID that was to be removed.

4.67.6.7   Failed Node Replace Command

The Failed Node Replace Command is used to replace a non-responding node with a new one in having the same nodeID. A non-responding node is put onto the failed node ID list in the controller. In case the node responds again at a later stage then it is removed from the failed node ID list. A node MUST be on the failed node ID list and as an extra precaution also fail to respond before it is removed or replaced. Responding nodes cannot be removed.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_FAILED_NODE_REPLACE							
Seq. No							
Node ID							
tx Options							
Mode							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Node ID (1 byte)

Node ID to be replaced.

tx Options (1 byte)

The tx Options field allows a controlling node to specify if transmissions MUST use special properties.

Option flags	Comment
NULL	Transmit at normal power level without any transmit options.
TRANSMIT_OPTION_LOW_POWER	Transmit at low output power level (1/3 of normal RF range)

Table 19, Encoding of Failed Node Replace :: Tx Options

Mode (1 byte)

Mode field indicating type of operation.

Mode values	Comment
START_FAILED_NODE_REPLACE	Initiate a failed node replace process.
STOP_FAILED_NODE_REPLACE	Cancel a failed node replace process.

Table 20, Encoding of Failed Node Replace :: Mode

CONFIDENTIAL

4.67.6.8   Failed Node Replace Status Command

The Failed Node Replace Status Command is used to indicate the progress of the Replace Failed Node Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_FAILED_NODE_REPLACE_STATUS							
Seq. No							
Status							
Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Status field indicating the result of the operation.

Status values	Comment
DONE	The process was completed successfully.
FAILED_NODE_REPLACE_FAIL	The requested process failed. Reasons include: * Controller is busy * The node responded to a NOP; thus the node is no longer failing.
FAILED_NODE_REPLACE_SECURITY_FAILED	Replace completed successfully but security handshake failed.

Table 21, Encoding of Status parameter of Failed Node Remove ID :: Status

The replace process MAY fail if the requested NodeID responds to requests. The error message FAILED_NODE_REMOVE_FAIL does not indicate why the removal operation failed.

A network management application SHOULD issue a NOP for the requested NodeID. If a response is received the user SHOULD be notified that the node MUST be removed using the normal removal operation.

Node ID (1 byte)

Node ID that was to be replaced.

4.67.6.9   Node Neighbor Update Request Command

The Node Neighbor Update Request Command is used to instruct a node with Node ID to perform a Node Neighbor Search, to update the topology on the controller.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_NODE_NEIGHBOR_UPDATE_REQUEST							
Seq. No							
Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Node ID (1 byte)

Node ID of the node that SHOULD perform Node Neighbor search.

4.67.6.10   Node Neighbor Update Status Command

The Node Neighbor Update Status Command Status of the Request Node Neighbor Update Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_NODE_NEIGHBOR_UPDATE_STATUS							
Seq. No							
Status							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Status field indicating the result of the operation.

Status values	Comment
NEIGHBOR_UPDATE_STATUS_DONE	New neighbor list received
NEIGHBOR_UPDATE_STATUS_FAIL	Getting new neighbor list failed

4.67.6.11   Return Route Assign Command

An application MAY call the Return Route Assign Command to make a controller assign static return routes (up to 4) to a Routing Slave node or Enhanced Slave node. This allows the Routing Slave node to communicate directly with either controllers or other slave nodes.

Up to 5 different destinations can be allocated return routes. Attempts to assign new return routes when all 5 destinations already are allocated will be ignored.
Allocated return routes can only be cleared by the call Return Route Delete.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_RETURN_ROUTE_ASSIGN							
Seq. No							
Source Node ID							
Destination Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Source Node ID (1 byte)

The controller calculates the shortest routes from the Routing Slave node (Source Node ID) to the destination node (Destination Node ID) and transmits the return routes to the Routing Slave node (Source Node ID).

Destination Node ID (1 byte)

Refer to Source Node ID.

CONFIDENTIAL

4.67.6.12   Return Route Assign Complete Command

The Return Route Assign Complete Command indicates status of the Return Route Assign Command. The call indicates that the function completed without errors in the communication. The call MAY have been ignored if there was no capacity left in the Destination Node for more return routes. Refer to Return Route Delete for details on how to re-gain route capacity.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_RETURN_ROUTE_ASSIGN_COMPLETE							
Seq. No							
Status							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Status field refers to the transmission status of the command.

Option flags	Comment
TRANSMIT_COMPLETE_OK	Successfully transmitted
TRANSMIT_COMPLETE_NO_ACK	No acknowledge is received before timeout from the destination node. Acknowledge is discarded in case it is received after the time out.
TRANSMIT_COMPLETE_FAIL	Not possible to transmit data because the Z-Wave network is busy (jammed).

CONFIDENTIAL

4.67.6.13   Return Route Delete Command

An application MAY call the Return Route Delete Command to make a controller delete all static return routes from a Routing Slave node or Enhanced Slave node. Allocated return routes can only be cleared by the call Return Route Delete. All return routes are cleared by this call.

After calling Return Route Delete, a calling application MUST call Return Route Assign repeatedly to create return routes for all relevant associations.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_RETURN_ROUTE_DELETE							
Seq. No							
Node ID							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Node ID (1 byte)

The call deletes all return routes in the routing slave identified by the Node ID.

4.67.6.14   Return Route Delete Complete Command

The Return Route Delete Complete Command indicates status of the Return Route Delete Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_INCLUSION							
Command = COMMAND_RETURN_ROUTE_DELETE_COMPLETE							
Seq. No							
Status							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

CONFIDENTIAL

Status (1 byte)

Status field refers to the transmission status of the command.

Option flags	Comment
TRANSMIT_COMPLETE_OK	Successfully transmitted
TRANSMIT_COMPLETE_NO_ACK	No acknowledge is received before timeout from the destination node. Acknowledge is discarded in case it is received after the time out.
TRANSMIT_COMPLETE_FAIL	Not possible to transmit data because the Z-Wave network is busy (jammed).

CONFIDENTIAL

4.67.7   Network Management Primary Command Class, version 1

The Network Management Primary Command Class provides functions to pass on the primary role to another controller.

4.67.7.1   Controller Change Command

The Controller Change Command is used to add a controller node to the network and make the new controller primary.

This function has the same functionality as Node Add with the exception that the new controller will be a primary controller and the controller invoking the function will become secondary.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_PRIMARY							
Command = COMMAND_CONTROLLER_CHANGE							
Seq. No							
Reserved							
Mode							
tx Options							

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Mode (1 byte)

Modes	Comment
CONTROLLER_CHANGE_START	Start the process of creating a new primary controller for the network
CONTROLLER_CHANGE_STOP	Stop the controller change and report a failure

Table 22, Encoding of Controller Change :: Mode parameter

The process of adding a new controller node and transferring control to the new controller node is started by the application sending a Controller Change command. The application receives a status message indicating if the process was successful.

tx Options (1 byte)

The tx Options field allows a controlling node to specify if transmissions MUST use special properties. Several flags MAY be combined via a logical OR.

Option flags	Comment
NULL	Transmit at normal power level without any transmit options.
TRANSMIT_OPTION_EXPLORE	Resolve new routes via explorer discovery if existing routes fail
TRANSMIT_OPTION_LOW_POWER	Transmit at low output power level (1/3 of normal RF range)

Table 23, Encoding of Controller Change :: Tx Options

4.67.7.2 Controller Change Status Command

The Controller Change Status Command is used to report the result of the Controller Change Command.

To avoid re-entrance issues and transmit queue overflows, the Controller Change Status message SHOULD be issued before or after processing the Controller Change message but never during the execution of the function.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NETWORK_MANAGEMENT_PRIMARY							
Command = COMMAND_CONTROLLER_CHANGE_STATUS							
Seq. No							
Status							
Reserved							
NewNodeID							
NodeInfoLength							
List.	Capability						
Opt. Func.	Security						
Basic Device Class							
Generic Device Class							
Specific Device Class							
Command Class 1 *)							
...							
Command Class n *)							

*) Command classes MAY be extended ⇒ spanning two bytes for one command class

Seq. No (1 byte)

The sending node MUST insert a unique sequence number starting from a random value. Each new message MUST be an increment of the previous message. When receiving response to this request the node MUST verify that the response carries the same sequence number.

Status (1 byte)

Indicates the status of the Controller Change process. The constant labels defined for the Node Add command are reused for the Controller Change Status message.

Status values	Comment
NODE_ADD_STATUS_DONE	The new node has now been included.
NODE_ADD_STATUS_FAILED	The process failed
NODE_ADD_STATUS_SECURITY_FAILED	Node has been included but the secure inclusion failed.

Table 24, Encoding of Controller Change Status :: Status parameter

Reserved (1 byte)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

New Node ID (1 byte)

The nodeID of the newly added node. Only valid if Status is NODE_ADD_STATUS_DONE; else this field MUST be zero.

Node Info Length (1 byte)

The length of the encapsulated Node Information. The length value includes the length field.

If status is NODE_ADD_STATUS_DONE, the Node Information fields carry valid information. Else this field MUST be ignored.

Node Info (n bytes)

The node info structure of the newly added node. The length is signaled in Node Info Length.

If status is NODE_ADD_STATUS_DONE, the Node Information fields carry valid information. Else this field MUST be ignored.

Basic Device Class (1 byte)

Generic Device Class (1 byte)

Specific Device Class (1 byte)

CONFIDENTIAL

Command Class (n bytes)

Variable-length part of the frame carrying one or more command classes implemented by the actual node. Any command classes listed before a mark is Supported Non-Secure. When a 0xEF is encounter the following command classes are Controlled Non-Secure. If 0xF1->0xFF is encounter they are extended command classes and the following byte MUST be included. 0xFFF1 is a Security Scheme 0 Command Class Mark that indicates that all following command classes are supported using Scheme 0 Security, after a Security Scheme 0 Mark a Command Class Mark and additionally be used to indicate Control for Scheme 0 Security Command Classes.

Command Class ID	Comment
0xF1->0xFF	Extended Command Classes, represented by two bytes – ID is made up of MS byte being 0xF1->0xFF and LS byte 0x00->0xFF
0xEF	Command Class Mark – Anything that follows this mark and until next mark is Controlled and not supported
0xF100	Security Scheme 0 Command Class Mark – Anything that follows this mark and until next mark is supported using Security Scheme 0

4.67.8 Use Cases

4.67.8.1 Intranode network management: TV OSD System controlling lamps

Intranode network management is the process closest to classic Z-Wave API programming. No messages ever leave the device. Messages only flow between different software modules.

Use Case: TV OSD System (island mode)

Using UDP/IP for carrying the messages allows for a simple integration interface between applications designed by different partners.

CONFIDENTIAL

4.67.8.2   Intranet network management: Remote controlling a primary controller

Intranet network management extends the use of command messages to separate physical devices. Messages flow between software modules but the modules reside in separate physical entities having individual IP addresses – or at least separate node IDs.

Use Case: Managing a primary static controller from a remote control
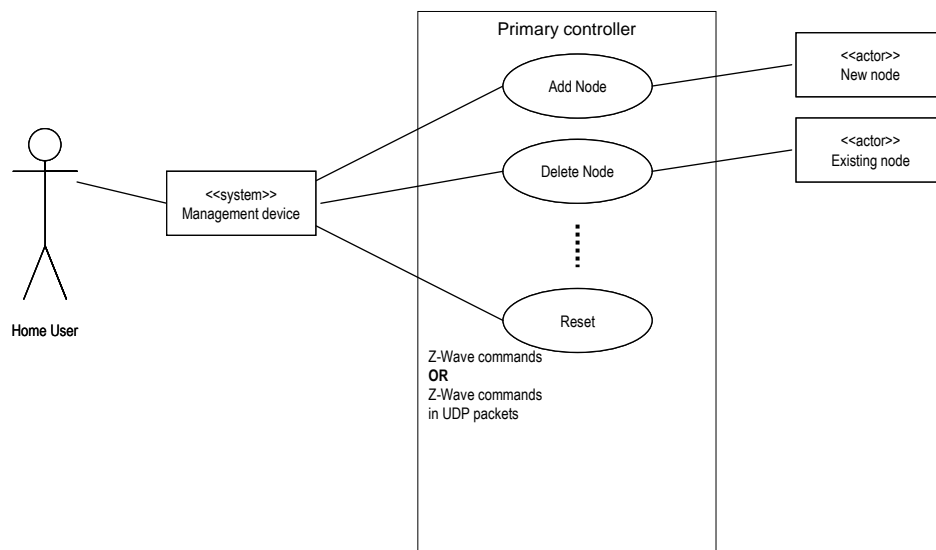
Network management via messages allows for sophisticated interfaces to the primary controller of a network. Controllers with SUC/SIS capability MAY also leverage from the Network Management command classes.

CONFIDENTIAL

4.67.8.3   Internet network management #1: Call-center support for TV OSD user

Internet network management uses the same command messages. Messages flow between software modules but the modules reside in separate physical entities in a non-trusted environment such as the Internet. Remote access technologies SHOULD be used to protect the communication.

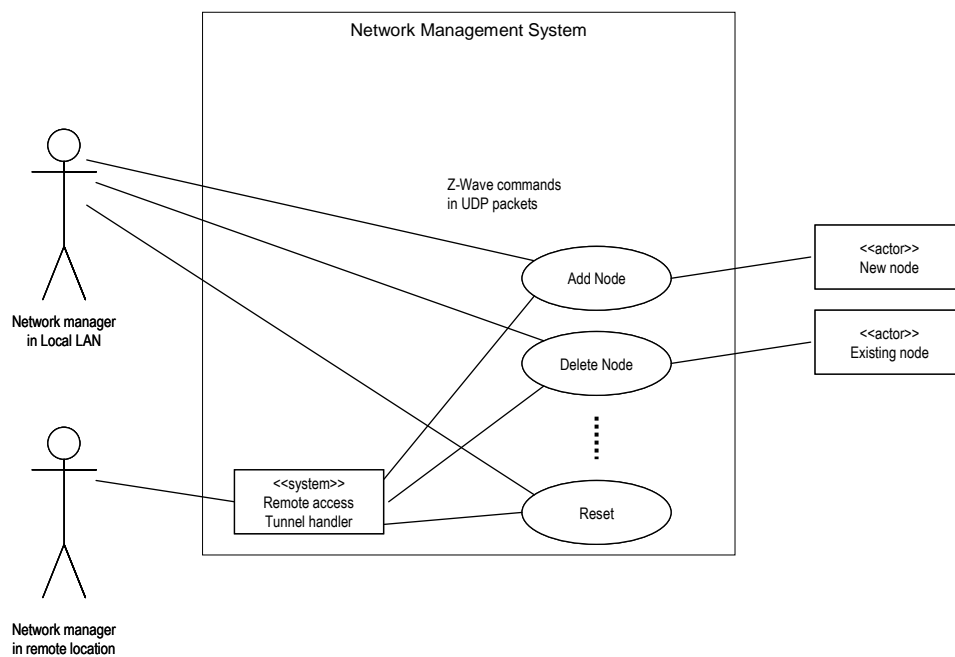In this use case a TV user MAY call the service provider for support in adding a new lamp to the network.

Use Case: TV OSD System (Connected)

CONFIDENTIAL

4.67.8.4   Internet network management #2: Remote management of Z/IP Network

In this use case a skilled user MAY use an IP based home control management system running in the LAN for setting up the Z/IP network. The user MAY use normal UDP transport in the LAN environment. Due to the critical nature of the network management command classes the user however SHOULD use remote access protection technologies over LAN as well as over Internet. The benefit of designing a home control system using remote access protection by default is that it MAY be moved from a location in the LAN to any place in the Internet and work completely unaffected.

Use Case: Z/IP Router in Consumer Premises

CONFIDENTIAL

4.67.8.5   Traffic flow: Gathering node information

The following sequence diagram introduces a new concept of gathering Node Information.

The node list provides an overview of the nodes in the network; as good as the Z/IP gateway can provide this information. Using that node list, the requesting host MAY request information on individual nodes from the Z/IP Gateway. The "Node Info Cached Get" command reports all supported and controlled classes.

CONFIDENTIAL

4.67.8.6 Traffic flow: Z/IP Gateway acts as proxy for Z-Wave SUC or Primary

CONFIDENTIAL

4.68 No Operation Command Class, version 1

The No Operation Command Class used to check if a node is reachable by sending a Command less frame to the specified destination. Feature used by the Z-Wave protocol in many situations e.g. checking that an excluded node is non-responding. This Command can also be used on application level e.g. checking if a SUC/SIS is reachable from a new node in the network. This command class contains no command identifier and data.

Notice: It is not necessary to announce the No Operation Command Class in the NIF.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NO_OPERATION							

4.69 Node Naming and Location Command Class, version 1

The Node Naming and Location Command Class used to assign a name and a location text string to all nodes in a Z-Wave network. The text strings MUST be stored in non-volatile memory by the application in the actual devices and can be requested by any other node in the network.

Notice: Please be aware that routing slaves based on future chip series can have limitations on non-volatile memory. It might then only be possible to base a design with the Node Naming and Location Command Class on libraries using an external EEPROM.

4.69.1 Node Name Set Command

The Node Name Set Command used to set the name of a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NODE_NAMING							
Command = NODE_NAMING_NODE_NAME_SET							
Reserved				Char. Presentation			
Node name char 1							
Node name char 2							
…							
Node name char x							

Node name char 1-x (variable)

Node name using specified character representation. The Node name can have a maximum of 16 characters. The number of character fields transmitted can be determined from the frame length. If a frame with more than 16 characters is received only the first 16 characters MUST be accept. The remaining characters MUST be ignored.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Char. Presentation (3 bits)

The char presentation identifier can be set to the following values:

Char. Presentation	Description
0	Using standard ASCII codes, see Appendix B (values 128-255 are ignored)
1	Using standard and OEM Extended ASCII codes, see Appendix B
2	Unicode UTF-16

CONFIDENTIAL

Note: Devices supporting Unicode UTF-16 characters can have strings of a maximum of 8 characters because each character is described by a 2 byte long decimal representation. The first byte is the most significant byte. E.g. if there is one Unicode character in the set frame the char 1 will be MSB and char 2 will be LSB of the Unicode character.

This list MAY evolve in the future. Undefined values of the character presentation identifier MUST be ignored.

4.69.2 Node Name Get Command

The Node Name Get Command is used to request the stored name from a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NODE_NAMING							
Command = NODE_NAMING_NODE_NAME_GET							

4.69.3 Node Name Report Command

The Node Name Report returns the stored node name from a node requested by the Node Name Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NODE_NAMING							
Command = NODE_NAMING_NODE_NAME_REPORT							
Reserved					Char. Presentation		
Node name char 1							
Node name char 2							
…							
Node name char x							

Node name char 1-x (variable)

Node name using specified character representation. The number of characters transmitted can be determined from the length field in the frame.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Char. Presentation (3 bits)

Refer to the description under the Node Name Set Command.

CONFIDENTIAL

4.69.4 Node Location Set Command

The Node Location Set Command used to set a location name in a node in a Z-Wave network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NODE_NAMING							
Command = NODE_NAMING_NODE_LOCATION _SET							
Reserved					Char. Presentation		
Node location char 1							
Node location char 2							
…							
Node location char x							

Node location char 1-x (variable)

Node location using specified character representation. The Node location can have a maximum of 16 characters. The number of character fields transmitted can be determined from the frame length. If a frame with more than 16 characters is received only the first 16 characters MUST be accept. The remaining characters MUST be ignored.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Char. Presentation (3 bits)

Refer to the description under the Node Name Set Command.

4.69.5 Node Location Get Command

The Node Location Command is used to request the stored node location from a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NODE_NAMING							
Command = NODE_NAMING_NODE_LOCATION_GET							

4.69.6 Node Location Report Command

The Node Location Report Command used to report the stored node location from a node requested by the Node Location Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NODE_NAMING							
Command = NODE_NAMING_NODE_LOCATION _REPORT							
Reserved					Char. Presentation		
Node location char 1							
Node location char 2							
…							
Node location char x							

Node location char 1-x (variable)

Node name using specified character representation. The number of characters transmitted can be determined from the length field in the frame.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Char. Presentation (3 bits)

Refer to the description under the Node Name Set Command.

4.70   Notification Command Class (version 3 of Alarm Command Class)

The Notification Command Class version 3 is extended from Alarm Command Class version 2 and covers the following:

- Additional Notification Types and Events defined by the Z-Wave Alliance
- Interview process includes Events
- An identifier to signal support for Event Parameters (which is now optional to support)
- Sequence field added for collection management of reports
- Notification Status can now indicate "no pending notifications"

Commands not mentioned in this document will remain the same as specified for Alarm Command Class version 1 and/or Alarm Command Class version 2.

NOTE: Be aware that the CC identifier for Notification CC V3 is the same as the Alarm CC V1 and V2, thus this new CC work is starting from revision 3. With such approach this new class will be backwards compatible with existing and Z-Wave certified products implementing support and/or control of Alarm CC V1 and V2.

IMPORTANT: When Alarm CC V1 fields are supported (implemented in OEM application), it is mandatory to map as many proprietary alarm types and levels as possible to an appropriate notification type and event of Notification CC V3. In addition, all alarm types and levels of Alarm CC V1 MUST be described in the product manual.

4.70.1   Notification Set Command

The Notification Set Command is used to set the status of the specific Notification Type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_SET							
Notification Type							
Notification Status							

Notification Type (8 bit)

See "Notification Report" command.

Notification Status (8 bit)

This field is used to set the status of the Notification Type. Valid values:

- 0x00 = deactivate the notification. Function is deactivated and/or no unsolicited Notification Report Commands will be transmitted.

- 0x01-0x0FE = reserved.

- 0xFF = activate the notification. Function is activated and/or unsolicited Notification Report Command will be transmitted. The destination node IDs to receive the unsolicited reports depends on the association settings by means of Association CC.

CONFIDENTIAL

NOTE a: All Z-Wave enabled and certified products MUST be able to operate based on a preset of configuration settings i.e. an end-user MUST not be forced to set-up the states of a device in order to operate. Factory default settings of a product MUST be described in the product manual.
Factory default setting of the Notification Status parameter to its corresponding Notification type(s) should be "activated".

NOTE b: For products that do not accept deactivation of a specific Notification Type, a received Notification Set (Notification Type = x; Notification Status = 0x00) MUST be responded with an **Application Rejected Request Command** of the Application Status CC, thus the Application Status CC MUST be advertised in the Node Information Frame.

4.70.2   Notification Get Command

The Notification Get Command is used to request for status of a specific Notification Type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_GET							
V1 Alarm Type							
Notification Type							
Event							

V1 Alarm Type (8 bit)

This field refers to the Alarm Type of Alarm CC V1 i.e. the application specific and proprietary alarm types that are not defined by the Z-Wave Alliance. If the 'V1 Alarm' field is set to '0' as reported via the Alarm Type Supported Report Command, this field MUST be set to '0'.

Notification Type (8 bit) & Event (8 bit)

See "Table of defined Notification Types & Events".

Notification Type = 0xFF MUST be accepted by the device implementing support for Notification CC V3. When Notification Type = 0xFF, the Event field MUST be set to '0' in the Notification Get command.

CAUTION: A device receiving a Notification Get Command containing a non-supported Notification Type MUST ignore the command. A device receiving a Notification Get Command containing a supported Notification Type but a non-supported Notification Event MUST respond with "Unknown Event".
The transmitter of the Notification Get Command MUST interview the device for supported Notification Type(s) by means of Notification Type Supported Get Command to correctly interview the device.

4.70.3   Notification Report Command

The Notification Report Command MUST be transmitted as a response to the Notification Get Command and in addition it may be transmitted unsolicited in case the OEM application deems this necessary.
In case unsolicited transmissions are implemented the OEM application MUST also support Association Command Class.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_REPORT							
V1 Alarm Type							
V1 Alarm Level							
Zensor Net Source Node ID							
Notification Status							
Notification Type							
Event							
Sequence	Reserved		Event Parameters Length				
Event Parameter 1							
...							
Event Parameter N							
Sequence Number							

V1 Alarm Type (8 bit) & V1 Alarm Level (8 bit)

These fields refer to the Alarm Type and Alarm Level of Alarm CC V1 i.e. the application specific and proprietary alarm types that are not defined by the Z-Wave Alliance. If not supported the fields MUST be set to '0'.

Zensor Net Source Node ID (8 bit)

In Zensor Net it is not possible to determine the Source Node ID due to the broadcast forwarded frame is without this information on protocol level. This field allows the OEM application to specify which Zensor Net Source Node ID detected the alarm. If Zensor Net is not supported this field MUST be set to '0'.

Notification Status (8 bit)

- 0x00 = unsolicited notification is deactivated.

- 0x01-0x0FD = Invalid/Reserved.

- 0xFE = No pending notifications.

- 0xFF = unsolicited notification is activated.

NOTE: If Notification Status = 0xFE, no further fields will be appended to the report, thus the report command is structured as Notification Report (V1 Alarm Type = 0, V1 Alarm Level = 0, Zensor Net Source Node ID = 0, Notification Status = 0xFE).

CONFIDENTIAL

Notification Type (8 bit)

See "Table of defined Notification Types & Events".

Event (8 bit)

See "Table of defined Notification Types & Events".

Sequence (1 bit)

0 = No "Sequence Number" field is appended after the "Event Parameter" field(s).

1 = A "Sequence Number" field is appended after the "Event Parameter" field(s).

Event Parameters Length (5 bit)

Indicates the length of the Event Parameters fields.

0 = No Event Parameter field(s) is/are appended after the Event Parameters Length field.

1-31 = Number of Event Parameter field(s) appended after the Event Parameters Length field.

Event Parameter 1 … Event Parameter N (N * Bytes)

If the "Event Parameters Length" field is not equal to '0', these field(s) contains the encapsulated information available to the "Notification Type" AND "Event".

NOTE: It is not allowed to deviate from the defined usage of the Event Parameter(s) as described in the "Table of defined Notification Types & Events".

Sequence Number (8 bit)

If the "Sequence" bit is set, the Sequence Number field contains an integer value (0-255) that MUST be incremented for each distinct Notification Report generated and transmitted.
The Sequence Number MUST be a sequence starting from 1 and MUST be related to the specific Notification Type AND Event i.e. each combination of Notification Type AND Event MUST have its own sequence number poll.

4.70.3.1   Table of defined Notification Types & Events

This table specifies the Notification Types and its <u>subordinate</u> parameters defined by the Z-Wave Alliance. New entries can be requested through Z-Wave Support at Sigma Designs. The fields that do not contain any definition for Notification Type MUST be set to '0' in the Notification Report Command. The implemented Notification Type(s) and Event(s) MUST be described in product manual.

NOTE: Event Parameters are only appended when the "Event Parameters Length" field is not equal to '0'. When appending Event Parameter(s) the device MUST advertise support of the corresponding Command Class, which is included/encapsulated into the specific Notification Type AND Event. Example: for Smoke Alarm (0x01), Smoke detected (0x01) the Node Naming and Location Command Class MUST be advertised as supported in the Node Information Frame. The Event Parameter(s) MUST include the complete command structure i.e. Command Class ID, Command ID and associated Command parameters.

CAUTION!
- Be careful not to exceed the max payload size.
- Values not listed in the Notification Type and Event table are reserved.

Notification Type (8 bit)		Event (8 bit)		Event Parameter(s) (N Byte)
Smoke Alarm	0x01	Smoke detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1).
		Smoke detected, Unknown Location	0x02	
		Smoke Alarm Test (V3)	0x03	
		Unknown Event	0xFE	
CO Alarm	0x02	Carbon monoxide detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Carbon monoxide detected, Unknown Location	0x02	
		Unknown Event	0xFE	
CO_2 Alarm	0x03	Carbon dioxide detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Carbon dioxide detected, Unknown Location	0x02	
		Unknown Event	0xFE	

CONFIDENTIAL

Notification Type (8 bit)		Event (8 bit)		Event Parameter(s) (N Byte)
Heat Alarm	0x04	Overheat detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Overheat detected, Unknown Location	0x02	
		Rapid Temperature Rise	0x03	Node Location Report (Node Naming and Location Command Class, version 1)
		Rapid Temperature Rise, Unknown Location	0x04	
		Under heat detected	0x05	Node Location Report (Node Naming and Location Command Class, version 1)
		Under heat detected, Unknown Location	0x06	
		Unknown Event	0xFE	
Water Alarm	0x05	Water Leak detected	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Water Leak detected, Unknown Location	0x02	
		Water Level Dropped	0x03	Node Location Report (Node Naming and Location Command Class, version 1)
		Water Level Dropped, Unknown Location	0x04	
		Unknown Event	0xFE	
Access Control	0x06	Manual Lock Operation	0x01	
		Manual Unlock Operation	0x02	
		Manual Not Fully Locked Operation (V3)	0x07	
		RF Lock Operation	0x03	
		RF Unlock Operation	0x04	
		RF Not Fully Locked Operation (V3)	0x08	

Notification Type (8 bit)		Event (8 bit)		Event Parameter(s) (N Byte)
		Keypad Lock Operation	0x05	User ID, User Code Report (User Code Command Class)
		Keypad Unlock Operation	0x06	User ID, User Code Report (User Code Command Class)
		Auto Lock Locked Operation (V3)	0x09	
		Auto Lock Not Fully Operation (V3)	0x0A	
		Lock Jammed (V3)	0x0B	
		All user codes deleted (V3)	0x0C	
		Single user code deleted (V3)	0x0D	
		New user code added (V3)	0x0E	
		New user code not added due to duplicate code (V3)	0x0F	
		Keypad temporary disabled (V3)	0x10	
		Keypad busy (V3)	0x11	
		New Program code Entered - Unique code for lock configuration (V3)	0x12	
		Manually Enter user Access code exceeds code limit (V3)	0x13	
		Unlock By RF with invalid user code (V3)	0x14	
		Locked by RF with invalid user codes (V3)	0x15	
		Window/Door is open (V3)	0x16	
		Window/Door is closed (V3)	0x17	

Notification Type (8 bit)		Event (8 bit)		Event Parameter(s) (N Byte)
		Unknown Event	0xFE	
Home Security	0x07	Intrusion	0x01	Node Location Report (Node Naming and Location Command Class, version 1)
		Intrusion, Unknown Location	0x02	
		Tampering, Product covering removed	0x03	
		Tampering, Invalid Code	0x04	
		Glass Breakage	0x05	Node Location Report (Node Naming and Location Command Class, version 1)
		Glass Breakage, Unknown Location	0x06	
		Motion Detection	0x07	Node Location Report (Node Naming and Location Command Class, version 1)
		Unknown Event	0xFE	
Power Management	0x08	Power has been applied	0x01	
		AC mains disconnected	0x02	
		AC mains re-connected	0x03	
		Surge detected	0x04	
		Voltage Drop/Drift	0x05	
		Over-current detected (V3)	0x06	
		Over-voltage detected (V3)	0x07	
		Over-load detected (V3)	0x08	
		Load error (V3)	0x09	
		Replace battery soon (V3)	0x0A	

Notification Type (8 bit)		Event (8 bit)		Event Parameter(s) (N Byte)
		Replace battery now (V3)	0x0B	
		Unknown Event	0xFE	
System	0x09	System hardware failure	0x01	
		System software failure	0x02	
		System hardware failure with OEM proprietary failure code (V3)	0x03	OEM proprietary system failure codes.
		System software failure with OEM proprietary failure code (V3)	0x04	OEM proprietary system failure codes.
		Unknown Event	0xFE	
Emergency Alarm	0x0A	Contact Police	0x01	
		Contact Fire Service	0x02	
		Contact Medical Service	0x03	
		Unknown Event	0xFE	
Clock	0x0B	Wake Up Alert	0x01	
		Timer Ended (V3)	0x02	
		Unknown Event	0xFE	
Return first detected Notification on supported list	0xFF			

4.70.3.2   Return first detected Notification on supported list

Notification Type = 0xFF is used by the Notification Get Command to retrieve the first (1st) detected and/or triggered Notification Type from the supported list.

When receiving a Notification Get (V1 Alarm Type = 0x00, Notification Type = 0xFF, Event = 0) the device MUST always respond with the first (1st) active event that it finds in its internal list.

Example: a device supports the Notification Types: Smoke, CO_2 and Heat.
The Heat Alarm is active e.g. overheat has been detected and subsequently it receives a Notification Get (V1 Alarm Type = 0x00, Notification Type = 0xFF, Event = 0). The device MUST then return Notification Report (…, Notification Type = 0x04, Event = 0x01/0x02) and associated Event Parameter(s).

In case the event is a persistent alarm, the device SHALL report the same event on subsequent received Notification Get (V1 Alarm Type = 0x00, Notification Type = 0xFF, Event = 0).
Two actions can cause the device to report other events:

CONFIDENTIAL

1. The Notification Type of the persistent event is deactivated by means of Notification Set command or

2. Another event with higher priority is activated (priority is application specific).

NOTE: In case no pending Notification Report exists at the time when a Notification Get (V1 Alarm Type = 0x00, Notification Type = 0xFF, Event = 0) is received, the report command is structured as Notification Report (V1 Alarm Type = 0, V1 Alarm Level = 0, Zensor Net Source Node ID = 0, Notification Status = 0xFE).

4.70.3.3   Handling GET commands from Alarm CC V1 and V2

The differences between V1-V3 Alarm/Notification Get commands are:

1. V1: Alarm Get (Alarm Type = x)

2. V2: Alarm Get (Alarm Type = x, Z-Wave Alarm Type = y)

3. V3: Notification Get (V1 Alarm Type = x, Notification Type = y, Event = z)

A device implementing support for Notification CC V3 MUST also support all previous revisions to maintain backwards compatibility and shall use the following guidelines:

I.   When a V1 Alarm Get command is received AND the device do not supports V1 Alarm Type i.e. proprietary alarm types, the device MUST reply with a V1 Alarm Report (Alarm Type = 0, Alarm Level = 0).

II. When a V2 Alarm Get command is received AND the device do not supports V1 Alarm Type AND the requested Z-Wave Alarm Type is supported BUT no event is pending, the device MUST reply with a V2 Alarm Report (Alarm Type = 0, Alarm Level = 0, Zensor Net Source Node ID = n, Z-Wave Alarm Status = 0x00/0xFF, Z-Wave Alarm Type = "the requested", Z-Wave Alarm Event = 0xFE, Number of Event Parameters = 0), as a compromise for Alarm CC V2 not having support for "no pending updates" indication in the Alarm Status field.

III. When a V2 Alarm Get command is received AND the device do not supports V1 Alarm Type AND the requested Z-Wave Alarm Type = 0xFF AND the pending updates are a mix of V2 + V3 events, the device MUST only reply with V2 events. For pending V3 events, the device MUST reply with "unknown event".

CONFIDENTIAL

4.70.4   Notification Supported Get Command

The Notification Supported Get Command is used to request for the supported Notification Types.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_SUPPORTED_GET							

4.70.5   Notification Supported Report Command

The Notification Supported Report Command is used to report the supported Notification Types in the application. The Notification Supported Report Command is transmitted as a result of a received Notification Supported Get Command and MUST not be sent unsolicited.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = NOTIFICATION_SUPPORTED_REPORT							
V1 Alarm	Reserved		Number of Bit Masks				
Bit Mask 1							
...							
Bit Mask N							

V1 Alarm (1 bit)

0 = the device implements the Alarm Type and Alarm Level of Alarm CC V1 i.e. the application specific and proprietary alarm types that are not defined by the Z-Wave Alliance.

1 = the device implements Notification Type(s) defined by the Z-Wave Alliance.

Number of Bit Masks (5 bit)

Indicates the Number of Bit Masks fields (1-31) used in bytes.

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported **Notification Type(s)** by the device.

Bit 1 in Bit Mask 1 field is used to indicate support of Notification Type = Smoke Alarm (0x01).
Bit 2 in Bit Mask 1 field is used to indicate support of Notification Type = CO Alarm (0x02).
Bit 3 in Bit Mask 1 field is used to indicate support of Notification Type = CO_2 Alarm (0x03) and so forth.
The Notification Type is supported if the corresponding bit is set to '1' and the opposite if set to '0'.
Notification Type = 0xFF cannot be indicated by the Bit Masks fields.

NOTE: It is only necessary to transmit Bit Mask 1 and up to the Bit Mask N indicating the last supported notification type.

4.70.6 Event Supported Get Command

The Event Supported Get Command is used to request the supported Events within a supported Notification Type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = EVENT_SUPPORTED_GET							
Notification Type							

4.70.7 Event Supported Report Command

The Event Supported Report Command is transmitted as a result of a received Event Supported Get Command and MUST not be sent unsolicited. If an Event Supported Get is received with a not supported Notification Type or Notification Type = 0xFF, the device MUST respond with Event Supported Report (Notification Type = the requested, number of bitmask = 0).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_NOTIFICATION							
Command = EVENT_SUPPORTED_REPORT							
Notification Type							
Reserved			Number of Bit Masks				
Bit Mask 1							
...							
Bit Mask N							

Notification Type (8 bit)

See "Table of defined Notification Types & Events".

Number of Bit Masks (5 bit)

Indicates the Number of Bit Masks fields (1-31) used in bytes.

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported **Events within the requested Notification Type**.
Example if Notification Type = Heat Alarm (0x04):

Bit 1 in Bit Mask 1 field is used to indicate support of Notification Type = Overheat detected (0x01).
Bit 2 in Bit Mask 1 field is used to indicate support of Notification Type = Overheat, unknown loc. (0x02).
Bit 3 in Bit Mask 1 field is used to indicate support of Notification Type = Rapid temp rise (0x03) and so forth. The Event is supported if the corresponding bit is set to '1' and the opposite if set to '0'.

NOTE: It is only necessary to transmit Bit Mask 1 and up to the Bit Mask N indicating the last supported notification type.

CONFIDENTIAL

4.71   Powerlevel Command Class, version 1

The Powerlevel Command Class defines RF transmit power controlling Commands useful when installing or testing a network. The Commands makes it possible for supporting controllers to set/get the RF transmit power level of a node and test specific links between nodes with a specific RF transmit power level.

NOTE: This Command Class is only used in an installation or test situation.

4.71.1   Powerlevel Set Command

The Powerlevel Set Command used to set the power level indicator value, which SHOULD be used by the node when transmitting RF, and the timeout for this power level indicator value before returning the power level defined by the application.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_POWERLEVEL							
Command = POWERLEVEL_SET							
Power level							
Timeout							

Power level (8 bits)

The power level indicator value to set.

Valid levels are:

- NormalPower
- minus1dBm
- minus2dBm
- minus3dBm
- minus4dBm
- minus5dBm
- minus6dBm
- minus7dBm
- minus8dBm
- minus9dBm

Timeout value is ignored if Power level is set to normalPower. The node MUST, when receiving this Command, call the ZW_SET_POWERLEVEL API function to effectuate the Command.

Timeout (8 bits)

The time in seconds the node SHOULD keep the Power level before resetting to normalPower level. It is fundamental, that the timeout IS implemented and followed by the application, for keeping the network consistent. Valid values are 1-255 resulting in timeouts from 1 second to 255 seconds.

4.71.2 Powerlevel Get Command

The Powerlevel Get Command is used to request the current power level indicator value in use by the node. The node receiving this Command SHOULD answer with a Powerlevel Report.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_POWERLEVEL							
Command = POWERLEVEL_GET							

4.71.3 Powerlevel Report Command

The Powerlevel Report Command used to report the current power level indicator value when transmitting and the timeout for this power level indicator value before returning the power level defined by the application. The Powerlevel Report Command can be send unsolicited or requested by the Powerlevel Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_POWERLEVEL							
Command = POWERLEVEL_REPORT							
Power level							
Timeout							

Power level (8 bits)

This value is the current power level indicator value in effect on the node.

Valid levels are:

- NormalPower
- minus1dBm
- minus2dBm
- minus3dBm
- minus4dBm
- minus5dBm
- minus6dBm
- minus7dBm
- minus8dBm
- minus9dBm

If the returned value is normalPower, the timeout value is ignored. The node MUST call the ZW_GET_POWERLEVEL API function to get the current power level indicator value.

Timeout (8 bits)

The time in seconds the node has back at Power level before resetting to normalPower level.

CONFIDENTIAL

4.71.4 Powerlevel Test Node Set Command

The Powerlevel Test Node Set Command used to instruct the destination node to transmit a number of test frames to the specified nodeID with the RF power level specified. After the test frame transmissions the RF power level is reset to normal and the result (number of acknowledged test frames) MUST be saved. The result of the test can be acquired with a Powerlevel Test Node Get Command, which results in a Powerlevel Test Node Report Command being transmitted.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_POWERLEVEL							
Command = POWERLEVEL_TEST_NODE_SET							
Test nodeID							
Power level							
Test frame count (MSB)							
Test frame count (LSB)							

Test nodeID (8 bits)

The test nodeID that SHOULD receive the transmitted test frames. The node MUST, when receiving this Command:

- Call the ZW_RF_POWERLEVEL_SET API function to set Power level.

- Call the ZW_SEND_TEST_FRAME API function to transmit the test frame with the Power level to Test nodeID this it MUST do the specified number of times.

- Finally the node MUST call the ZW_RF_POWERLEVEL_SET API function to reset the power level back to normal to effectuate the Command.

Power level (8 bits)

The power level indicator value to use in the test frame transmission.

Valid levels are:

- NormalPower
- minus1dBm
- minus2dBm
- minus3dBm
- minus4dBm
- minus5dBm
- minus6dBm
- minus7dBm
- minus8dBm
- minus9dBm

Test frame count (16 bits)

The Test frame count field contains the number of test frames to transmit to Test nodeID. The first byte is the most significant byte. Valid Test frame count range is 1-65535.

CONFIDENTIAL

4.71.5 Powerlevel Test Node Get Command

The Powerlevel Test Node Get Command is used to request for the result of the latest Powerlevel Test Node Set Command effectuated. The node receiving this Command SHOULD answer with a Powerlevel Test Node Report.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_POWERLEVEL							
Command = POWERLEVEL_TEST_NODE_GET							

CONFIDENTIAL

4.71.6   Powerlevel Test Node Report Command

The Powerlevel Test Node Report Command used to report the latest result of a test frame transmission started by the Powerlevel Test Node Set Command. The test report can be send either as a response to a Powerlevel Test Node Get Command or unsolicited, for example when a requested test run is done the test report can be send to the originating node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_POWERLEVEL							
Command = POWERLEVEL_TEST_NODE_REPORT							
Test NodeID							
Status of operation							
Test frame acknowledged count (MSB)							
Test frame acknowledged count (LSB)							

Test NodeID (8 bits)

This field contains the NodeID on the node, which is or has been under test. Test NodeID contains the nodeID set in the latest Powerlevel Test Node Set Command. If Test NodeID is ZW_TEST_NOT_A_NODEID then no test has been made and the Status of operation and the Test frame acknowledged count fields can be ignored.

Status of operation (8 bits)

This field contains the result of latest Powerlevel Test Node Set Command. Valid values are:

ZW_TEST_SUCCES

ZW_TEST_FAILED

ZW_TEST_INPROGRESS

ZW_TEST_IN_PROGRESS is returned if a test is still in progress. ZW_TEST_SUCCES is returned if at least 1 test frame transmission has been acknowledged (TRANSMIT_COMPLETE_OK) else ZW_TEST_FAILED (no test frame transmissions has been acknowledged) is returned.

Test frame acknowledged count (16 bits)

Number of test frames transmitted, which the Test NodeID has acknowledged. The first byte is the most significant byte.

CONFIDENTIAL

4.72   Prepayment Command Class, version 1

The Prepayment Command Class defines the Commands necessary to implement a Z-Wave encapsulation of Prepayment data and to distribute prepayment information between devices

4.72.1   Prepayment Balance Get Command

The Prepayment Balance Get Command is used to request the balance of the Prepayment.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT							
Command = PREPAYMENT_BALANCE_GET							
Balance Type		Reserved					

Balance Type (2 bit)

The field specifies which balance type is requested in the response report. The available balance types can be requested using the Prepayment Supported Get Command.

Balance Type	Value
Utility Balance	0x00
Monetary Balance	0x01
Reserved	0x02 – 0x03

4.72.2   Prepayment Balance Report Command

The Prepayment Balance Report Command is used to report the current balances. The Prepayment Balance Report Command can be sent unsolicited or requested by the Prepayment Balance Get Command.

The report includes the following main elements:

- Balance

- Debt

- Emergency Credit

The elements can be given in either monetary values or in utility units depending on the Balance Type field.

CONFIDENTIAL

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT							
Command = PREPAYMENT_BALANCE_REPORT							
Balance Type		Meter type					
Balance Precision		Scale					
Balance Value 1							
Balance Value 2							
Balance Value 3							
Balance Value 4							
Debt Precision		Reserved					
Debt 1							
Debt 2							
Debt 3							
Debt 4							
Emer. Credit Precision		Reserved					
Emer. Credit 1							
Emer. Credit 2							
Emer. Credit 3							
Emer. Credit 4							
Currency 1							
Currency 2							
Currency 3							
Debt Recovery Percentage							

Balance Type (2 bit)

The field specifies which type of balance is given in the report.

All available Balance Types can be found in the section 4.72.1 Prepayment Balance Get Command.

Meter Type (6 bit)

Meter Type specifies the type of metering device the command originates. Meter Type defined as the *Meter Type* variable; refer to section 3.3.5 for a definition of the variable.

Scale (5 bit)

The Scale used to indicate the scale (unit) of the balance, debt and emergency credit value in the report. Scale field only used for a report of type "Utility Balance", for other reports set the scale to 0x1F. The Scale parameter is of the variable type *Meter Scale*; refer to Section 4.54.13 for a definition of the variable.

CONFIDENTIAL

Currency 1 ... 3

ISO 4217 defines the currency code; this field is only used for a report of type "Monetary Balance". For other reports, these fields MUST be set to the currency code "No Currency".

The table below shows some examples of ISO4217 codes:

Currency Code	Currency 1	Currency 2	Currency 3
Pound sterling	G	B	P
US Dollar	U	S	D
No Currency	X	X	X

Balance, Debt and Emergency Credit Precision (3 bit)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Balance, Debt and Emergency Credit

The Balance, Debt and Emergency Credit are 32 bit signed fields. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

Debt Recovery Percentage (8 bit)

The Debt Recovery Percentage indicates the percentage of the payment that is for debt recovery. This can take the value from 0 – 50%, the value is always given with a precision of 0 decimal places. The value 0xFF indicates that this field is unspecified.

This field is only used for only used for a report of type "Monetary Balance". For other reports this field MUST be set to 0xFF (unspecified).

CONFIDENTIAL

4.72.3 Prepayment Supported Get Command

The Prepayment Supported Get Command is used to request type of Balance Reports that are available in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT							
Command = PREPAYMENT_SUPPORTED_GET							

4.72.4 Prepayment Supported Report Command

The Prepayment Supported Report Command reports the types of Balance Reports that are available in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT							
Command = Command = PREPAYMENT_SUPPORTED_REPORT							
Reserved				Bit Mask Balance Types Supported			

Bit Mask - Balance Types Supported (8 bit)

The Bit Mask - Balance Types Supported byte describes the type of Balance Reports that are available in the device.

Bit 0 in Bit Mask is used to indicate if the Balance Report Type "Utility Balance" is supported, 0 indicating not supported and 1 indicating supported. Bit 1 in the Bit Mask is used to indicate if the Balance Report Type "Monetary Balance" is available in the device, 0 indicating not supported and 1 indicating supported.

All available Balance Types can be found in the section 4.72.1 Prepayment Balance Get Command.

4.73   Prepayment Encapsulation Command Class, version 1

The Prepayment Encapsulation Command Class is used to smartcard preinstalled security mechanisms.

4.73.1   Prepayment Encapsulation Command

The Prepayment Encapsulation Command is used to encapsulate Smart card related data communication (e.g. between a Card reader and Meter), allowing the smartcard preinstalled security mechanisms to be applied transparent to Z-Wave.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PREPAYMENT_ENCAPSULATION							
Command = CMD_ENCAPSULATION							
Data1							
...							
Data N							

Data 1… data N (variable)

Contains prepayment smart card data.

CONFIDENTIAL

4.74   Proprietary Command Class, version 1

> **It´s RECOMMENDED using Manufacturer Proprietary Command Class instead for new devices.**

The Proprietary Command Class used to transfer data between devices. The data content MUST be vendor specific and MUST be non-value added with respect to the Home Automation application in general.

Note: Do not use the Proprietary Command Class without written approval from Sigma Designs.

4.74.1   Proprietary Set Command

The Proprietary Set Command used to transfer data to a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROPRIETARY							
Command = PROPRIETARY_SET							
Data 1							
…							
Data N							

Data 1 .. Data N (variable)

The data fields can be used to set various data in the device. Each data field is 8 bits and the maximum number of data fields in one single cast or broadcast frame is 54 bytes for a non-secure Z-Wave solution. The number of data fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.74.2 Proprietary Get Command

The Proprietary Get Command is used to request data from a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROPRIETARY							
Command = PROPRIETARY_GET							
Data 1							
…							
Data N							

Data 1 ... Data N (variable)

Refer to explanation under the Proprietary Set Command.

4.74.3 Proprietary Report Command

The Proprietary Report Command used to retrieve various data from a device. The Proprietary Report Command can be send unsolicited or requested by the Proprietary Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROPRIETARY							
Command = PROPRIETARY_REPORT							
Data 1							
…							
Data N							

Data 1 ... Data N (variable)

Refer to explanation under the Proprietary Set Command.

CONFIDENTIAL

4.75 Protection Command Class, version 1

The Protection Command Class version 1 used to protect a device against unintentionally control by e.g. a child.

4.75.1 Protection Set Command

The Protection Set Command used to set the protection state in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
PROTECTION_SET							
Protection State							

Protection State (8 bits)

The protection state field used to set the protection state of the device.

Protection State	Description
0x00	Unprotected - The device is not protected, and can be operated normally via the user interface.
0x01	Protection by sequence - The device is protected by altering the way the device normally is operated into a more complicated sequence of actions, e.g. if a device normally is controlled by a single press of a button on the device it might be changed to require 3 rapid presses on a button to control it.
0x02	No operation possible - It is not possible at all to control a device directly via the user interface.

Control via Z-Wave is always possible independent of protection state.

4.75.2 Protection Get Command

The Protection Get Command is used to request the protection state from a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_GET							

CONFIDENTIAL

4.75.3 Protection Report Command

The Protection Report Command used to report the protection state of a device. The Protection Report Command can be send unsolicited or requested by the Protection Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_REPORT							
Protection State							

Protection State (8 bits)

Refer to explanation under Protection Get Command.

CONFIDENTIAL

4.76   Protection Command Class, version 2

The Protection Command Class version 2 is extended to specify whether a device can be controlled via RF Commands or not. When a video recorder is powered by an outlet that can be controlled by RF the user would like to prevent the video recorder from being turned of when it is programmed to record her/his favorite show. In this case the Protection Command Class version 2 can be used to protect the outlet from being turned off by setting the outlet in "No RF Control" state.

The following Commands have been added or changed in version 2. The Commands not mentioned here will remain the same**.**

Notice: This command class is suggested for convenience applications. For liability reasons the command class is not RECOMMENDED for safety applications.

4.76.1   Protection Set Command

The Protection Set Command used to set the protection state in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_SET							
Reserved				Local Protection State			
Reserved				RF Protection State			

Local Protection State (4 bits)

The protection state field used to set the protection state of the device.

Local Protection State	Description
0	Unprotected - The device is not protected, and can be operated normally via the user interface.
1	Protection by sequence - The device is protected by altering the way the device normally is operated into a more complicated sequence of actions, e.g. if a device normally is controlled by a single press of a button on the device it might be changed to require 3 rapid presses on a button to control it.
2	No operation possible - It is not possible at all to control a device directly via the user interface.

Note! – Local Protection can only protect a device from "normal operation". This means only the operation that is intended by the application of the device. It is NOT allowed to protect the device from network functionalities. The device cannot be protected from being put into learn mode nor from sending out the NIF.

RF Protection State (4 bits)

CONFIDENTIAL

The RF protection state field used to set the RF protection state of the device. In the case where a device set into a RF Protection State which instructs the device not to answer to a "normal operation" Command, the device MUST return with the Application Rejected Request Command (Status = 0) from the Application Status Command Class. Please refer to 4.7 for more details about the Application Status Command Class.

RF Protection State	Description
0	Unprotected - The device MUST accept and respond to all RF Commands.
1	No RF control - all runtime Commands are ignored by the device. The device MUST still respond with status on requests.
2	No RF response at all. The device will not even reply to status requests.

Note! – It is only possible to un-protect the device with the Protection Set Command. It is not allowed ignore Protection Commands. If a device is excluded from the network, the protection states MUST be reset.

4.76.2 Protection Report Command

The Protection Report Command used to report the protection state of a device. The Protection Report Command can be send unsolicited or requested by the Protection Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_REPORT							
Reserved				Local Protection State			
Reserved				RF Protection State			

4.76.3 Protection Supported Get Command

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_SUPPORTED_GET							

CONFIDENTIAL

4.76.4   Protection Supported Report Command

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_SUPPORTED_REPORT							
Reserved						Exclusive Control	Timeout
Local Protection State Byte 1							
Local Protection State Byte 2							
RF Protection State Byte 1							
RF Protection State Byte 2							

Local Protection State Byte 1 .. 2

The list of all Local Protection States can be found in section 4.76.1. The two bytes MUST be interpreted as bit masks where byte 1 bit 0 represent Local Protection State 0, byte 1 bit 1 represent Protection State 1, byte 2 bit 0 represent Protection State 8 etc.

RF Protection State Byte 1 .. 2

The list of all RF Protection States can be found in section 4.76.1. The two bytes MUST be interpreted as bit masks where byte 1 bit 0 represent RF Protection State 0, byte 1 bit 1 represent Protection State 1, byte 2 bit 0 represent Protection State 8 etc.

Exclusive Control

When this bit is set to 1 the device support Exclusive Control. When Exclusive Control is supported the device MUST support the Commands Protection Exclusive Control Set, Get and Report described below.

Timeout

When this bit is set to 1 the device supports a timeout for RF Protection State. When the timeout is supported the device MUST support the Commands Protection Timeout Set, Get and Report described below.

CONFIDENTIAL

4.76.5 Protection Exclusive Control

The Protection Exclusive Control is an OPTIONAL feature. The Commands in this chapter can only be implemented if the device supporting Protection Command Class version 2 announces support for Exclusive Control in the Protection Supported Report Command.

4.76.5.1 Protection Exclusive Control Set Command

The Protection Exclusive Control Set Command used to set the node ID of a Z-Wave device that can override the protection state in a protected device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_EC_SET							
Node ID							

Node ID

The node ID that has exclusive control can override the RF protection state of the device and can control it regardless of the protection state. Commands from any other nodes in the network MAY be restricted by the RF protection state and, where necessary, the Application Rejected Request Command MUST be sent.

All of the Protection Command Class commands will be accepted and processed regardless of whether or not a node has exclusive control.

Factory default setting of the Node ID for exclusive control MUST be set to 0. To reset the exclusive control state in a device an Exclusive Control Set Command with Node ID 0 as parameter MUST be send to the device.

4.76.5.2 Protection Exclusive Control Get Command

The Protection Exclusive Control Get Command is used to request a Protection Exclusive Control Report Command from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_EC_GET							

4.76.5.3 Protection Exclusive Control Report Command

The Protection Exclusive Control Report Command used to return the node ID of a Z-Wave device that has exclusive control over this device in protection mode.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_EC_REPORT							
Node ID							

Node ID

See description under the Protection Exclusive Control Set Command section 4.76.5.1.

4.76.6 Protection Timeout

The Protection Timeout is an OPTIONAL feature. The Commands in this chapter can only be implemented if the device supporting Protection Command Class version 2 announces support for Timeout in the Protection Supported Report Command.

4.76.6.1 Protection Timeout Set Command

The Protection Timeout Set Command used to set the timeout for protection mode in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_TIMEOUT_SET							
Timeout							

Timeout

The timeout describe the time a device will remain in RF Protection mode.

Factory default setting for the Timeout parameter MUST be 0x00.

Timeout	Description
0x00	No timer is set. All "normal operation" Commands MUST be accepted.
0x01-0x3C	Timeout is set from 1 second (0x01) to 60 seconds (0x3C) in 1-second resolution.
0x41-0xFE	Timeout is set from 2 minutes (0x41) to 191 minutes (0xFE) in 1-minute resolution.
0xFF	No Timeout – The Device will remain in RF Protection mode infinitely.

4.76.6.2 Protection Timeout Get Command

The Protection Timeout Command is used to request a Protection Timeout Report Command from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_TIMEOUT_GET							

4.76.6.3 Protection Timeout Report Command

The Protection Timeout Report Command used to return the remaining time that a device will remain in protection mode.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_PROTECTION							
Command = PROTECTION_TIMEOUT_REPORT							
Timeout							

Timeout

Timeout	Description
0x00	No timer is set. All "normal operation" Commands MUST be accepted.
0x01-0x3C	If the remaining time for protection mode is 1 minute or less the remaining time will be returned in 1-second resolution from 1 second (0x01) to 60 seconds (0x3C).
0x41-0xFE	If the remaining time for protection mode is more than 1 minute the remaining time will be returned in 1-minute resolution from 2 minutes (0x41) to 191 minutes (0xFE).
0xFF	No Timeout is set – The Device will remain in RF Protection mode infinitely.

CONFIDENTIAL

4.77 Pulse Meter Command Class, version 1

> It´s RECOMMENDED using Meter Command Class instead for new devices.

The Pulse Meter Command Class defines the Commands necessary to implement the pulse meter functionality. The Pulse Meter Command Class is intended for all kinds of meters that generate pulses, such as gas and water meters.

4.77.1 Pulse Meter Get Command

The Pulse Meter Get Command is used to request the number of pulses that has been counted.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_PULSE							
Command = METER_PULSE_GET							

4.77.2 Pulse Meter Report Command

The Pulse Meter Report Command used to report the number of pulses detected in the device. The Pulse Meter Report Command can be send unsolicited or requested by the Pulse Meter Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_METER_PULSE							
Command = METER_PULSE_REPORT							
Pulse Count 1							
Pulse Count 2							
Pulse Count 3							
Pulse Count 4							

Pulse Count (32 bits)

The Pulse Count field contains the number of pulses generated by the meter. The first byte is the most significant byte.

CONFIDENTIAL

4.78   Rate Table Configuration Command Class, version 1

The Rate Table Configuration Command Class defines the parameter sets for a range of rates.

The Rate Table configuration commands are separated for the Rate Table monitoring commands in the rate Table Monitor Command Class, allowing the classes to be OPTIONALly supported at different Z-Wave security levels. (E.g. Rate table monitoring commands could be supported in any device, while enabling a strict and certificate based security solution for the Rate Table Configuration Command class). Please refer to the Hybrid Security Command class for more details regarding Z-Wave security levels.

Every rate is described as a combination of time, maximum consumption, maximum demand and DCP Rate ID.

Time: Time of day. E.g. 7.15am to 9.20am

Maximum Consumption: E.g. 200kWh. This allows the Utility Supplier to provide special rates for 'utility conserving' end users.

Maximum Demand: E.g. 4000W. This allows the Utility Supplier to provide special rates for end-users which manages to keep the 'peak' demand under 'control'.

DCP Rate ID: E.g. DCP Rate ID = 16. This allows the Utility Supplier to provide special rates for end-users participating in DCP event with a specific DCP Rate ID.

4.78.1   Rate Table Set Command

The Rate Table Set Command adds a *rate parameter set* to a given *rate parameter set identifier*.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_CONFIG							
Command = RATE_TBL_REPORT							
Rate Parameter Set ID							
Reserved	Rate Type		Number of Rate Char.				
Rate Character 1							
…							
Rate Character N							
Start Hour Local Time							
Start Minute Local Time							
Duration Minute 1							
Duration Minute 2							

The following part of the command is OPTIONAL depending on the parameters supported. Use the Rate Table Supported Get Command to obtain supported parameters beside the default above. A not supported parameter is not included into the command layout.

CONFIDENTIAL

Consumption Precision 1	Consumption Scale 1	
Min. Consumption Value 1		
Min. Consumption Value 2		
Min. Consumption Value 3		
Min. Consumption Value 4		
Max. Consumption Value 1		
Max. Consumption Value 2		
Max. Consumption Value 3		
Max. Consumption Value 4		
Max. Demand Precision 1	Max. Demand Scale 1	
Max. Demand Value 1		
Max. Demand Value 2		
Max. Demand Value 3		
Max. Demand Value 4		
DCP Rate ID		

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID indicates the requested parameter set.

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

Reserved (1 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Number of Rate Characters (5 bit)

Number of characters defining the rate text (1..32).

Rate Character 1 … N (Number of Rate Characters * 8 bit)

The Rate character fields hold the string identifying the rate parameter set. The character presentation uses standard ASCII codes (values 128-255 are ignored).

Start Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

CONFIDENTIAL

Duration Minute (16 bit)

Specify the duration in minutes (1-1440) since the start in local time. The value 1440 indicates that the parameter set applies the entire day.

Consumption Precision (3 bit)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Consumption Scale (5 bit)

The Consumption Scale used to indicate the scale (unit) is applicable for the rate. The Consumption Scale parameter is of the variable type *Meter Scale*; refer to Section 4.54.13 for a definition of the variable.

Minimum / Maximum Consumption Value (4 * 8 bit)

The Minimum / Maximum Consumption Value are a 32 bit unsigned field. The first byte is the most significant byte.

The Minimum / Maximum Consumption Value are used in the Rate Table when Block Tariffs are used. Block tariffs assign blocks of energy at a set cost. The billing period is defined by the Utility Supplier.

Example of Block Tariff based Rate Table: (Two Block tariff system)

The first block from 0 kWh to 200 kWh is charged at 2 USD/kWh and all other units consumed over 200kWh will be charged at 2.5 USD/kWh

> Rate1: Min Consumption value = 0kWh, Max Consumption Value = 200kWh
> Rate2: Min Consumption value = 200kWh, Max Consumption Value = 2147483648kWh

NOTE: The actual charge is set using the Tariff Table Configuration Command Class, version 1.

NOTICE: The device receiving the Rate Table Report MUST always show the value even though the Scale is not supported.

Max. Demand Precision (3 bit)

The Maximum Demand Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Max. Demand Scale (5 bit)

The Maximum Demand Scale field describes what unit is applicable for the rate. The Maximum Demand Scale parameter is of the variable type *Meter Scale*; refer to Section 4.54.13 for a definition of the variable.

Max. Demand Value (4 * 8 bit)

The Maximum Demand Value is a 32 bit unsigned field. The first byte is the most significant byte.

The maximum demand value 0xFFFFFFFF is reserved and represents an unlimited maximum demand value.

CONFIDENTIAL

NOTICE: The device receiving the Rate Table Report MUST always show the value even though the Scale is not supported.

DCP Rate ID (8 bit)

The DCP Rate ID addresses the Demand Control Plan parameters, which will overrule other parameter sets defined in the Rate Table Command Class. A DCP Rate ID equal to zero disables Demand Control Plan mapping.

4.78.2 Rate Table Remove Command

The Rate Table Remove Command is used to remove rate parameter set(s).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_CONFIG							
Command = RATE_TBL_REMOVE							
Reserved		Rate Parameter Set IDs					
Rate Parameter Set ID 1							
..							
Rate Parameter Set ID N							

Reserved (2 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Rate Parameter Set IDs (6 bit)

The rate parameter set id's indicates the number of rate parameter set id's in the command.

Rate Parameter Set ID 1 .. Rate Parameter Set ID N

These fields contain a list of Rate Parameter Set ID's that SHOULD be removed from the Rate Table. All Rate Parameter Set ID's are cleared in case no Rate Parameter Set ID's are supplied.

4.79   Rate Table Monitor Command Class, version 1

The Rate Table Monitor Command Class defines the parameter sets for a range of rates.

4.79.1   Rate Table Supported Get Command

The Rate Table Supported Get Command is used to request the number of rates and parameter sets supported by the Rate Table Command Class.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_SUPPORTED_GET							

4.79.2   Rate Table Supported Report Command

The Rate Table Supported Report Command reports the number of rates and parameter sets supported by the Rate Table Command Class and MUST be requested by the Rate Table Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = Command = RATE_TBL_SUPPORTED_REPORT							
Rates Supported							
Parameter Set Supported Bit Mask 1							
Parameter Set Supported Bit Mask 2							

Rates Supported (8 bit)

Number of rates supported (1…255).

CONFIDENTIAL

Parameter Set Supported Bit Mask 1, 2 (16 bit)

The Bit Mask field describes the supported parameter set in addition to the default-supported parameter set. The default parameter set comprises of Rate Parameter Set ID, Rate text, Start time and Duration.

It is possible to extend the default parameter set as follows:

Parameter Set Supported	Bit Map	Description
1	Bit 0	Reserved
1	Bit 1	Supports <u>Block tariffs</u> if the bit is 1 and the opposite if 0. Block tariffs assign blocks of energy at a set cost, for example in a two block tariff the first block say from 0 kWh to 200 kWh is charged at X currency per unit(kWh) all other units consumed over 200kWh will be charged at Y currency per unit for the billing period.
1	Bit 2	Supports <u>Maximum demand tariffs</u> if the bit is 1 and the opposite if 0. Maximum demand tariffs are based on the maximum load that is measured for example 20kw over an averaging period. The charge is based on the max load; hence a 30kW maximum demand would be more costly than a 20kW maximum demand.
1	Bit 3	Supports <u>Subscribed demand tariffs</u> if the bit is 1 and the opposite if 0. This type of tariff is used in France and Italy primarily; the standing charge is calculated from the maximum load, for example 10A = 10 currency/month, 20A = 20 currency / month.
1	Bit 4	Supports Demand Control Plan mapping (DCP ID) if the bit is 1 and the opposite if 0.
1	Bit 5-7	Reserved
2	Bit 0 7	Reserved

This list MAY evolve in the future.

CONFIDENTIAL

4.79.3   Rate Table Get Command

The Rate Table Get Command is used to request the rate parameter set for a given rate parameter set identifier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_GET							
Rate Parameter Set ID							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID addresses the wanted parameter set. The rate parameter set identifier MUST be a sequence starting from 1 to Rates Supported.

4.79.4   Rate Table Report Command

The Rate Table Report Command reports rate parameter set for a given rate parameter set identifier and MUST be requested by the Rate Table Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL							
Command = RATE_TBL_REPORT							
Rate Parameter Set ID							
Reserved	Rate Type		Number of Rate Char.				
Rate Character 1							
…							
Rate Character N							
Start Hour Local Time							
Start Minute Local Time							
Duration Minute 1							
Duration Minute 2							

The following part of the command is OPTIONAL depending on the parameters supported. Use the Rate Table Supported Get Command to obtain supported parameters beside the default above. A not supported parameter is removed from the command layout.

CONFIDENTIAL

Consumption Precision 1	Consumption Scale 1
Min. Consumption Value 1	
Min. Consumption Value 2	
Min. Consumption Value 3	
Min. Consumption Value 4	
Max. Consumption Value 1	
Max. Consumption Value 2	
Max. Consumption Value 3	
Max. Consumption Value 4	
Max. Demand Precision 1	Max. Demand Scale 1
Max. Demand Value 1	
Max. Demand Value 2	
Max. Demand Value 3	
Max. Demand Value 4	
DCP Rate ID	

Refer to description of fields under the Rate Table Set Command (section 4.78.1).

4.79.5   Rate Table Active Rate Get Command

The Rate Table Active Rate Get Command is used to retrieve the rate currently active in the meter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_ACTIVE_RATE _GET							

4.79.6   Rate Table Active Rate Report Command

The Rate Table Active Rate Get Command return the rate parameter set ID of the rate currently active in the meter.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_ACTIVE_RATE _REPORT							
Rate Parameter Set ID							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID of the rate currently active in the meter.

CONFIDENTIAL

4.79.7 Rate Table Current Data Get Command

The Rate Table Current Data Get Command is used to request a number of time stamped values (current) in physical units according to the dataset mask.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_CURRENT_DATA_GET							
Rate Parameter Set ID							
Dataset Requested 1							
Dataset Requested 2							
Dataset Requested 3							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID addresses the wanted parameter set. The rate parameter set identifier MUST be a sequence starting from 1 to Rates Supported.

Dataset Requested 1, 2, 3 (24 bit)

The dataset requested parameter indicates which data is requested by the command. Dataset requested parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

CONFIDENTIAL

4.79.8   Rate Table Current Data Report Command

The Rate Table Current Data Report Command used to report a number of time stamped values (current) in physical units in the device. The Rate Table Data Report Command can be sent unsolicited or requested by the Rate Table Current Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_CURRENT_DATA_REPORT							
Reports to Follow							
Rate Parameter Set ID							
Dataset 1							
Dataset 2							
Dataset 3							
Year 1							
Year 2							
Month							
Day							
Hour Local Time							
Minute Local Time							
Second Local Time							
Current Precision 1			Current Scale 1				
Current Value 1,1							
Current Value 1,2							
Current Value 1,3							
Current Value 1,4							
…							
Current Precision N			Current Scale N				
Current Value N,1							
Current Value N,2							
Current Value N,3							
Current Value N,4							

Reports to Follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

CONFIDENTIAL

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID addresses the wanted parameter set. The rate parameter set identifier MUST be a sequence starting from 1 to Rates Supported.

Dataset 1, 2, 3 (24 bit)

The dataset parameter indicates which data is included in the report. The Dataset parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that a accumulated value is not determined yet.

Day (8 bit)

Specify the day of the month between 01 and 31.

Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

Precision (3 bit)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Current Scale N (5 bit)

The Current Scale is used to indicate the scale (unit) of the following value. Current scale parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

CONFIDENTIAL

Current Value N (32 bit)

The Current Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

NOTICE: The device receiving the Rate Table Current Data Report MUST always show the value even though the Scale is not supported.

CONFIDENTIAL

4.79.9 Rate Table Historical Data Get Command

The Rate Table Historical Data Get Command is used to request a number of time stamped values (historical) in physical units according to rate type, dataset mask and time interval.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_HISTORICAL_DATA_GET							
Maximum Reports							
Rate Parameter Set ID							
Dataset Requested 1							
Dataset Requested 2							
Dataset Requested 3							
Start Year 1							
Start Year 2							
Start Month							
Start Day							
Start Hour Local Time							
Start Minute Local Time							
Start Second Local Time							
Stop Year 1							
Stop Year 2							
Stop Month							
Stop Day							
Stop Hour Local Time							
Stop Minute Local Time							
Stop Second Local Time							

Maximum Reports (8 bit)

The maximum reports parameter is used to indicate the maximum number of reports to return based on the get. Reports are always returned with the most recently recorded value first. If set to 0x00 the meter will return all reports based on the request.

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID addresses the wanted parameter set. The rate parameter set identifier MUST be a sequence starting from 1 to Rates Supported.

Dataset Requested 1, 2, 3 (24 bit)

The dataset requested parameters indicate data requested. Dataset requested parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

CONFIDENTIAL

Start/Stop Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Start/Stop Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that an accumulated value is not determined yet.

Start/Stop Day (8 bit)

Specify the day of the month between 01 and 31.

Start/Stop Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start/Stop Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Start/Stop Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

CONFIDENTIAL

4.79.10 Rate Table Historical Data Report Command

The Rate Table Historical Data Report Command used to report a number of time stamped values (historical) in physical units in the device. The Rate Table Data Report Command can be sent unsolicited or requested by the Rate Table Historical Data Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_RATE_TBL_MONITOR							
Command = RATE_TBL_HISTORICAL_DATA_REPORT							
Reports to Follow							
Rate Parameter Set ID							
Dataset 1							
Dataset 2							
Dataset 3							
Year 1							
Year 2							
Month							
Day							
Hour Local Time							
Minute Local Time							
Second Local Time							
Historical Precision 1			Historical Scale 1				
Historical Value 1,1							
Historical Value 1,2							
Historical Value 1,3							
Historical Value 1,4							
…							
Historical Precision N			Historical Scale N				
Historical Value N,1							
Historical Value N,2							
Historical Value N,3							
Historical Value N,4							

Reports to follow (8 bit)

This value indicates how many report frames there are left, the value 0xFF means that the number of reports have not been calculated yet or that there is more than 255 reports to follow.

CONFIDENTIAL

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID addresses the wanted parameter set. The rate parameter set identifier MUST be a sequence starting from 1 to Rates Supported.

Dataset 1, 2, 3 (24 bit)

The dataset parameter indicates which data is included in the report. Dataset parameters defined as the *Meter Dataset* variable; refer to section 0 for a definition of the variable.

Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that a accumulated value is not determined yet.

Day (8 bit)

Specify the day of the month between 01 and 31.

Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

Historical Precision (3 bit)

The Historical Precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Historical Scale (5 bit)

The Historical Scale used to indicate the scale (unit) of the following value. The Historical Scale parameter is of the variable type *Meter Scale*; refer to Section 4.54.13 for a definition of the variable.

CONFIDENTIAL

Historical Value

The Historical Value is a 32 bit signed field defined by dataset requested field. The first byte (Value 1) is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

NOTICE: The device receiving the Rate Table Historical Data Report MUST always show the value even though the Scale is not supported.

CONFIDENTIAL

4.80　Remote Association Activation Command Class, version 1

The Remote Association Activation Command Class used to remote activations of Association grouping identifiers in other nodes.

Mandatory requirement: Both 'local' and 'remote' node MUST implement the Association Command Class as illustrated below. In addition the 'local' node MUST implement the Remote Association Configuration Command Class as 'Supported'.

The Remote Association Activation Command Class and the Remote Association Configuration Command Class are additions to the functionality to the existing Association Command Class.
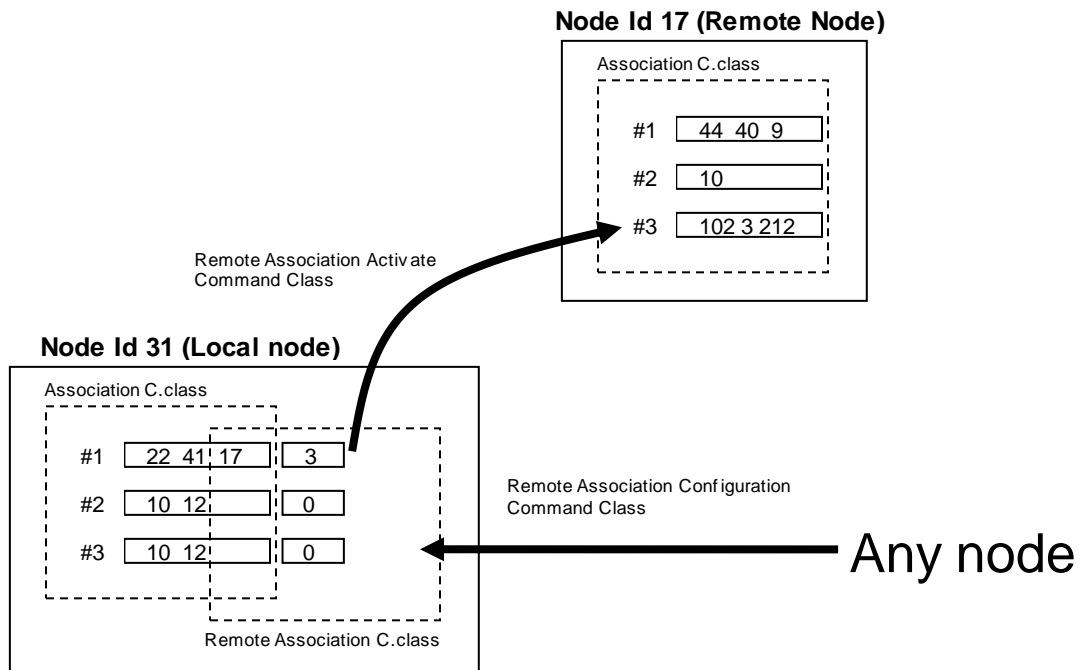
Figure 12, Remote Association Activation Command Class

The Remote Association Configuration Command Class enables a 1st node (any node) to configure a 2nd node (local node) to issue a Remote Association Activate Command to a 3rd node (remote node), which instruct the 3rd node to activate one of its locally stored association group identifiers as defined by the Association Command Class.

CONFIDENTIAL

4.80.1 Remote Association Activate Command

The Remote Association Activate Command used to instruct a 'remote' node to activate one of its locally stored association group identifiers as defined by the Association Command Class. This will subsequently generate a number of Commands being issued from the 'remote node to the NodeIDs associated to the grouping identifier

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_REMOTE_ASSOCIATION_ACTIVATE							
Command = REMOTE_ASSOCIATION_ACTIVATE							
Grouping identifier							

Grouping identifier (8 bits)

This group identifier used to specify the grouping identifier on the remote node

CONFIDENTIAL

4.81 Remote Association Configuration Command Class, version 1

The Remote Association Configuration Command Class used to configuration of the Remote Association Activation Command Class.

Mandatory requirement: Both 'local' and 'remote' node MUST implement the Association Command Class as 'supported'. In addition the 'local' node MUST implement the Remote Association Configuration Command Class as 'supported' and the node used to make the configuration ('any node' in the description below) MUST implement it as 'controlled'.

The Remote Association Configuration Command Class is an addition to the functionality of the existing Association Command Class.
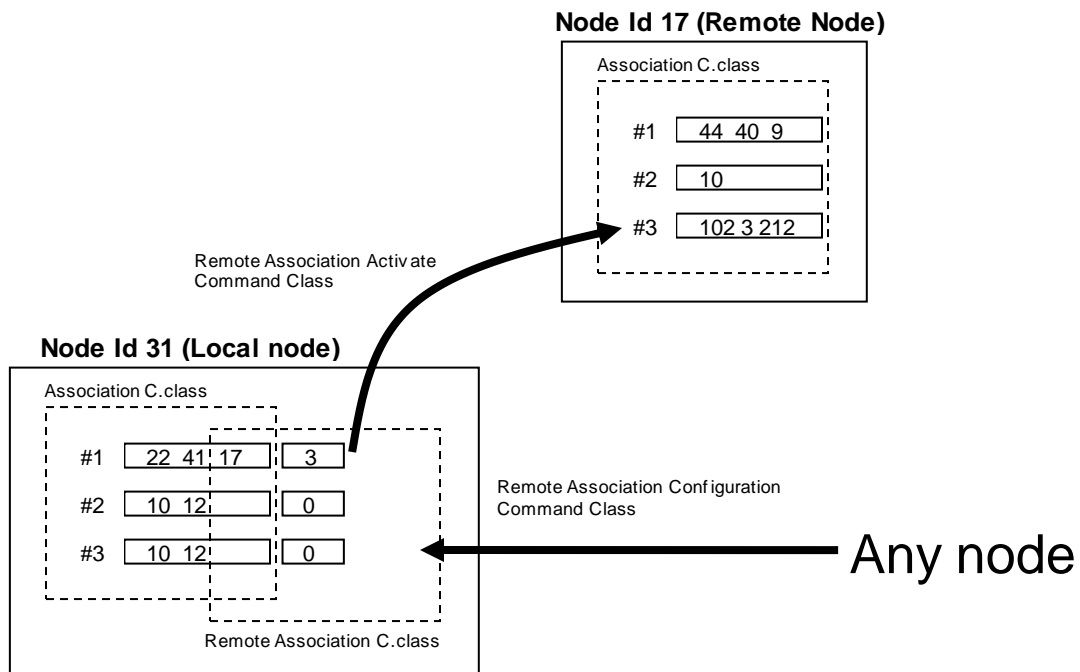
Figure 13, Remote Association Configuration Command Class

The Remote Association Configuration Command Class enables a 1st node (Any node) to configure a 2nd node (Local node) to issue a Remote Association Activate Command to a 3rd node (Remote node), which instructs the 3rd node to activate one of its locally stored association group identifiers as defined by its Association Command Class.

CONFIDENTIAL

4.81.1 Remote Association Configuration Set Command

The Remote Association Configuration Set Command links two nodes' 'Association Command Class' defined grouping identifiers together. It allows one node (local node) to use its grouping identifiers to control a second node's (remote node) grouping identifiers, using the Remote Association Activation Command Class.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_REMOTE_ASSOCIATION							
Command = REMOTE_ASSOCIATION_CONFIGURATION_SET							
Local Grouping identifier							
Remote NodeId							
Remote Grouping identifier							

Local Grouping identifier (8 bits)

This group identifier is as explained in the Association Command Class used to specify the grouping identifier on the local node.

A Local grouping identifier = 0x0 will erase all links between local and remote grouping identifiers.

Remote NodeId (8 bits)

This NodeId used to specify the Node, which SHOULD receive the Remote Association Activate Command.

A nodeId = 0x0 will remove a link between the specified local grouping identifier and a remote grouping identifier.

Remote Grouping identifier (8 bits)

This group identifier used to specify the grouping identifier on the remote node.

4.81.2 Remote Association Configuration Get Command

The Remote Association Configuration Get Command is used to request the link between a Local Grouping identifier and a Remote Grouping identifier on a node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_REMOTE_ASSOCIATION							
Command = REMOTE_ASSOCIATION_CONFIGURATION_GET							
Local Grouping identifier							

Local Grouping identifier (8 bits)

This group identifier is as explained in the Association Command Class used to specify the grouping identifier on the local node.

CONFIDENTIAL

4.81.3 Remote Association Configuration Report Command

The Remote Association Configuration Report Command returns the remote node ID and the grouping identifier. The Remote Association Configuration Report can be send requested by the Remote Association Configuration Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_REMOTE_ASSOCIATION							
Command = REMOTE_ASSOCIATION_CONFIGURATION_REPORT							
Local Grouping identifier							
Remote NodeId							
Remote Grouping identifier							

Local Grouping identifier (8 bits)

This group identifier used to specify the grouping identifier on the local node

Remote NodeId (8 bits)

This NodeId used to specify the remote node that the Remote Association Activate Command is sent to. If no link is established between the Local Grouping Identifier and a Remote Grouping Identifier, the Remote NodeId will return zero (0x0)

Remote Grouping identifier (8 bits)

This Remote grouping identifier used to specify the grouping identifier on the remote node that SHOULD be activated.

CONFIDENTIAL

4.82   Scene Activation Command Class, version 1

The Scene Activation Command Class used for the actual scene launching in a number of devices e.g. a another scene-controlling unit, in a multilevel switch, in a binary switch etc. This command class requires an initial configuration of the scenes to be launched by the Scene Actuator Configuration Set or Scene Controller Configuration Set Command depending on device used.

The advantage of this approach is that since it is a common identifier that is sent out, the multicast frame type can be used, which MAY eliminate potential popping effect which could be the result if individual set-level Commands were send out to a large number of nodes distributed over a vast area. The multicast MUST still be followed by individual singlecasts to ensure all the nodes addressed got the message.

4.82.1   Scene Activation Set Command

The Scene Activation Set Command used to activate the setting associated to the scene ID. The Scene Activation Set Command is sent as a multicast to assure all nodes within direct range responds immediately. After the multicast follows a sequence of single casts to each device to ensure all devices received the scene ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_ACTIVATION							
Command = SCENE_ACTIVATION_SET							
Scene ID							
Dimming Duration							

Scene ID (8 bits)

Scene ID (1…255) to be activated in the device.

Dimming Duration (8 bits)

The Dimming Duration can either use a pre-configured value, it can be instantly, or it can be a duration that is communicated as part of the Scene Activation Set Command. Only the Multilevel Scene Switch specific device classes interpret this field. The table below shows how to obtain the wanted functionality:

Dimming Duration	Description
0x00	Instantly
0x01-0x7F	Obtain dimming durations from 1 second (0x01) to 127 seconds (0x7F) in 1-second resolution
0x80-0xFE	Specify dimming durations from 1 minute (0x80) to 127 minutes (0xFE) in 1-minute resolution.
0xFF	Specify dimming duration configured by the Scene Actuator Configuration Set and Scene Controller Configuration Set Command depending on device used.

4.83   Scene Actuator Configuration Command Class, version 1

The Scene Actuator Configuration Command Class used to configure scenes in a scene device e.g. a multilevel scene switch, binary scene switch etc. A scene device MUST always support 255 scene IDs.

4.83.1   Scene Actuator Configuration Set Command

The Scene Actuator Configuration Set Command used to associate the specified scene ID to the defined settings.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_ACTUATOR_CONF							
Command = SCENE_ACTUATOR_CONF_SET							
Scene ID							
Dimming Duration							
Over-ride	Reserved						
Level							

Scene ID (8 bits)

Scene ID (1...255) to be associated with the current settings.

Dimming Duration (8 bits)

Dimming Duration specify how long time it MUST take to reach the wanted level associated to the Scene ID. Dimming always start from current level. So the dimming duration specified is the same independent of the number of levels to be changed. Only the Multilevel Scene Switch specific device classes interpret this field. The table below shows how to obtain the wanted functionality:

Dimming Duration	Description
0x00	Specify Instantly
0x01-0x7F	Specify dimming durations from 1 second (0x01) to 127 seconds (0x7F) in 1-second resolution
0x80-0xFE	Specify dimming durations from 1 minute (0x80) to 127 minutes (0xFE) in 1-minute resolution.
0xFF	Specify factory default dimming duration.

Override (1 bit)

If the Override bit is set to 0 then the current settings in the device is associated with the Scene ID. If the Override bit is set to 1 then the Level value in the Command is associated to the Scene ID.

Reserved (7 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Level (8 bits)

The value specified MUST correspond to the format the device uses when receiving Basic Set Commands.

4.83.2 Scene Actuator Configuration Get Command

The Scene Actuator Configuration Get Command is used to request the settings for a given scene identifier or for the currently active scene settings.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_ACTUATOR_CONF							
Command = SCENE_ACTUATOR_CONF_GET							
Scene ID							

Scene ID (8 bits)

Scene ID (1...255) to request. If scene ID is equal to 0 then current active scene is requested.

CONFIDENTIAL

4.83.3 Scene Actuator Configuration Report Command

The Scene Actuator Configuration Report Command used to report the locally stored configuration for a given scene identifier in the device requested by the Scene Actuator Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_ACTUATOR_CONF							
Command = SCENE_ACTUATOR_CONF_REPORT							
Scene ID							
Level							
Dimming Duration							

Scene ID (8 bits)

The scene ID (1...255) indicates scene settings being returned. If scene ID is equal to 0 it indicate that no scene is currently active in the device.

Level (8 bits)

The value reported by the device MUST correspond to the format the device uses to respond to Basic Get Commands.

Dimming Duration (8 bits)

Dimming Duration specify how long time it MUST take to reach the wanted level associated to the Scene ID. Only the Multilevel Scene Switch specific device classes interpret this field. The table below shows how the different dimming durations are reported:

Dimming Duration	Description
0x00	Instantly
0x01-0x7F	Dimming durations from 1 second (0x01) to 127 seconds (0x7F) in 1-second resolution
0x80-0xFE	Specify dimming durations from 1 minute (0x80) to 127 minutes (0xFE) in 1-minute resolution.

4.84   Scene Controller Configuration Command Class, version 1

The Scene Controller Configuration Command Class used to configure scenes controlled from a scene controlling device by some kind of physical activation. A scene device MUST always support 255 scene IDs.

4.84.1   Scene Controller Configuration Set Command

The Scene Controller Configuration Set Command used to configure settings for a given physical item on the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_CONTROLLER_CONF							
Command = SCENE_CONTROLLER_CONF_SET							
Group ID							
Scene ID							
Dimming Duration							

Group ID (8 bits)

The grouping identifier is mapped into a physical item e.g. a push button on the device in question. The grouping identifier values MUST be a sequence starting from 1. The Association Supported Groupings Get Command can be used to request the number of groupings that the device supports.

Scene ID (8 bits)

Scene ID (1…255) to be associated with the grouping identifier. To disable an associated scene for the specified group ID set scene ID equal to 0.

Dimming Duration (8 bits)

Dimming Duration specify how long time it MUST take to reach the wanted level associated to the Scene ID. Dimming always start from current level. So the dimming duration specified is the same independent of the number of levels to be changed. Only the Multilevel Scene Switch specific device classes interpret this field. The table below shows how to obtain the wanted functionality:

Dimming Duration	Description
0x00	Specify Instantly
0x01-0x7F	Specify dimming durations from 1 second (0x01) to 127 seconds (0x7F) in 1-second resolution
0x80-0xFE	Specify dimming durations from 1 minute (0x80) to 127 minutes (0xFE) in 1-minute resolution.
0xFF	Specify factory default dimming duration.

4.84.2 Scene Controller Configuration Get Command

The Scene Controller Configuration Get Command is used to request the settings for a given grouping identifier or the active settings.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_CONTROLLER_CONF							
Command = SCENE_CONTROLLER_CONF_GET							
Group ID							

Group ID (8 bits)

Group ID field indicates what grouping identifier the get Command is referring to. The grouping identifier values MUST be a sequence starting from 1. A grouping identifier equal to 0 requests the currently active group and scene ID. The grouping identifier is mapped into a physical item e.g. a push button on the device in question.

CONFIDENTIAL

4.84.3 Scene Controller Configuration Report Command

The Scene Controller Configuration Report Command used to report the current settings in the device requested by the Scene Controller Configuration Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCENE_CONTROLLER_CONF							
Command = SCENE_CONTROLLER_CONF_REPORT							
Group ID							
Scene ID							
Dimming Duration							

Group ID (8 bits)

The requested or active grouping identifier.

Scene ID (8 bits)

Scene ID (1…255) setting for the specified grouping identifier. In case the scene ID is disabled then 0 is returned.

Dimming Duration (8 bits)

The Dimming Duration to be used when the specified Scene ID is launched with the Scene Activation Command Class. Dimming Duration specify how long time it MUST take to reach the wanted level associated to the Scene ID. Only the Multilevel Scene Switch specific device classes interpret this field. The table below shows how the different dimming durations are reported:

Dimming Duration	Description
0x00	Instantly
0x01-0x7F	Dimming durations from 1 second (0x01) to 127 seconds (0x7F) in 1-second resolution
0x80-0xFE	Specify dimming durations from 1 minute (0x80) to 127 minutes (0xFE) in 1-minute resolution.

4.85   Schedule Command Class, version 1

The Schedule Command Class allows devices to exchange schedules which specify when to set a new behaviour. The Schedule Command Class is a generic scheduling command class that can be used to make schedules for any device type.

The first part of the Schedule Command Class will setup the time for which the schedule is to be performed and afterwards existing command classes can be used to set the behavior of the schedule. A schedule SHOULD be understood as a delayed execution of a command. The commands used in the schedule are therefore intended to be SET commands. This could be setting a value or setting behaviors with the specified delay/set time. However other command can also be used in combination with the schedule e.g. Get Commands

All schedules will have a unique Schedule ID. A Schedule consists of a start time and duration for with the desired behavior is to be valid. The Schedule ID is used to distinguish between different schedules. This gives the option of enabling/disabling, updating or removing a schedule.

4.85.1   Schedule Supported Get Command

The Schedule Support Get Command is used to examine what a device supports with respect to; number of supported schedule ID's, supported command classes and supported override types.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_GET							

4.85.2   Schedule Supported Report Command

The Schedule Support Report *Command* is used to report the number of supported schedule ID's, supported command classes and supported override types.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SUPPORTED_REPORT							
Number of Supported Schedule ID							
Support Enable/ Disable	Fall- back Support	Start Time Support					
Number of supported CC							
Supported CC 1							
Reserved							Supported Command 1
......							
Supported CC N							
Reserved							Supported Command N

CONFIDENTIAL

Override Support	Supported Override Types

Number of Supported Schedule ID (8 bits)

The number of supported schedule ID's in a device. The number of supported schedule ID's MUST start from 1 and each schedule MUST have a unique schedule ID.

Reserved Schedule ID's:

A description of the reserved schedule ID's can be found in the *Schedule Set Command*

Schedule ID = 0xFF is used for override schedules

Schedule ID = 0xFE is used as a Fall Back schedule

Schedule ID = 0x00 is reserved and used in the *Schedule Remove Command* to remove all schedule in a device.

Start Time Support (6 bit)

The schedule start time can be specified in different ways.

> Start Now
> Start based on Hour and Minute
> Calendar start time (Year, Month, Day)
> Weekdays (Monday, Tuesday etc.)

Start Time Support is used to specify the supported start time options of the device. The different options of *start time support* are:

Bit	Description
0	Only start now is supported
1	Start Hour and Minute is supported
2	Calendar time is supported
3	Weekdays is supported
4 - 5	Reserved

The bit MUST be set to 1 if the start time is supported and 0 if the start time is not supported.

Start Now:

Devices that supports start now will start the schedule when receiving the *Schedule Set Command* and run for the specified duration. The device will only run the schedule once, but MUST still send a *Schedule Report Command* as a response to a *Schedule Get Command.*

The device setting up the schedule MUST set the following values in the *Schedule Set Command*:

> Year = 0xFF
> Month = 0x00
> Day of Month = 0x00
> Weekdays = 0x00
> Hour = 0x1F

CONFIDENTIAL

Minute = 0x3F

Start based on Hour and Minute:

If the start time is specified with Hour and Minute the device MUST start the schedule every day at the specified start time. The device setting up the schedule MUST set the following values in the *Schedule Set Command* when specifying start Hour and Minute:

Year = 0xFF
Month = 0x00
Day of Month = 0x00
Weekdays = 0x00

Calendar Start Time (Year, Month, Day):

Devices that supports calendar time to enable the start time to be specified as *Year, Month* and *Day of Month*.

- If *Year* is not specified the schedule MUST run every year at the specified *Month* and *Day of Month.*

- If *Year* and *Month* are not specified the schedule MUST run every month at the specified *Day of Month.*

The device setting up the schedule MUST set the following values in the *Schedule Set Command* when specifying start time:

Weekdays = 0x00

Weekdays (Monday, Tuesday etc.):

Devices that supports running the schedule on different weekdays. The weekdays are indicated as a bit mask in the *Schedule Set Command.*

The device setting up the schedule MUST set the following values in the *Schedule Set Command* when specifying start time:

Year = 0xFF
Month = 0x00
Day of Month = 0x00

CONFIDENTIAL

Fall Back Support (1 bit)

This bit is used to indicate if the device supports a Fall Back schedule (Schedule ID 0xFE). More information about the use of Fall Back can be found in the description of the *Schedule Set Command*.

Support Enable/Disable (1 bit)

This bit is used to indicate if the device support enabling/disabling of schedules through the *Schedule State Set Command*. If the bit is 0 the device supports enabling/disabling of schedules using the *Schedule State Set Command.* If the bit is 1 the devices does not support enabling/disabling of schedules.

Number of supported CC (8 bit)

The number of supported commands. If the value is set to 0xFF all command classes in the Node Information Frame are supported and no command classes SHALL then be listed in the *Schedule Supported Report Command*.

A command class supported in the Schedule Command Class MUST also be supported without the Schedule Command Class. This means that all command class listed in the Node Information Frame MUST be implemented as stand-alone but can OPTIONALly be supported through the Schedule Command Class.

Supported CC (8 bit)

The supported command classes.

Supported Command (2 bit)

The *supported Command* indicates the supported commands for the command class. It is possible to support only Set commands, Get commands or both. The possible values are:

Value	Description
0x00	Both Set and Get Commands are supported
0x01	Only Set Commands are supported
0x02	Only Get Commands are supported
0x03	Reserved

CONFIDENTIAL

Supported Override Types (7 bit)

The *Supported Override Types* is a bit mask filed used to indicate what the device supports as temporary override. The temporary override state is enabled by setting *Schedule ID* to 0xFF and the *Duration Type* to Override in the *Schedule Set Command* and the override types are set in the *Duration Bytes (LSB)*. More information about the use of override can be found in the description of the *Schedule Set Command*.

The below table shows the possible override types that can be used. If the override type is supported the bit in the *Supported Override Types* byte MUST be set to 1. If the override type is not supported the bit MUST be set to 0.

Bit	Name	Duration value (LSB)	Description
0	Advance	0x01	- The override schedule will run until the start of the next normal schedule.
1	Run forever	0x02	- The override schedule will run until stopped
2 - 6	Reserved	0x03 – 0x07	- Reserved

Override Support (1 bit)

This bit is used to indicate if the device supports override. The bit is set to 0 if override is not supported and 1 if override is supported. If override is not supported all *Schedule Set Commands* with *Schedule ID* 0xFF MUST be ignored.

4.85.3 Schedule Set Command

The Schedule Set Command is used to set a new schedule in a device for a specific weekday. A schedule defines when to apply the specified behaviour and the duration for the behaviour.

Overlapping schedules are allowed as long as they don't address the same functionality in the target device. This means that a schedule can be started even when a schedule is already in use. But the second schedule can't use the same function or functionality as the already running schedule. It is the device that controls the schedule that is responsible for complying with this requirement for overlapping schedules.

Allowed Overlapping Schedules: The *Schedule Command Class* is used to schedule when user codes (User Identifier) for a door lock system is active. Two user codes are configured and both MUST be active from 1 - 3 pm every day. In this case it is allowed to have overlapping schedules as the schedules don't conflict.

Not Allowed Overlapping Schedules: The *Schedule Command Class* is used to schedule the temperature *Setpoint* in a thermostat. One schedule is active from 7 – 9 am every day with a *Setpoint* of 22 deg. A second schedule is active from 8 – 11 am every day with a *Setpoint* of 20 deg. This is not allowed as the two schedules are adjusting the Setpoint at the same time.

CONFIDENTIAL

If a device receives an invalid schedule the device SHOULD discard the schedule. It is therefore RECOMMENDED that the device setting up the schedule use the *Schedule Get Command* to verify that the schedule is correctly setup. The *Schedule State Get Command* can also be used to verify that the Schedule ID is changed to active and thereby presume that the schedule is correctly setup.

At the end of a schedule the device will default back to the Fall Back schedule if supported. If the Fall Back schedule is not supported the device MUST default back to normal mode. It is up to the designer to define what normal mode is for the specific device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_SET							
Schedule ID							
User Identifier							
Start Year							
Reserved				Start Month			
Reserved			Start Day of Month				
Res.	Start Weekday						
Duration Type			Start Hour				
Reserved		Start Minute					
Duration Byte 1 MSB							
Duration Byte 2 LSB							
Reports to Follow							
Number of Cmd to Follow							
Cmd Length							
Cmd Byte 1							
……							
Cmd Byte N							
Cmd Length							
Cmd Byte 1							
……							
Cmd Byte N							

CONFIDENTIAL

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Schedule ID (8 bit)

The *Schedule ID* is a unique ID for each schedule. The device controlling the schedule MUST ask the device supporting the schedule for number of supported ID's and active ID's. The schedule ID's MUST start from 1 to the number of supported schedule ID's

Reserved ID

- Schedule ID = 0xFF is used for override schedules

- Schedule ID = 0xFE is used as a Fall Back schedule

- Schedule ID = 0x00 is used in the *Schedule Remove Command* to remove all schedule in a device.

Override:

Schedule ID = 0xFF is a reserved ID used for temporary override. When a device receives a *Schedule Set Command* with *Schedule ID* = 0xFF all normal schedules MUST be disabled and only the override schedule is active.

The device setting up the override schedule MUST verify that the receiving device support override. This is done through the *Schedule Support Report Command*. A device not supporting override MUST ignore schedules with schedule ID 0xFF.

When a device is running an override schedule all normal schedules MUST be disabled and this MUST be indicated through the *Schedule Report Command* and *Schedule State Report Command* in the *Active ID* parameter.

An override schedule can be removed by using the *Schedule Remove Command* with *Schedule ID* 0xFF. When the override schedule is removed the normal schedules is enabled. If a schedule is enabled within the defined execution time of the schedule, the schedule MUST become active. The following is an example of this:

> A schedule (*Schedule ID* = 1) is defined to run from 10 AM and duration 3 hours (1 PM). An override schedule stops at 11 AM, and the normal schedules are enabled. In this case the override schedule is removed within the execution time of *Schedule ID* = 1. Therefore *Schedule ID* = 1 MUST become active when enabled and run as expected to 1 PM.

A device MAY support pre-defined override types. This is indicated in the *Schedule Support Report Command*. The pre-defined override types specify duration and MUST therefore be used in combination with the *Duration Type* = Override in the *Schedule Set Command*. Override MAY also be used without the pre-defined durations and in this case *Duration Type* MUST be Minutes, Hours or Days.

Fall Back:

Schedule ID = 0xFE is a reserved ID used for Fall Back schedule. The device setting up the schedule MUST verify that the receiving device support Fall Back. This is done through the *Schedule Support Report Command*. A device not supporting Fall Back MUST ignore schedules with schedule ID 0xFE.

The Fall Back schedule is used to set the behavior of the device when the device is not running a schedule. The following is an example of this:

CONFIDENTIAL

A schedule (*Schedule ID* = 1) is defined to run from 10 AM and duration 3 hours (1 PM). A second schedule (*Schedule ID* = 2) is defined to run from 2 PM and duration 2 hours (4 PM). In the time between the two schedules, from 1 PM to 2 PM the device will run the Fall Back Schedule (*Schedule ID* = 0xFE).

A Fall Back schedule can be removed by using the *Schedule Remove Command* with *Schedule ID* 0xFE.

If the Fall Back schedule is not supported or not specified the device MUST Fall Back to normal mode. Normal mode is specified by the designer and will be manufacture specific.

User Identifier (8 bit)

The User Identifier is used for schedules related to a user identifier (*User Code Command Class*). If the schedule is not related to a user identifier this byte MUST be 0x00.

Note: The user Identifier MUST first be set in the device by the *User Code Command Class* before used in a schedule. Therefore the device receiving the schedule (Supporting *Schedule Command Class*) MUST have support for *User Code Command Class* in the *Node Information Frame* to support the use of *User Identifier* in the *Schedule Command Class.* But the *User Code Command Class* does not need to be supported in the *Schedule Support Report Command.*

Start Year (8 bit)

Year is used to set the year for which the schedule MUST start. Year can have a value from 0 to 99 that specify the year of the century. If year is not used the value MUST be 0xFF.

Start Month (4 bit)

Month is used to set the month for which the schedule MUST start. Month can have a value from 1 to 12 that specify the month of the year. If month is not used the value MUST be 0x00.

If year is specified not used (0xFF), the schedule MUST be used every year at the specified time.

Start Day of Month (5 bit)

Day of Month is used to set the day for which the schedule MUST start. The value can be from 1 to 31 to specify the day of the month. If Day of Month is not used the value MUST be 0x00.

If year and Month is specified not used (Year = 0xFF and Month = 0x00), the schedule MUST be used every Month at the specified time.

Start Weekday (7 bit)

Weekday is used to set a schedule for a fixed weekday. Each bit indicates a weekday. The bit MUST be 1 for the days the schedule is to be used, and 0 for the days not used.

Bit	Description
0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

CONFIDENTIAL

Note:

- If Year, Month and Day of Month are used all Weekdays MUST be 0 and no assumption MUST be made to this field!

- if Year, Month, Day of Month and Weekday all are specified not used, the schedule MUST be used every day at the specified Hour and Minute.

- A detailed description of specifying the start time can be found in Appendix B.

Duration Type (3 bit)

The duration type used for setting the duration. If *Duration Type* is Override (0x03) the duration values are specified by the *Override Types* (see values under *Schedule Support Report Command*). Duration Type Override can only be used in combination with override schedules (Schedule ID 0xFF).

Hex	Description
0x00	Minutes
0x01	Hours
0x02	Days
0x03	Override
0x04 – 0x07	Reserved

Start Hour (5 bit):

A value from 0 to 23 that specify the hour of the day. If Hour is nor used the value MUST be 0x1F

Start Minute (6 bit):

A value from 0 to 59 that specify the minute of the hour. If Minute is not used the value MUST be 0x3F

Note:

- if Hour and Minute both are 0xFF, the schedule MUST start immediately. In this case the schedule MUST only be used once.

- A detailed description of specifying the start time can be found under *Schedule Support Report Command*.

Duration (16 bit)

Duration specifies for how long the schedule is to last. The duration is specified by *Duration Type* with the first byte as the most significant byte.

If *Duration Type* is Override (0x03) the duration values are specified by the *Override Types* (see values under *Schedule Support Report Command*). The *Override Type* MUST be specified in the LSB and the MSB MUST be 0x00.

Reports to Follow (8 bit)

The *Reports to Follow* is used if the schedule is divided into more frames. The reports to follow indicate the remaining frames and MUST therefore be decremented for each frame. If e.g. two frames are needed, reports to follow MUST be 0x01 in first frame and 0x00 in second frame.

CONFIDENTIAL

If more frames are used the header bytes (schedule ID, start time and duration) MUST be the same for all frames.

Number of Cmd to Follow (8 bit)

Number of Cmd to Follow specified the number of commands included within the frame.

Note: The maximum frame length is 48 Bytes

Cmd Length (8 bit)

The *Cmd Length* is used to indicate the number of used *Cmd Byte* in the following command.

Cmd Byte 1-N (8 bit)

The command classes to be send. The commands are expected to be Set commands but not REQUIRED.

If a Get command is used the device supporting the schedule MUST send the Report command to the device setting up the schedule. In this case the device setting up the schedule MUST guarantee that the device supporting the schedule has the needed return routes to send the Report command.

Example:

A controller (Node ID 1) is used to setup a schedule in a slave (Node ID 2) with a Get command. In this case the slave (Node ID 2) will send the Report to the controller (Node ID 1). The controller MUST guarantee that the slave has assigned return routes to send to the controller (Node ID 1)
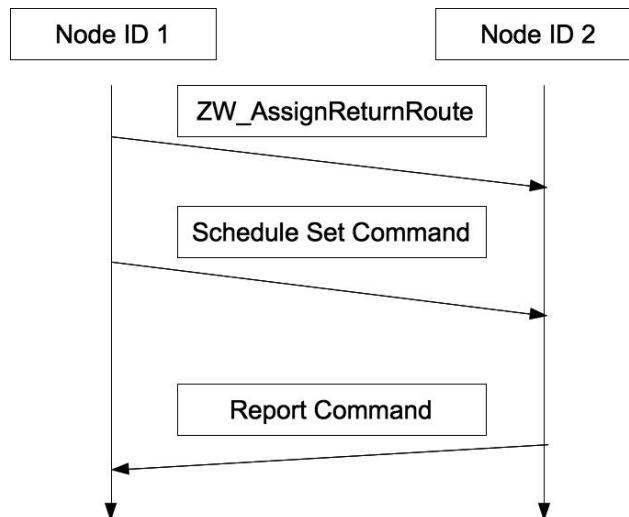
4.85.4   Schedule Get Command

The Schedule Get Command is used to request the schedule in a device for a specific schedule ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_GET							
Schedule ID							

4.85.5   Schedule Report Command

The Schedule Report Command is used to report the schedule in a device for a specific schedule ID. The *Schedule Report Command* MUST be send as a respond to receiving a Schedule Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REPORT							
Schedule ID							
User Identifier							
Start Year							
Active_ID				Start Month			
Reserved			Start Day of Month				
Res.		Start Weekday					
Duration Type			Start Hour				
Reserved			Start Minute				
Duration Byte 1							
Duration Byte 2							
Report to Follow							
Number of Cmd to Follow							
Cmd Length							
Cmd Byte 1							
Cmd Byte 2							
……							
Cmd Byte N							

The parameters are described in the Schedule set command.

The Active_ID parameter is described in the Schedule State Report Command

If a *Schedule Get Command* is received for a Free/Not Used schedule all parameters until *Number of Cmd to Follow* MUST be set to 0x00 except for the *Active_ID* parameter that MUST indicate the status. No *Cmd Length* and *Cmd Bytes* MUST be send.

CONFIDENTIAL

4.85.6   Schedule Remove Command

The Schedule Remove Command is used to remove one or all Schedule in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_REMOVE							
Schedule ID							

Schedule ID (8 bit)

The ID of the schedule that will be removed.

If the *Schedule ID* is set to 0x00 all schedules will be removed.

If a device is running an override schedule the override schedule can be Removed by using *Schedule Remove Command* with *Schedule ID* = 0xFF. This will at the same time enable all normal schedules

4.85.7   Schedule State Set Command

The Schedule State Set Command is used to enable/disable schedules in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_SET							
Schedule ID							
Schedule State							

Schedule ID (8 bits)

The ID of the schedule that is to be enabled/disabled. If the *Schedule ID* is set to 0x00 all Schedules will be affected.

Override schedules cannot be enabled/disabled but only removed using the *Schedule Remove Command*

Schedule state (8 bits)

The schedule state value is used to enable/disable schedules. If schedule state is 0x00 it means that the schedule ID MUST be disabled in the device. If schedule state is 0xFF it means that the schedule ID MUST be enabled in the device.

NB: Disabling the schedule in a device will not erase/remove the schedules.

4.85.8   Schedule State Get Command

The Schedule State Get Command is used to get the schedule states of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_GET							

4.85.9   Schedule State Report Command

The Schedule State Report Command is used to report the state of the schedules in a device. The *Schedule State Report Command* MUST be sent as a response to a *Schedule State Get Command*.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_STATE_REPORT							
Number of Supported Schedule ID							
Reports to Follow							Override
Active_ID 2				Active_ID 1			
Active_ID N				Active_ID 3			

Number of Supported Schedule ID (8 bits)

The number of supported schedule ID's. The number of supported schedule ID's in a device MUST start from 1 and each schedule MUST have a unique schedule ID.

Override (1 bit)

Indicates if the device is running an override, *Schedule ID* 0xFF. If the device is running an override this bit MUST be 1, and 0 if not.

Reports to Follow (7 bit)

Reports to Follow indicated the number of reports/frames to follow. This can be used for devices supporting many *Schedule ID's*.

Active_ID (4 bit)

Active_ID is used to indicate which schedule ID's are used. Four Bits are used for each Schedule ID to indicate the status. This means that Bit 0 to Bit 3 in Active_ID Byte 1 represents Schedule ID = 1. The following indications are valid:

Hex	Description	Description
0x00	Free	- The Schedule ID is free to be used for new schedules
0x01	Free Override	- The Schedule ID is free but the device is running an override
0x02	Active Not Running	- The schedule ID is used but outside the scheduled time
0x03	Active Running	- The schedule ID is used an currently in the scheduled time
0x04	Active But Disabled	- The schedule ID is used but disabled
0x05	Active Override Running	- The schedule ID is used and in the schedule time. But the device is running an override
0x06	Active Override Not Running	- The schedule ID is used but outside the schedule time. Also the device is running an override
0x07	Active Override Disabled	- The schedule ID is used but disabled. Also the device is running an override
0x08 – 0x0F	Reserved	

Note:

A new schedule is by default enabled

When changing a disabled schedule using *Schedule Set Command* the schedule will remain disabled

If a disabled schedule is overridden, when the override finishes the schedule MUST remain disabled.

CONFIDENTIAL

4.86   Schedule Entry Lock Command Class, version 1

> It´s RECOMMENDED using Schedule Command Class instead for new devices.

The Schedule Entry Lock Command Class provides Z-Wave devices the capability to exchange scheduling information.   The Schedule Entry Lock Type Commands are for controlling the schedules of a Entry Lock using schedule based user code Ids. The Entry Lock supports two types of schedules for each user ID supported in the device.   The two schedule types are a time-fenced weekly schedule and a time-fenced one-time range schedule. When these schedules are configured and enabled, it allows the specified user ID's code to be active during the time intervals configured in the scheduling slots.

The Week Day schedule is a day-to-day schedule that will repeat weekly for the enabled user ID. A single schedule slot cannot span days.

Example: A homeowner has a Secure Keypad Door Lock and a dog that needs walking three times a week. The dog walker can be given access to the house using this schedule. The homeowner would give the dog walker a keypad code that would be active M, W, F from 1pm – 2pm.

The Year Day schedule is an extended schedule that allows two points in time to be specified that is beyond a daily schedule. A particular slot can span weeks, months or years. Once the end point is reached that schedule slot is no longer valid because it is out of range.

Example: A homeowner is going away on vacation for two weeks. The homeowner could give the neighbor a keypad code to the neighbor that would be active from April 2nd, 2008 to April 16th 2008. The code would be invalid after April 16th 2008.

CONFIDENTIAL

4.86.1   Schedule Entry Lock Enable Set Command

The Schedule Entry Lock Enable Set Command enables or disables schedules for a specified user code ID. It affects only the schedules associated with the specific user ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_ENABLE_SET							
User Identifier							
Enabled							

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored.

Enabled (8 bits)

Enabled	Description
0	Schedule for the user identified is disabled
1	Schedule for the user identified is enabled

4.86.2   Schedule Entry Lock Enable All Set Command

The Schedule Entry Lock Enable All Set Command enables or disables all schedules for type Entry Lock.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_ENABLE_ALL_SET							
Enabled							

Enabled (8 bits)

See description in Schedule Entry Lock Enable Set Command.

CONFIDENTIAL

4.86.3   Schedule Entry Lock Supported Get Command

The Schedule Entry Lock Supported Get Command is used to request the number of schedule slots each type of schedule the device supports for every user.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_TYPE_SUPPORTED_GET							

4.86.4   Schedule Entry Lock Supported Report Command

The Schedule Entry Lock Supported Report Command is used to report the number of supported schedule slots an Entry Lock schedule device supports for each user in the system. It lists how many schedule slots there are for Week Days type and how many slots for the Year Day type.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_TYPE SUPPORTED_REPORT							
Number of Slots Week Day							
Number of Slots Year Day							

Number of Slots Week Day (8 bits)

A number from 0 – 255 that represents how many different schedule slots are supported each week for every user in the system for type Week Day.

Number of Slots Year Day (8 bits)

A number from 0 – 255 that represents how many different schedule slots are supported for every user in the system for type Year Day.

CONFIDENTIAL

4.86.5   Schedule Entry Lock Week Day Schedule Set Command

The Schedule Entry Lock Week Day Schedule Set Command set or erase a weekday schedule for a identified user who already has valid user access code.

When setting, the week day schedule is automatically enabled and the identified user if it is not already. The start parameters of the time fence needs to occur prior to the stop parameters. When erasing the schedule slot ID, the user code ID will continue to use week day type scheduling.

Note: Each user can only use one type of scheduling at a time.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_WEEK_DAY_SET							
Set Action							
User Identifier							
Schedule Slot ID							
Day of Week							
Start Hour							
Start Minute							
Stop Hour							
Stop Minute							

Set Action (8 bits)

Set Action	Description
0	Erase the schedule slot
1	Modify the schedule slot for the identified user

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored

Schedule Slot ID (8 bits)

A value from 1 to *Number of Slots Week Day Supported.*

Day of Week (8 bits)

A value from 0 to 6 where 0 is Sunday.

Start Hour (8 bits)

A value from 0 to 23 representing the starting hour of the time fence.

CONFIDENTIAL

Start Minute (8 bits)

A value from 0 to 59 representing the starting minute of the time fence.

Stop Hour (8 bits)

A value from 0 to 23 representing the stop hour of the time fence.

Stop Minute (8 bits)

A value from 0 to 59 representing the stop minute of the time fence

4.86.6   Schedule Entry Lock Week Days Schedule Get Command

The Schedule Entry Lock Week Days Schedule Get Command get a week day schedule slot for a identified user and specified schedule slot ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_WEEK_DAY_GET							
User Identifier							
Schedule Slot ID							

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored

Schedule Slot ID (8 bits)

A value from 1 to *Number of Slots Week Day Supported*.

4.86.7   Schedule Entry Lock Week Day Schedule Report Command

The Schedule Entry Lock Week Day Schedule Report Command returns week day schedule report for the requested schedule slot ID for identified user.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_WEEK_DAY_REPORT							
User Identifier							
Schedule Slot ID							
Day of Week							
Start Hour							
Start Minute							
Stop Hour							
Stop Minute							

Refer to the description under the Schedule Set Week Day Schedule.

Note: If a requested schedule slot is erased/empty, then the time fields SHOULD be set to 0xFF.

CONFIDENTIAL

4.86.8   Schedule Entry Lock Year Day Schedule Set Command

The Schedule Entry Lock Year Day Schedule Set Command set or erase a schedule slot for a identified user who already has valid user access code. The year day schedule represents two days, any time apart, where the specified user ID's code is valid. When setting the schedule slot, the start parameters of the time fence needs to occur prior to the stop parameters and the year day schedule is automatically enabled for the identified user. When erasing, the user code does not change from year day scheduling.

Note: Each user can only use one type of scheduling at a time.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_YEAR_DAY_SET							
Set Action							
User Identifier							
Schedule Slot ID							
Start Year							
Start Month							
Start Day							
Start Hour							
Start Minute							
Stop Year							
Stop Month							
Stop Day							
Stop Hour							
Stop Minute							

Set Action (8 bits)

Set Action	Description
0	Erase the schedule slot
1	Modify the schedule slot for the identified user

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored

Schedule Slot ID (8 bits)

A value from 1 to *Number of Slots Year Day Supported*.

Start Year (8 bits)

A value from 0 to 99 that represents the 2 year in the century.

Start Month (8 bits)

A value from 1 to 12 that represents the month in a year.

Start Day (8 bits)

A value from 1 to 31 that represents the date of the month.

Start Hour (8 bits)

A value from 0 to 23 representing the starting hour of the time fence.

Start Minute (8 bits)

A value from 0 to 59 representing the starting minute of the time fence.

Stop Year (8 bits)

A value from 0 to 99 that represents the 2 year in the century.

Stop Month (8 bits)

A value from 1 to 12 that represents the month in a year.

Stop Day (8 bits)

A value from 1 to 31 that represents the date of the month.

Stop Hour (8 bits)

A value from 0 to 23 representing the stop hour of the time fence.

Stop Minute (8 bits)

A value from 0 to 59 representing the stop minute of the time fence

4.86.9   Schedule Entry Lock Year Day Schedule Get Command

The Schedule Entry Lock Year Day Schedule Get Command get a year/day schedule slot for an identified user and specified schedule slot ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_YEAR_DAY_GET							
User Identifier							
Schedule Slot ID							

CONFIDENTIAL

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored

Schedule Slot ID (8 bits)

A value from 1 to *Number of Slots Year Day Supported*.

4.86.10 Schedule Entry Lock Year Day Schedule Report Command

The Schedule Entry Lock Year Day Schedule Report Command returns year/day schedule report for the requested schedule slot ID for the identified user.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_YEAR_DAY_REPORT							
User Identifier							
Schedule Slot ID							
Start Year							
Start Month							
Start Day							
Start Hour							
Start Minute							
Stop Year							
Stop Month							
Stop Day							
Stop Hour							
Stop Minute							

Refer to the description under Schedule Set Year Day Schedule command.

Note: If a requested schedule slot is erased/empty then the time fields SHOULD be set to 0xFF.

CONFIDENTIAL

4.87   Schedule Entry Lock Command Class, version 2

The Schedule Entry Lock Command Class provides Z-Wave devices the capability to exchange scheduling information. The Schedule Entry Lock Type Commands are for controlling the schedules of an Entry Lock using schedule based user code Ids. The Entry Lock supports two types of schedules for each user ID supported in the device. The two schedule types are a time-fenced weekly schedule and a time-fenced one-time range schedule. When these schedules are configured and enabled, it allows the specified user ID's code to be active during the time intervals configured in the scheduling slots.

In Version 2 local time is used instead of UTC time, and Time Offset commands are added.

The commands not mentioned here remain the same as in version 1.

4.87.1   Schedule Entry Lock Time Offset Get Command

The Schedule Entry Lock Time Offset Get Command is used to request time zone offset and daylight savings parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_TIME_OFFSET_GET							

4.87.2   Schedule Entry Lock Time Offset Set Command

The Schedule Entry Time Offset Set Command is used to set the current local TZO and DST offsets into an Entry Lock Device. Any schedules that are already in the device before or after issuing the Schedule Entry Time Offset Set command are now assumed to be programmed in the Local time set by this command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_TIME_OFFSET_SET							
Sign TZO	Hour TZO						
Minute TZO							
Sign Offset DST	Minute Offset DST						

Sign TZO (1 bit)

Plus (0) or minus (1) sign to indicate a positive or negative offset from UTC.

Hour TZO (7 bits)

Specify the number of hours that the originating time zone deviates from UTC. Refer to the DST field regarding daylight savings handling.

CONFIDENTIAL

Minute TZO (7 bits)

Specify the number of minutes that the originating time zone deviates UTC. Refer to the DST field regarding daylight savings handling.

Sign Offset DST (1 bit)

Plus (0) or minus (1) sign to indicate a positive or negative offset from UTC.

Minute Offset DST (7 bits)

Specify the number of complete minutes the time MUST be adjusted when daylight savings mode is enabled.

4.87.3 Schedule Entry Lock Time Offset Report Command

The Schedule Entry Lock Time Offset Report Command returns time zone offset and daylight savings parameters and can be requested by the Schedule Entry Lock Time Offset Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_TIME_OFFSET_REPORT							
Sign TZO	Hour TZO						
Minute TZO							
Sign Offset DST	Minute Offset DST						

Refer to description under the Schedule Entry Lock Time Offset Set command.

CONFIDENTIAL

4.88   Schedule Entry Lock Command Class, Version 3

The Schedule Entry Lock Command Class provides a scheduling type along side the existing types Week Day and Year Day. The new type is similar to Week Day functionality but provides a simpler implementation to repeat a time slot daily (selected days) and repeat those days weekly. The commands not mentioned here remain the same as in Version 2.

4.88.1   Schedule Entry Type Supported Report Command

The Schedule Entry Type Supported Report Command is used to report the number of supported schedule slots an Entry Lock schedule device supports for each user in the system. It lists how many schedule slots there are for Week Day, Year Day, and Daily Repeating types.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE							
Command = SCHEDULE_ENTRY_TYPE_SUPPORTED_REPORT							
Number of Slots Week Day							
Number of Slots Year Day							
Number of Slots Daily Repeating							

Number of Slots Week Day (8 bits)

A number from 0 – 255 that represent how many different schedule slots are supported each week for every user in the system for type Week Day.

Number of Slots Year Day (8 bits)

A number from 0 – 255 that represent how many different schedule slots are supported for every user in the system for type Year Day.

Number of Slots Daily Repeating (8 bits)

A number from 0 – 255 that represent how many different schedule slots are supported for every user in the system for type Daily Repeating Day.

4.88.2 Schedule Entry Lock Daily Repeating Set Command

The Control device uses the Schedule Entry Lock Daily Repeating Set Command to set or erase a daily repeating schedule for an identified user who already has valid user access code.

When setting; the daily repeating schedule is automatically enabled for the identified user if it is not already. The start parameters of the time fence needs to occur prior to the stop parameters. When erasing the schedule slot ID, the user code ID will continue to use daily repeating type scheduling.

Note: Each user can only use one type of scheduling at a time.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_DAILY_REPEATING_SET							
Set Action							
User Identifier							
Schedule Slot ID							
Week Day Bitmask							
Start Hour							
Start Minute							
Duration Hour							
Duration Minute							

Set Action (8 bits)

Set Action	Description
0	Erase the schedule slot
1	Modify the schedule slot for the identified user

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored

Schedule Slot ID (8 bits)

A value from 1 to *Number of Slots Daily Repeating Supported*.

Week Day Bitmask (8 bits)

A bitmask of the days of the week for this schedule entry is active.

CONFIDENTIAL

7	6	5	4	3	2	1	0
Res	Sat	Fri	Thr	Wed	Tue	Mon	Sun

Start Hour (8 bits)

A value from 0 to 23 representing the starting hour of the time fence.

Start Minute (8 bits)

A value from 0 to 59 representing the starting minute of the time fence.

Duration Hour (8 bits)

A value from 0 to 23 representing how many hours the time fence will last. Duration hour will be maxed at the documented capability of the specific device since this scheduling type is memory conscious.

Duration Minute (8 bits)

A value from 0 to 59 representing how many minutes the time fence will last past the Duration Hour field.

4.88.3   Schedule Entry Lock Daily Repeating Get Command

The Control device uses the Schedule Entry Lock Daily Repeating Get Command to get a daily repeating schedule slot for a identified user and specified schedule slot ID.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_DAILY_REPEATING_GET							
User Identifier							
Schedule Slot ID							

User Identifier (8 bits)

The User Identifier is used to recognize the user identity. A valid User Identifier MUST be a value starting from 1 to the maximum number of users supported by the device; refer to the User Code Command Class. If the user identifier is out of range, the command will be ignored

Schedule Slot ID (8 bits)

A value from 1 to *Number of Slots Daily Repeating Supported*.

CONFIDENTIAL

4.88.4   Schedule Entry Lock Daily Repeating Report

The Schedule Entry Lock Daily Repeating Report Command returned for the requested schedule slot ID for identified user.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCHEDULE_ENTRY_LOCK							
Command = SCHEDULE_ENTRY_LOCK_DAILY_REPEATING_REPORT							
User Identifier							
Schedule Slot ID							
Week Day Bitmask							
Start Hour							
Start Minute							
Duration Hour							
Duration Minute							

Refer to the description under the Schedule Set Daily Repeating Schedule.

CONFIDENTIAL

4.89 Screen Attributes Command Class, version 1

This Screen Attribute Command Class used to retrieve screen attributes from the device hosting the screen. This allows another device to send data formatted according to the screen attributes to the device hosting the screen. The screen can be located on any device in the Z-Wave network.

4.89.1 Screen Attributes Get Command

The Screen Attributes Get Command is used to request the screen attributes.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_ATTRIBUTES							
Command = SCREEN_ATTRIBUTES_GET							

CONFIDENTIAL

4.89.2   Screen Attributes Report Command

The Screen Attributes Report Command can be send unsolicited or requested by the Screen Attributes Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_ATTRIBUTES							
Command = SCREEN_ATTRIBUTES_REPORT							
Reserved				Number of Lines			
Number of Characters per Line							
Size of Line Buffer							
Numerical Presentation of a Character							

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Number of Lines (5 bits)

Number of lines the screen supports (1..16).

Number of Characters per Lines (8 bits)

Number of characters the screen supports on each line (1..255).

Size of Line Buffer (8 bits)

Number of characters the line buffer supports for each line (1..255). Size of line buffer will always be equal or larger than the number of visual characters per line. The text will typically scroll in case it is larger than the number of visual characters.

Numerical Presentation of a Character (8 bits)

The screen supports the following numerical presentations of a character:

Bit Map	Description
Bit 0	Supports ASCII codes if the bit is 1 and the opposite if 0. See Appendix B (values 128-255 are ignored)
Bit 1	Supports ASCII codes and Extended ASCII codes if the bit is 1 and the opposite if 0. See Appendix B
Bit 2	Supports Unicode UTF-16 if the bit is 1 and the opposite if 0.
Bit 3	Supports ASCII codes and Player codes, see Appendix B (undefined values are ignored)

This list MAY evolve in the future.

4.90   Screen Attributes Command Class, version 2

This Screen Attribute Command Class is enhanced with a parameter specifying Screen Timeout.

The Commands not mentioned here will remain the same as in version 1.

4.90.1   Screen Attributes Report Command

The Screen Attributes Report Command can be send unsolicited or requested by the Screen Attributes Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_ATTRIBUTES							
Command = SCREEN_ATTRIBUTES_REPORT							
Reserved		Escape Sequence	Number of Lines				
Number of Characters per Line							
Size of Line Buffer							
Numerical Presentation of a Character							
Screen Timeout							

Reserved (2 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Escape Sequence(1 bit)

If set to 0 escape sequences are not supported by the device. If set to true then escape sequences are supported by the device.

Number of Lines (5 bits)

Number of lines the screen supports (1..16).

Number of Characters per Lines (8 bits)

Number of characters the screen supports on each line (1..255).

Size of Line Buffer (8 bits)

Number of characters the line buffer supports for each line (1..255). Size of line buffer will always be equal or larger than the number of visual characters per line. The text will typically scroll in case it is larger than the number of visual characters.

CONFIDENTIAL

Numerical Presentation of a Character (8 bits)

The screen supports the following numerical presentations of a character:

Bit Map	Description
Bit 0	Supports ASCII codes if the bit is 1 and the opposite if 0. See Appendix B (values 128-255 are ignored)
Bit 1	Supports ASCII codes and Extended ASCII codes if the bit is 1 and the opposite if 0. See Appendix B
Bit 2	Supports Unicode UTF-16 if the bit is 1 and the opposite if 0.
Bit 3	Supports ASCII codes and Player codes, see Appendix B (undefined values are ignored)

This list MAY evolve in the future.

Screen Timeout (8 bits)

If Screen Timeout is set to 0 display is always on. If set to larger than 0, defines the display timeout in seconds.

CONFIDENTIAL

4.91   Screen Meta Data Command Class, version 1

The Screen Meta Data Command Class used to streaming data containing user related information to a screen located on a device in a Z-Wave network. The screen can request single or multiple data packets. The device having the data containing user related information to the screen can also initiate the data streaming.

The API call ZW_SendDataMeta MUST be used when streaming data to ensure that this traffic don't prevent control data from getting through in the network, especially important for 9.6kbps nodes because they can't detect 40kbps RF communication. Refer to [1] regarding a detailed description of the API call ZW_SendDataMeta.

4.91.1   Screen Meta Data Get Command

The Screen Meta Data Get Command is used to request the Screen Meta Data Report Command. The Screen Meta Data Get Command used as handshake to avoid buffer overflow in the receiving device. The Screen Meta Data Get Command will OPTIONALly be able to request multiple Screen Meta Data Report Commands to improve the effective bandwidth.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_MD							
Command = SCREEN_MD_GET							
Number of Reports							
Node ID							

Number of Reports (8 bits)

Number of Screen Meta Data Report Commands to be received without requesting each Screen Meta Data Report Command (1..255). Be aware of overflow when requesting multiple reports.

Node ID (8 bits)

The Node ID (1..232) specifies the device to receive the requested reports. In case node ID is equal to 0x00 then the information is requested by the source node ID of the Screen Meta Data Get Command.

CONFIDENTIAL

4.91.2 Screen Meta Data Report Command

The Screen Meta Data Report Command used to send data to the device hosting the screen. The Screen Meta Data Report Command can be send unsolicited or requested by the Screen Meta Data Get Command. The size of the payload SHOULD NOT be bigger than 48 bytes because routing over 4 hops can be necessary to reach the destination. It's possible to write characters to multiple lines in the same frame.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_MD							
Command = SCREEN_MD_REPORT							
More Data	Reser-ved	Screen Settings			Char. Presentation		
Line Settings A			Clear A	Line Number A			
Character Position A							
Number of Characters A							
Character 1,A							
…							
Character N,A							
…							
…							
Line Settings B			Clear B	Line Number B			
Character Position B							
Number of Characters B							
Character 1,B							
…							
Character N,B							

More Data (1 bit)

The more data bit indicates if additional reports are expected before the whole data streaming is completed. If the more data bit is set to 1 then additional reports are expected and the opposite if 0.

CONFIDENTIAL

Reserved (1 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Screen Settings (3 bits)

The screen settings identifier can be set to the following values:

Screen Settings	Description
0	Whole screen is cleared before lines are written
1	Current content on screen is scrolled one line down
2	Current content on screen is scrolled one line up
7	Do not change the current content on the screen

This list MAY evolve in the future. Undefined values of the screen settings identifier MUST be ignored.

Char. Presentation (3 bits)

The character presentation identifier can be set to the following values:

Char. Presentation	Description
0	Using standard ASCII codes, see Appendix B (values 128-255 are ignored)
1	Using standard ASCII codes and OEM Extended ASCII codes, see Appendix B
2	Unicode UTF-16
3	Using standard ASCII codes and Player codes, see Appendix B (undefined values are ignored)

Note: Devices supporting Unicode UTF-16 characters are described by a 2 byte long decimal representation. The first byte is the most significant byte. E.g. if there is one Unicode character in the set frame the char 1 will be MSB and char 2 will be LSB of the Unicode character.

This list MAY evolve in the future. Undefined values of the character presentation identifier MUST be ignored.

CONFIDENTIAL

Line Settings (3 bits)

The line settings identifier can be set to the following values:

Line Settings	Description
0	Characters are written in selected font
1	Characters are written as highlighted
2	Characters are written using a larger font compared to line settings equal to 0
3	
4	
5	
6	
7	

Clear (1 bit)

Determine if the characters are written directly or line is cleared first.

Clear	Description
0	Characters are written directly
1	Line is cleared before characters are written

Line Number (4 bits)

The line number field indicates the line to write the characters to counting from zero (0..15).

Character Position (8 bits)

The character position field indicates where on the line to write the characters counting from zero (0..255). The character position can be larger than the display size in case the line buffer is bigger (See the Screen Attributes Report Command).

Number of Characters (8 bits)

The number of characters field indicates how many characters to be written on the screen for the specified line number, counting from 1.

Character 1 .. Character N (variable)

The character fields hold the string to output in specified character representation. Characters will be ignored in case there is no room left in the line buffer.

CONFIDENTIAL

4.92   Screen Meta Data Command Class, version 2

The Screen Meta Data Command Class is enhanced with an extended settings bit that enabled an extra byte for settings. This version includes a setting for Screen Timeout which can be specified by the Screen Attribute Command Class version 2.

The Commands not mentioned here will remain the same as in version 1.

4.92.1   Screen Meta Data Report Command

The Screen Meta Data Report Command used to transfer data to the device hosting the screen. The Screen Meta Data Report Command can be send unsolicited or requested by the Screen Meta Data Get Command. The size of the payload SHOULD NOT be bigger than 48 bytes because routing over 4 hops can be necessary to reach the destination. It is possible to write characters to multiple lines in the same frame.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SCREEN_MD							
Command = SCREEN_MD_REPORT							
More Data	Ex-tended Setup	Screen Settings			Char. Presentation		
Line Settings A			Clear A	Line Number A			
Character Position A							
Number of Characters A							
Character 1,A							
…							
Character N,A							
…							
…							
Line Settings B			Clear B	Line Number B			
Character Position B							
Number of Characters B							
Character 1,B							
…							
Character N,B							
Reserved							Screen Timeout

More Data (1 bit)

The more data bit indicates if additional reports are expected before the whole data streaming is completed. If the more data bit is set to 1 then additional reports are expected and the opposite if 0.

Extended Setup (1 bit)

If set to true, the last byte of the payload defines an extended setup.

Screen Settings (3 bits)

The screen settings identifier can be set to the following values:

Screen Settings	Description
0	Whole screen is cleared before lines are written
1	Current content on screen is scrolled one line down
2	Current content on screen is scrolled one line up
7	Do not change the current content on the screen

This list MAY evolve in the future. Undefined values of the screen settings identifier MUST be ignored.

Char. Presentation (3 bits)

The character presentation identifier can be set to the following values:

Char. Presentation	Description
0	Using standard ASCII codes, see Appendix B (values 128-255 are ignored)
1	Using standard ASCII codes and OEM Extended ASCII codes, see Appendix B
2	Unicode UTF-16
3	Using standard ASCII codes and Player codes, see Appendix B (undefined values are ignored)

Note: Devices supporting Unicode UTF-16 characters are described by a 2 byte long decimal representation. The first byte is the most significant byte. E.g. if there is one Unicode character in the set frame the char 1 will be MSB and char 2 will be LSB of the Unicode character.

This list MAY evolve in the future. Undefined values of the character presentation identifier MUST be ignored.

CONFIDENTIAL

Line Settings (3 bits)

The line settings identifier can be set to the following values:

Line Settings	Description
0	Characters are written in selected font
1	Characters are written as highlighted
2	Characters are written using a larger font compared to line settings equal to 0
3	Characters are written using a larger font (font B) & highlighted
4	Characters are written in selected font (font A), no scroll
5	Characters are written in selected font (font A) & highlighted, no scroll
6	Characters are written using a larger font (font B), no scroll
7	Characters are written using a larger font (font B) & highlighted, no scroll

For values 0-3, text will be scrolled. For values 4-7 the text will not be scrolled, and will be truncated if it is longer than the width of the display.

Clear (1 bit)

Determine if the characters are written directly or line is cleared first.

Clear	Description
0	Characters are written directly
1	Line is cleared before characters are written

Line Number (4 bits)

The line number field indicates the line to write the characters to counting from zero (0..15).

Character Position (8 bits)

The character position field indicates where on the line to write the characters counting from zero (0..255). The character position can be larger than the display size in case the line buffer is bigger (See the Screen Attributes Report Command).

Number of Characters (8 bits)

The number of characters field indicates how many characters to be written on the screen for the specified line number, counting from 1.

Character 1 .. Character N (variable)

The character fields hold the string to output in specified character representation. Characters will be ignored in case there is no room left in the line buffer. If the Escape Sequence Bit is true in the SCREEN_ATTRIBUTES_REPORT Command, the device supports advanced display features by making escape sequences in the form of an Escape char followed by a char value 0-255.

Screen Timeout (1 bit)

If the screen timeout is set to 0 the devices preset timeout SHOULD be used.

If set to 1 the device SHOULD keep the display powered. This does not affect on the RF.

Reserved (7 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

4.93 Security Command Class, version 1

The Security Command Class and an Application Security layer specification [3] create the foundation for secure application communication between nodes in a Z-Wave network. The security layer provides confidentiality, authentication and replay attack robustness through AES-128.
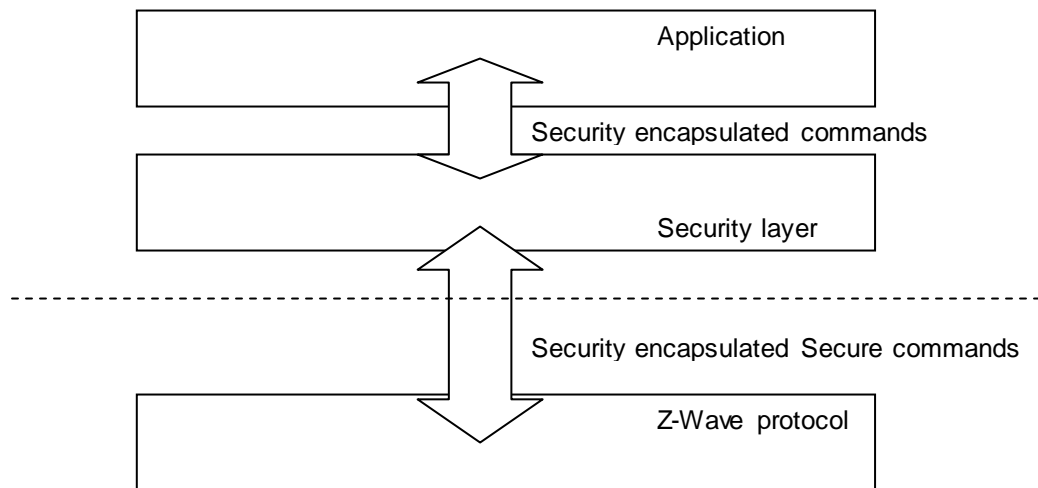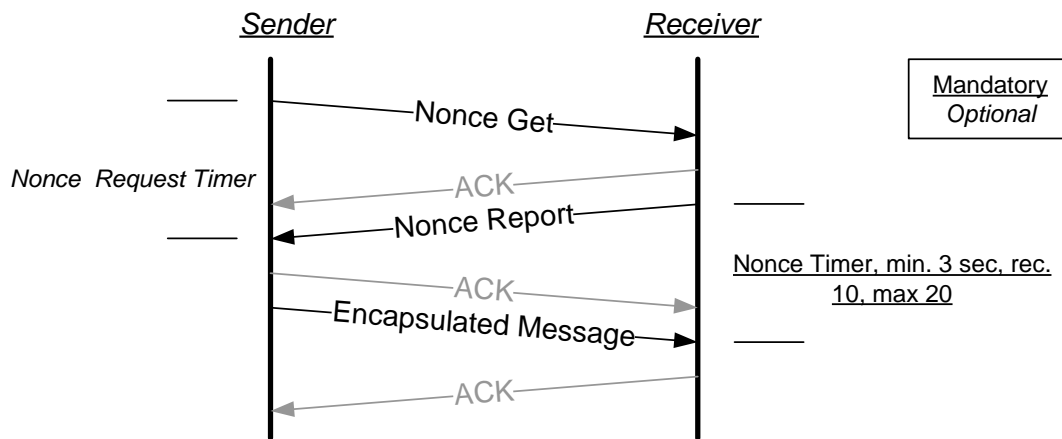
Figure 14, Protocol layers extended with security solution

The Security Command Class defines a number of commands used to facilitate handling of encrypted frames in a Z-Wave Network. The commands deal with three main areas:

- Message Encapsulation. The task of taking a plain text frame and encapsulating the frame into an encrypted Security Message.

- Command Class Handling. The task of handling what command classes are supported when communicating with a Security enabled device

- Network Key Management. The task of initial key distribution.

4.93.1   Message Encapsulation and Command Class Handling

For encapsulating messages, Z-Wave requires four commands. Before sending an encrypted frame, the sender MUST acquire a nonce (number used once) from the recipient. The sender then uses this number along with the locally generated nonce along with the network key to generate the Security Message Encapsulation Command as illustrated below.

Figure 15, Sending secure messages

This mechanism generates an overhead of three commands for each single frame that is sent encrypted (plus an acknowledge frame).

A number of timers MUST be implemented to defend against attacks.

First, an OPTIONAL but RECOMMENDED timer SHOULD be started when Nonce Get has been sent, the Nonce Report MUST then be received before this timer runs out. The length of this timer will depend on the application it is trying to protect.

The second timer is mandatory and MUST be activated after the Nonce Report has been sent. The Encapsulated Message MUST be received within the specified timeout in order to be accepted.

Note that all timers MUST be started when the frame has been sent, not when Acknowledge has been received, since an attacker could just delay the Acknowledge frame.

The Nonce Timers MUST be used in all communication that uses the mentioned commands.

Implicit ACK for Nonce Request is NOT permitted. If a Nonce Get ACK frame is lost the corresponding Nonce Report MUST be invalidated if received.

In order to optimize the performance the device MUST use streaming when transmitting multiple frames. The overhead using this option will then converge towards two (instead of three) as the number of frames increases.

Figure 16, Streaming secure messages

Notice: Due to the security overhead, the maximum command size becomes 28 bytes instead of the usual 48 bytes. Larger commands can use sequencing as described in 4.93.1.3.

4.93.1.1   Nonce Challenge Request Command

The Device uses Security Nonce Get Command to request an external nonce from the receiving node. For a description of the algorithm for generating a Z-Wave Security Nonce, see [3]. Note that a nonce will only be valid for one attempt. Nonce is thrown away when the receiver has used it for decrypting the message and a new nonce MUST be exchanged.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Security Header = SECURITY_NONCE_GET							

4.93.1.2   Nonce Challenge Response Command

The device uses the Security Nonce Report Command to return the next nonce to the requesting node at the receipt of a Security Nonce Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Security Header = SECURITY_NONCE_REPORT							
Nonce byte 1							
Nonce byte 2							
Nonce byte 3							
Nonce byte 4							
Nonce byte 5							
Nonce byte 6							
Nonce byte 7							
Nonce byte 8							

Nonce byte 1..8 (8 Bytes)

This field contains the 8 bytes external nonce used for encryption.

4.93.1.3   Security Message Encapsulation Command

The device uses the Security Message Encapsulation command to encapsulate Z-Wave commands using AES-128.

The device will also request a new external nonce from the receiver when transmitting the message Security Message Encapsulation Nonce Get. The device uses the external nonce when streaming multiple secure messages without having to call Nonce Get after receiving each message as shown in This mechanism generates an overhead of three commands for each single frame that is sent encrypted (plus an acknowledge frame).

A number of timers MUST be implemented to defend against attacks.

First, an OPTIONAL but RECOMMENDED timer SHOULD be started when Nonce Get has been sent, the Nonce Report MUST then be received before this timer runs out. The length of this timer will depend on the application it is trying to protect.

The second timer is mandatory and MUST be activated after the Nonce Report has been sent. The Encapsulated Message MUST be received within the specified timeout in order to be accepted.

Note that all timers MUST be started when the frame has been sent, not when Acknowledge has been received, since an attacker could just delay the Acknowledge frame.

The Nonce Timers MUST be used in all communication that uses the mentioned commands.

Implicit ACK for Nonce Request is NOT permitted. If a Nonce Get ACK frame is lost the corresponding Nonce Report MUST be invalidated if received.

In order to optimize the performance the device MUST use streaming when transmitting multiple frames. The overhead using this option will then converge towards two (instead of three) as the number of frames increases.

.

As illustrated in For encapsulating messages, Z-Wave requires four commands. Before sending an encrypted frame, the sender MUST acquire a nonce (number used once) from the recipient. The sender then uses this number along with the locally generated nonce along with the network key to generate the Security Message Encapsulation Command as illustrated below.
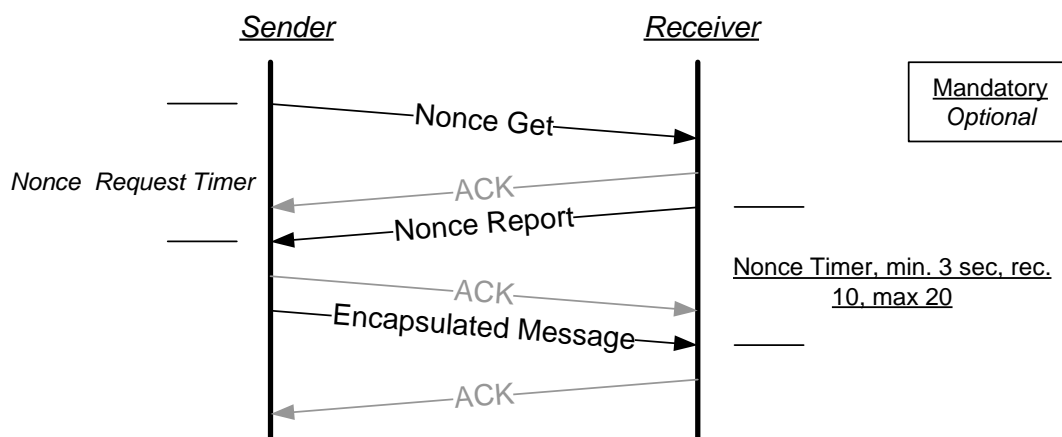
Figure 15, Sending secure messages and This mechanism generates an overhead of three commands for each single frame that is sent encrypted (plus an acknowledge frame).

A number of timers MUST be implemented to defend against attacks.

First, an OPTIONAL but RECOMMENDED timer SHOULD be started when Nonce Get has been sent, the Nonce Report MUST then be received before this timer runs out. The length of this timer will depend on the application it is trying to protect.

The second timer is mandatory and MUST be activated after the Nonce Report has been sent. The Encapsulated Message MUST be received within the specified timeout in order to be accepted.

Note that all timers MUST be started when the frame has been sent, not when Acknowledge has been received, since an attacker could just delay the Acknowledge frame.

The Nonce Timers MUST be used in all communication that uses the mentioned commands.

Implicit ACK for Nonce Request is NOT permitted. If a Nonce Get ACK frame is lost the corresponding Nonce Report MUST be invalidated if received.

In order to optimize the performance the device MUST use streaming when transmitting multiple frames. The overhead using this option will then converge towards two (instead of three) as the number of frames increases.

, the device MUST receive the Security Message Encapsulation command within 3 seconds after the creation of the nonce. The device MUST start a 3 seconds timer at creation of nonce to keep track of the validity of the nonce.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Security Header = SECURITY_MESSAGE_ENCAPSULATION (_NONCE_GET)							
Initialization Vector byte 1							
Initialization Vector byte 2							
Initialization Vector byte 3							
Initialization Vector byte 4							
Initialization Vector byte 5							
Initialization Vector byte 6							
Initialization Vector byte 7							
Initialization Vector byte 8							
Reserved			Second Frame	Sequenced	Sequence Counter		
(Command Class identifier)							
(Command identifier)							
Command byte 1							
..							
Command byte n							
Receiver's nonce Identifier							
Message Authentication Code byte 1							
Message Authentication Code byte 2							
Message Authentication Code byte 3							
Message Authentication Code byte 4							
Message Authentication Code byte 5							
Message Authentication Code byte 6							
Message Authentication Code byte 7							
Message Authentication Code byte 8							

Initialization Vector byte 1..8 (8 byte)

The initialization vector is the internal nonce generated by the sender. The payload is encrypted with the external and internal nonce concatenated together. See [3].

Reserved (2 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of the fields nor perform processing based on their content.
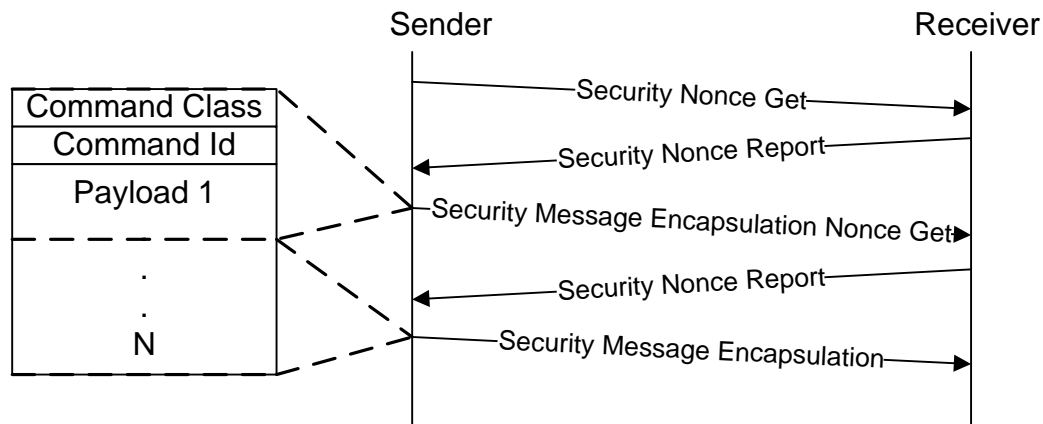
Figure 17, Frame flow for sequenced frames

Sequenced(1 bit)

If this flag is set, the device transmits the command using multiple frames due to payload space shortage. After the receiver has received all the frames, it will reassemble the command. If this flag is not set, the command is contained entirely in this frame. As shown in figure, the first frame in a sequence MUST be sent using Security Message Encapsulation Nonce Get to minimize overhead the second can be send using Security Message Encapsulation if there are no following frames to send.

Notice that device only lists Command class identifier and command identifier in the first frame.

Second Frame (1 bit)

If this flag and the Sequenced flag are set, the frame is the second out of two. If the flag is not set, and Sequenced flag is set, it is the first frame out of two. Valid combinations are:

	Sequenced 1	Sequenced 0
Second Frame 1	Second frame of two	-
Second Frame 0	First frame of two	Single Frame

Sequence Counter (4 bits)

If Sequenced flag is set, the frame is one out of two. In order to tell multiple sequences apart, they MUST be uniquely identified based on the sender node ID and the Sequence Counter. For each sequenced set of frames a node sends it MUST increment the Sequence Counter by one.

Command Class Identifier (8 bits) (Part of Encrypted Payload)

This field contains the identifier of the Command class, which the device sends to the NodeID.

Command identifier (8 bits) (Part of Encrypted Payload)

This field contains the identifier of the Command, which the device sends to the NodeID.

Command byte1 . Command byte n (Part of Encrypted Payload)

These fields contain the parameters, which the device sends to the NodeID.

Receiver's nonce Identifier (8 bits)

Identifies nonce being used. See [3].

Message Authentication Code byte 1..8 (8 byte)

Data used for authenticating the received message to prevent tampering. See [3].

CONFIDENTIAL

4.93.2   Network Key Management

The same network key is used by all secure nodes in the network. Distribution of network keys uses a temporary key to protect the key exchange. Exchange of network key happens immediately after successful inclusion of the node. It requires a secure primary/inclusion controller to include a secure node into the secure network as secure.

4.93.2.1   Network Inclusion

The first step of including a node to a secure network is using the standard Z-Wave inclusion process. If both the new node and the inclusion controller support Security command class, the controller will subsequently send the network key to the newly included node.
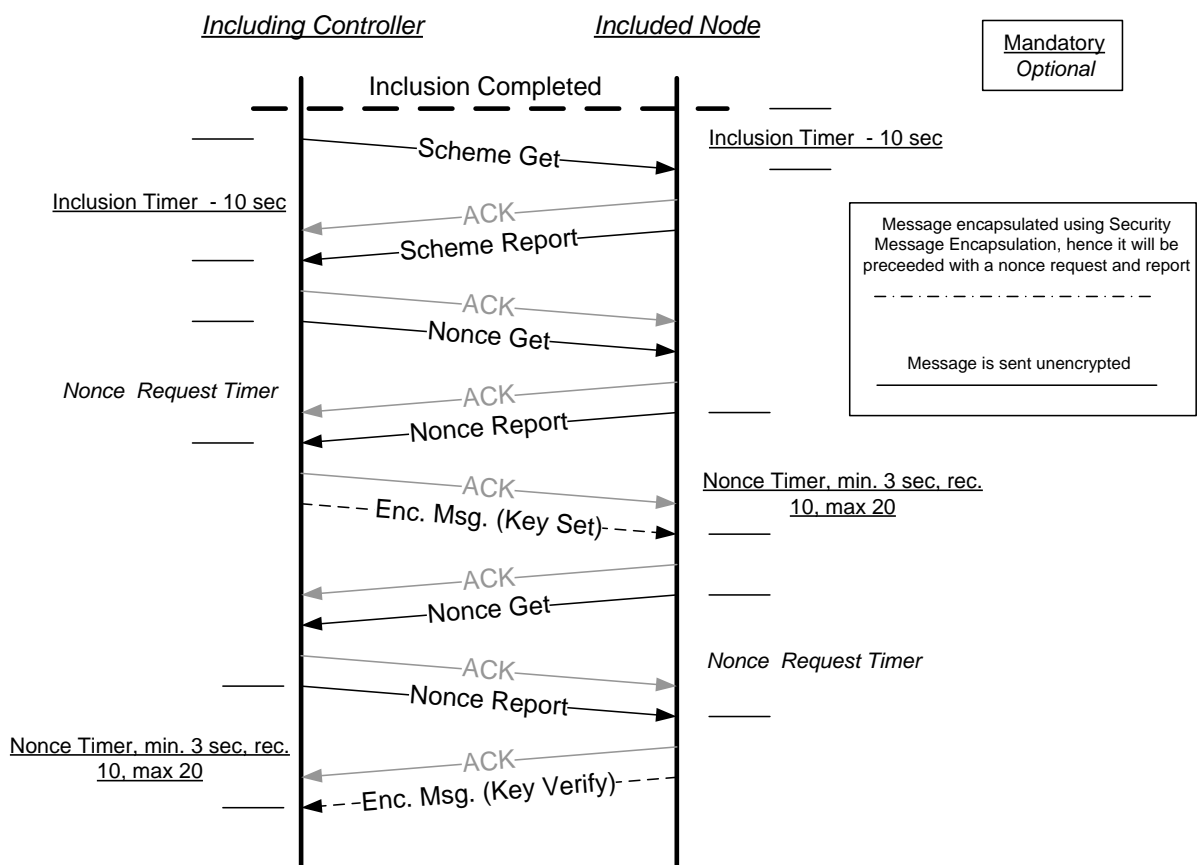
If Network Key Verify fails or the timeout is reached, Trust Center informs installer and the node must be excluded and included using same process again. If the process is not attemped again the included node will not be part of the secure network but will be present in the network as a non-secure node. The node may communicate with other non-secure nodes in the network
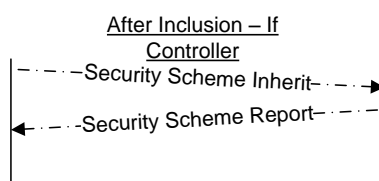
Figure 18, Inclusion into a secure network

To protect the security of a secure network it is RECOMMENDED that all controllers SHOULD require a PIN to unlock the security inclusion process and slaves SHOULD require a PIN to accept being included and excluded.

Following the inclusion of the node into the network, the controller will request the security scheme supported by the included node. Battery operated devices SHOULD stay awake for the duration of the setup of the Security Command class.

Currently one security scheme exist which is extendable at a later stage:

1. **Security 0/N:** 0x00 repeated 16 times as temporary key for encrypting the network key when it is transferred using normal power.

The validity of the key is verified in both the added node and the including controller. The node verifies the key based on the Message Authentication Code and then transmits an encrypted Network Key Verify command as response to the controller. When a device supporting the Security Command class does not manage to enter the secure network, it will function as a non-secure device. The node requires exclusion from the network before another attempt comprising of inclusion and network key exchange is possible.

For the currently available Security 0/N scheme, the same network key is used by all nodes in the network.

For the including controller to allow inclusion of a Secure device into the secure network, a common security scheme MUST be supported by both devices. When supporting multiple common schemes the highest possible scheme used. If no common schemes are supported the device cannot be included into the network.

When nodes in the secure network wish to establish a connection to a device that supports the Security Command class, they MUST send the Security Command Supported Get command to the device. Receiving no Security Command Supported Report (since the recipient does not have the key to decrypt the request), it will not be able to talk to the device securely. The same applies for the situation where a secure device does not become part of the secure network because it was included by a non-secure controller.

CONFIDENTIAL

4.93.2.1.1   Inclusion through Non-Secure Inclusion controller

It is allowed in a network comprising of a Security enabled SIS to perform secure setup after inclusion from a non-secure inclusion controller. As soon as the Security enabled SIS (hereafter SIS), receives information from the non-secure inclusion controller that a node with support for the Security command class has been included, the SIS MUST start the secure setup process of sending the network key to the newly included node as illustrated in Figure 19. At this stage the SIS acts as if it, itself had performed the inclusion and carries out all the steps REQUIRED for secure setup, included making sure the timeouts are not exceeded.

Before starting the Secure inclusion process, the SIS MUST be put into a state that allows it to carry out the secure setup for 1 node for the next 3 minutes and no longer. The SIS MUST be put in this state through a password-protected menu to avoid unintentional reveal of the network key by a fake controller.

It SHOULD be noted that performing the secure setup on behalf of a non-secure inclusion controller might add to the complexity of the actions REQUIRED by the user, and thus make it easier for a hacker to perform social engineering to circumvent the security so care MUST be taken to inform the user accordingly.
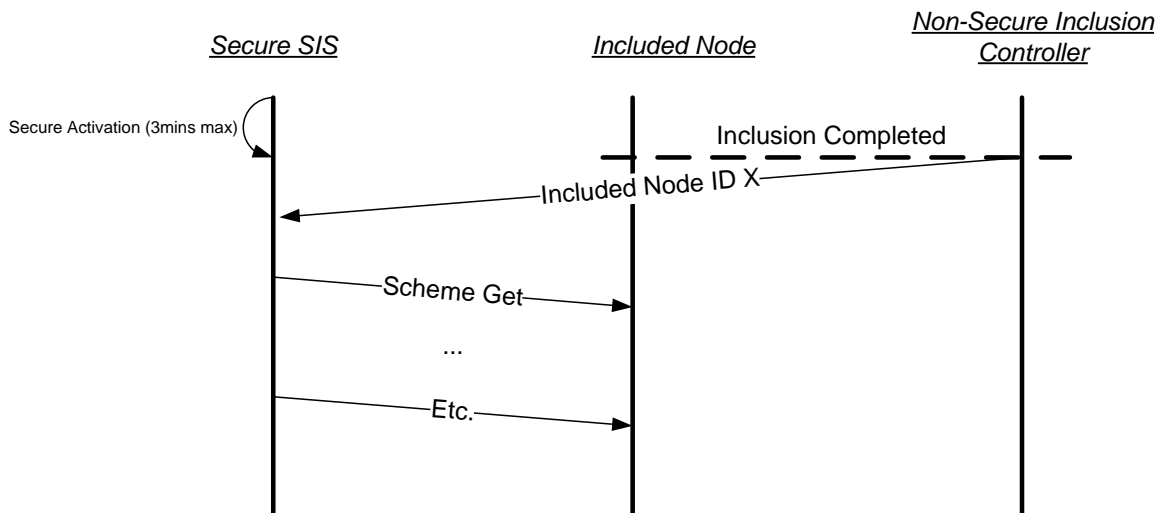
Figure 19, Secure Inclusion through Non-Secure Inclusion Controller

CONFIDENTIAL

4.93.2.1.2   Inclusion Timers

As shown in Figure 18, a number of timeout MUST be complied with. For the including controller see Figure 20.
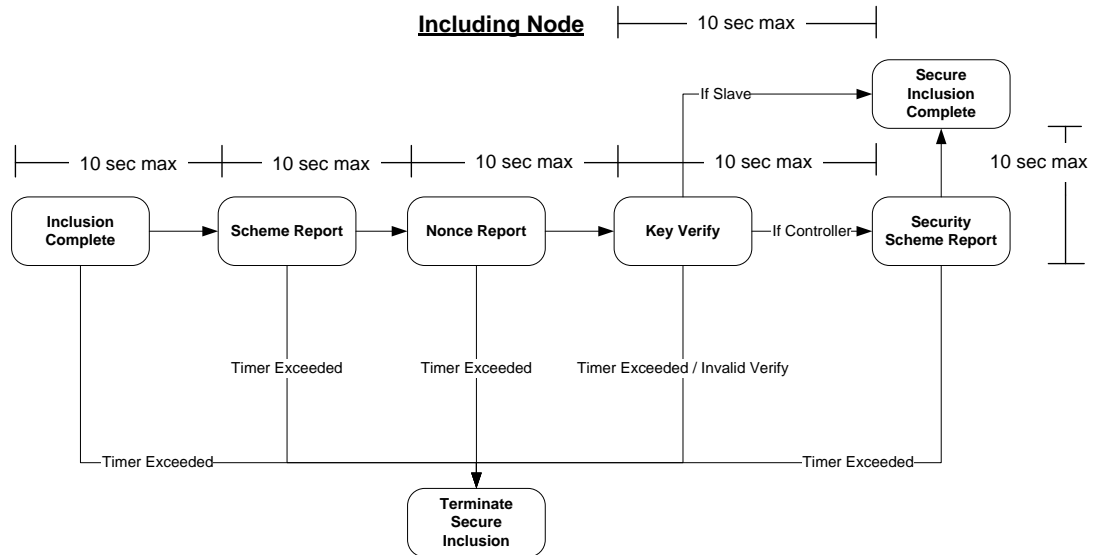
Figure 20, Timers on Including Controller

For the new included node, the timers in Figure 21 MUST be complied with.
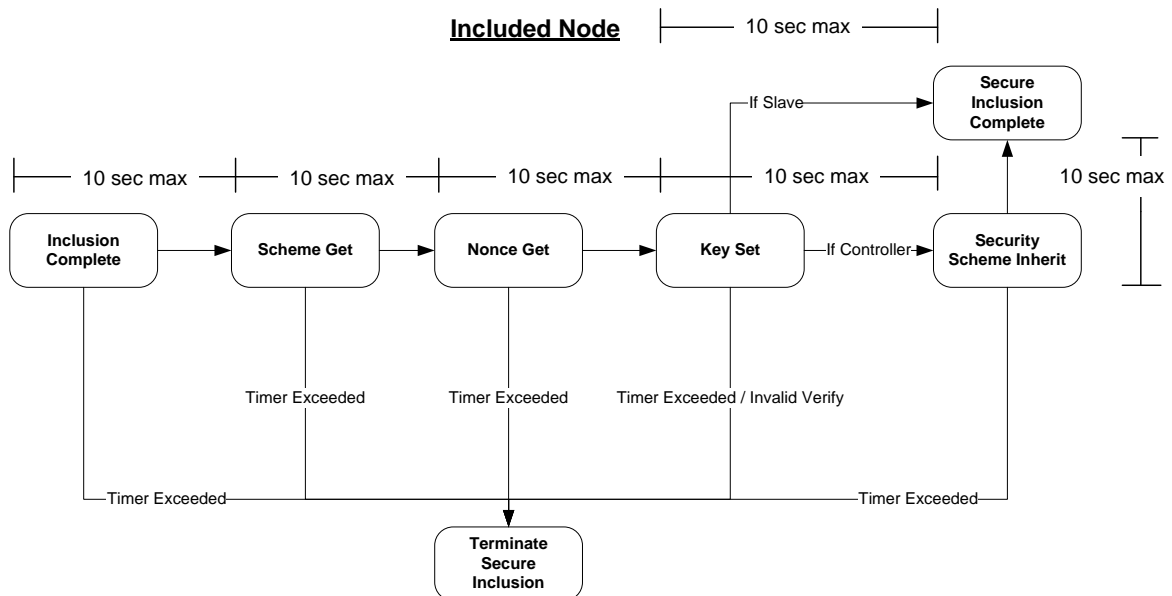
Figure 21, Timers on newly Included Node

The Network Key MUST only be sent to the new node if a Security Scheme Report command has been received by the including controller within 10 seconds after successful inclusion of the node. The controller SHOULD NOTify the user of an error condition in case of timeout because the device functions only as non-secure. In addition, the included node MUST only accept and respond to a Scheme Get it is received within 10 seconds of inclusion. When the REQUIRED frame is received within the timeout, the timeout is extended to allow the next part of the inclusion process, if that part is not reached within 10 seconds inclusion process MUST terminate.

CONFIDENTIAL

4.93.2.2   Security Scheme Get Command

The device MUST send Security Scheme Get Command immediately after the successful inclusion of a node with support for the Security Command class. The Security Scheme Get will request a report specifying the security scheme supported by the new node, and report the controllers own supported security scheme to the new node. The new node MUST then select the highest common security scheme for entering the secure network as described in the previous section.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = SECURITY_SCHEME_GET							
Supported Security Schemes							

Supported Security Schemes (8bits)

The Security Schemes which are supported by the primary/inclusion controller. At least one security scheme MUST be supported. Possible values are:

Bit	Supports
0	Security 0 using normal power = 0
1	Reserved = 0
2	Reserved = 0
3	Reserved = 0
4	Reserved = 0
5	Reserved = 0
6	Reserved = 0
7	Reserved = 0

The reserved bit fields are for future use. The implementation SHALL zero these bit fields and SHALL make no assumptions on the values of the fields nor perform processing based on their content. Bit 0 MUST always be set to 0, indicating support for Security 0. Any future scheme MUST set their appropriate bit value to 1.

CONFIDENTIAL

4.93.2.3   Security Scheme Report Command

The device MUST send Security Scheme Report Command as response to a Security Scheme Get command. The Security Scheme Report will inform the controller of the security scheme the node supports. If the only common scheme is Security 0, the device MUST send network key immediately without waiting for input, by using 16 times 0x00 as the temporary key. If no common scheme exists, the controller MUST NOTify the installer that the node cannot be included to the secure network, but will continue as a non-secure node in the network.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = SECURITY_SCHEME_REPORT							
Supported Security Schemes							

Supported Security Schemes (8bits)

See Security Scheme Get for a definition.

4.93.2.4   Network Key Set Command

The Device can use the Network Key Set Command to set the network key in a Z-Wave node. Transmission of the Network Key Set command requires existence of a common agreed security scheme. The device uses the agreed temporary key to encapsulate the Network Key Set command. The included node MAY only accept the Network Key Set command under the guidelines describes in section 4.93.2.

This command MUST only be send encapsulated by the Security Message Encapsulation command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = NETWORK_KEY_SET							
Network Key byte 1							
..							
Network Key byte N							

Network Key byte1 .. Network Key byte N

The Network key to exchange application data secure in the network.

4.93.2.5 Network Key Verify Command

When the included node has received a Network Key Set that is has successfully decrypted, verified by the MAC, it MUST send a Network Key Verify Command to the including controller. If the controller is capable of decrypting the Network Key Verify command it would indicate that the included node has successfully entered the secure network. Since there is no timeout for the Network Key Verify, the controller can send a Security Commands Supported Get command, and if no response is received, it can be concluded that it has not been included properly.

This command MUST only be send encapsulated by the Security Message Encapsulation command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = NETWORK_KEY_VERIFY							

4.93.2.6 Security Scheme Inherit Command

When a controller is included to the network, it MUST inherit the same security scheme as the including controller allowing it to become an inclusion controller. This is achieved through the Security Scheme Inherit command, which is sent when the network key has successfully been setup, as shown in Figure 18.

When including a controller into the secure network, the new controller MUST inherit any common supported security schemes. For example, if the new controller supports security scheme bit 1 and bit 4 but the including controller only supports security scheme bit 1, the new controller MUST after inclusion also only support security scheme bit 1.

This command MUST only be send encapsulated by the Security Message Encapsulation command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = SECURITY_SCHEME_INHERIT							
Supported Security Schemes							

Supported Security Schemes (8bits)

See Security Scheme Get command, for a definition.

To ensure that the included controller has inherited the correct security scheme, it MUST respond with a Security Scheme Report command as illustrated in Figure 18. If the reported security scheme does not match, the installer MUST be notified that the included controller is violating the security scheme, and the node SHOULD be excluded again as an error situation has occurred.

4.93.3   Encapsulated Command Class Handling

Since the Node Info Frame MUST only be used to communicate all the command classes that are supported non-secure, command classes supported security encapsulated MUST be reported by using the Security Commands Supported Get/Report.

To make a security enabled device compatible with non-secure applications a secure node MAY choose to report support for some command classes non-secure in the Node Info Frame, as well as in the Security Command Supported Report. Initially the node info frame MUST contain all command classes that can be supported/controlled non-secure.
If the node is included into a secure network, it MAY then choose to remove all or some command classes from the node info frame, and thus only support them securely – removing support for the command classes for all non-secure nodes. However, a secure node supporting command classes insecurely MUST also accept the same command classes security encapsulated. It SHOULD be noted that the rules apply for supported and controlled command classes.
If the node is included into a non-secure network, it MAY choose to support command classes it would not support if it had been included into a secure network.

An example of this could be a relay / switch.

	Before Inclusion	**Included Secure**	**Included Non-Secure**
Security Command Supported Report Frame	-	Binary Switch	-
Node Info Frame	Security Binary Switch Version	Security Version	Security Binary Switch Version

It is up to the implementation of each application to decide which commands SHOULD be supported using security encapsulation and non-secure.

If a command class is only supported using security encapsulation it MUST NOT be listed in the node info frame, but MUST instead be listed in the security commands supported report frame. Additionally, the node information frame MUST contain the security command class.

An exception to this rule is the Basic Command Class, which MUST be listed in the Node Info Frame if it is supported unsecure, and if not listed implied Support Secure without being listed in the secure list.

Precautions SHOULD be taken when setting up the network, since the order of inclusion will be able to change the supported functionality of the devices. For example if a non-secure controller included the above relay it would be able to switch it on / off, but if included using a secure controller version request would be possible non-securely.

In a secure network, initially only the including controller will have any knowledge about what nodes in the network have been setup securely. If a node wishes to talk to another node it MAY send a Security Command Supported Get command encapsulated to the other node. If a Security Commands Supported Report is returned the node is in possession of a valid network key, and is part of the secure network. This mechanism MAY also be used by the including controller to ensure that the node has been included properly.

CONFIDENTIAL

4.93.3.1 Multi Channel Handling

Any device that supports Security and Multi Channel Command MAY choose to support a different set of Command Classes securely for each endpoint. An End Point with support for Security MUST report the Security Command Class as supported for that End Point. The command classes supported for each endpoint securely is determined by using the Security Commands Supported Get command sent to each individual endpoint Security Encapsulated. Hence, the encapsulation order is: Security Encapsulation – Multi Channel Encapsulation – Security Commands Supported Get Command

When communicating with a device that supports multiple endpoints, the Security Encapsulation is added outside on the Multi Channel Command Class. Meaning the receiving device MUST first remove the Security Encapsulation, store the information how the package was received and forward it to the individual handler that the Multi Channel encapsulation specified.

4.93.3.2 Security Commands Supported Get Command

The device uses Security Commands Supported Get Command to request which commands the device supports using Security Encapsulation. A node MAY choose only to support a command as 'supported' and/or 'controlled', when it is security encapsulated, in which case it MUST NOT be shown in the NIF, but it will be shown in the Security Commands Supported Report Command.

This command MUST only be send encapsulated by the Security Message Encapsulation command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = SECURITY_COMMANDS_SUPPORTED_GET							

CONFIDENTIAL

4.93.3.3   Security Commands Supported Report Command

The device uses Security Commands Supported Report Command as a response to a Security Commands Supported Get command. The report informs the requesting node of which command classes is supported using security encapsulation. It is mandatory to report all command classes that the device supports and controls using the Security command class.

This command MUST only be send encapsulated by the Security Message Encapsulation command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = SECURITY_COMMANDS_SUPPORTED_REPORT							
Reports to follow							
Command Class (0x20 – 0xEE) 1 (support)							
…							
Command Class (0x20 – 0xEE) N (support)							
COMMAND_CLASS_MARK							
Command Class (0x20 – 0xEE) 1 (control)							
…							
Command Class (0x20 – 0xEE) K (control)							

CONFIDENTIAL

To support extended command classes use the following format. Note that these can be mixed. Please refer to section 6 in [3].

This command MUST only be send encapsulated by the Security Message Encapsulation command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SECURITY							
Command = SECURITY_COMMANDS_SUPPORTED_REPORT							
Reports to follow							
Command Class MSB (0xF1 – 0xFF) 1							
Command Class LSB (0x00 – 0xFF) 1							
…							
Command Class MSB (0xF1 – 0xFF) N							
Command Class LSB (0x00 – 0xFF) N							
COMMAND_CLASS_MARK							
Command Class MSB (0xF1 – 0xFF) 1							
Command Class LSB (0x00 – 0xFF) 1							
…							
Command Class MSB (0xF1 – 0xFF) K							
Command Class LSB (0x00 – 0xFF) K							

Reports to follow (8 bits)

This value indicates how many report frames left before transferring the entire list of command classes.

Command Class 1 … N (8 bits / N * 8 bits)

The command class identifier as described in section 6 in [3]. 0x20 – 0xEE Application Command Classes, 0xF1 – 0xFF Extended Application Command Classes.

Command Class Mark (8 bits)

The COMMAND_CLASS_MARK used to indicate that all preceding command classes are supported, and all following command classes are controlled.

CONFIDENTIAL

4.94   Sensor Configuration Command Class, version 1

This Sensor Configuration Command Class adds the possibility for sensors to act on either a measured value or on a preconfigured value. With this command class an application can act on a specific event. It is up to the application to implement the actual event. This could e.g. be implementation of the Association Command Class where the application would activate a group based on a trigger from the sensor.

The trigger types that can be configured are the same types as the values specified in the Multilevel Sensor Command Class

Most movement sensors can be configured to "ignore" movement if it is not dark. Typically this is done mechanically. With the Sensor Configuration Command Class this can be configured via Z-Wave in an open command class (not the Configuration Command Class since it would then be different for each manufacturer).

A device supporting the Sensor Configuration Command Class can be configured via the trigger level, but the decision on what the level change SHOULD trigger is up to the application. For the movement sensor this trigger level could be an input parameter to the logic that controls the light.

4.94.1   Sensor Trigger Level Set Command

The Sensor Trigger Level Set Command can be used to set different triggers to either a specified value or to the current measured value. The Command also supports to restore a factory default value.

All configurable trigger type and values MUST be mapped direct from the Multilevel Sensor Command Class.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_CONFIGURATION							
Command = SENSOR_TRIGGER_LEVEL_SET							
Default	Current	Reserved					
Sensor Type							
Precision		Scale		Size			
Trigger Value							
…							

Default (1 bit)

Reset level of trigger type to factory default when this bit is set to 1. If any value is set in this frame when the Default bit is 1 this value will be ignored.

Current (1 bit)

The current measured value will be stored as trigger value when this bit is set to 1. The trigger value in this frame will be ignored when the Current bit is set to 1.

Reserved (6 bits)

CONFIDENTIAL

This field is reserved for future use. Controlling devices MUST set this field to 0 (zero), while supporting devices MUST ignore this field.

Precision (3 bits)

The precision field describes what the precision of the trigger value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bits)

The Scale used to indicate what unit the trigger uses. Refer to the table in the Multilevel Sensor Command Class with respect to defined scales for the relevant triggers. New scales/values can be requested from Sigma Designs.

Size (3 bits)

The size field indicates the number of bytes used for the trigger value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

Sensor Type (8 bits)

Type specifies what type of trigger this Command will set. Refer to the Multilevel Sensor Command Class specification, where Sensor Type is defined in the Multilevel Sensor Report Command.

Trigger Value

Refer to the Multilevel Sensor Report Command for information on what trigger values to set.

4.94.2 Sensor Trigger Level Get Command

The Sensor Trigger Level Get Command can request the stored trigger level.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_CONFIGURATION							
Command = SENSOR_TRIGGER_LEVEL_GET							

4.94.3 Sensor Trigger Level Report Command

The Sensor Trigger Level Report Command returns the stored trigger value.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_CONFIGURATION							
Command = SENSOR_TRIGGER_LEVEL_REPORT							
Sensor Type							
Precision			Scale		Size		
Trigger Value							
…							

Sensor Type (8 bits)

Refer to the Sensor Trigger Level Set Command Class.

Precision (3 bits)

Refer to the Sensor Trigger Level Set Command Class.

Scale (2 bits)

Refer to the Sensor Trigger Level Set Command Class.

Size (3 bits)

Refer to the Sensor Trigger Level Set Command Class.

Trigger Value

Refer to the Sensor Trigger Level Set Command Class.

4.94.4 Mapping example

The report structure from the Multilevel Sensor Command Class can be mapped direct into the Sensor Configuration Set Command Class. This example frame below will set the trigger level in the receiving device to 10.25 degree Celsius.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SENSOR_CONFIGURATION							
Command = SENSOR_TRIGGER_LEVEL_SET							
Default(0)	Current(0)	Reserved					
Temperature (0x01)							
Precision (010b)			Celsius (00b)		Size (010b)		
0x04							
0x01							

CONFIDENTIAL

4.95   Simple AV Control Command Class, version 1-4

This Simple AV Control Command Class used to control an AV device in a Z-Wave network. The Simple AV Control Command Class is suited for IR remote replacement. Furthermore, this command class supports Windows Vista Media Center and Media Center 2005 remote controls.

4.95.1   Simple AV Control Set Command

The Simple AV Control Set Command used to control an AV device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SIMPLE_AV_CONTROL							
Command = SIMPLE_AV_CONTROL_SET							
Sequence Number							
Reserved				Key Attributes			
Item ID MSB							
Item ID LSB							
Command MSB,1							
Command LSB,1							
…							
Command MSB,N							
Command LSB,N							

Sequence Number (8 bits)

The sequence number is incremented each time a Simple AV Control Set Command is issued. The receiving device uses the sequence number to ignore duplicates.

Key Attributes (3 bits)

The key attributes specifies the state of the key. Currently the following key attribute definitions exist:

Key Attribute	Description
0x00	Key Down – Sent when a new key is pressed. It is mandatory to send a Simple AV Control Set Command when this event occurs.
0x01	Key Up – Sent when the key is released. It is OPTIONAL to send a Simple AV Control Set Command when this event occurs. Only the sequence number and key attribute parameter is changed in the Command.
0x02	Keep Alive – Sent every 100-200ms while the key is still held down. Event used as a failsafe feature for the ramping function, e.g. avoid volume jumps to maximum in case a key up event is not received. The keep alive event can also be used to control the speed of the ramping function, e.g. the first few seconds of the key held down is the speed slow and afterwards will it gradually accelerate. It is OPTIONAL to send a Simple AV Control Set Command when this event occurs. Only the sequence number and key attribute parameter is changed in the Command.

This list MAY evolve in the future. In case a key attribute is not supported then it SHOULD be ignored.

Reserved (5 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Item ID (16 bits)

The ID of the media item the Command SHOULD relate to. No media item is selected in case ID is equal to 0. This field is only used if the AV Content Directory Meta Data Command Class is used.

Command MSB, Command LSB (N x 16 bits)

A 2 byte AV control Command according to the table below. It is possible to send a sequence of Commands in one frame. In case a Command number is not supported then it SHOULD be ignored. Be aware of that device related labels are not sent but only used internal in the remote to set up the appropriate address. The address can be a node ID. Command number 1 through 40 is the most popular Commands used in remotes. Command number 41 through 363 is less popular and is sorted in alphanumerical order. Finally is support for Windows Vista Media Center and Media Center 2005 remote controls added from 364 to 377 including 16, 200 and 231. This list MAY evolve in the future.

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
1	0x0001	Mute	
2	0x0002	Volume Down	Level Down
3	0x0003	Volume Up	Level Up
4	0x0004	Channel Up	Program Up
5	0x0005	Channel Down	Program Down
6	0x0006	0	Preset 10
7	0x0007	1	Preset 1
8	0x0008	2	Preset 2
9	0x0009	3	Preset 3
10	0x000A	4	Preset 4
11	0x000B	5	Preset 5
12	0x000C	6	Preset 6
13	0x000D	7	Preset 7
14	0x000E	8	Preset 8
15	0x000F	9	Preset 9
16	0x0010	Last Channel	Recall, Previous Channel (WMC)
17	0x0011	Display	Info
18	0x0012	Favorite Channel	Favorite
19	0x0013	Play	
20	0x0014	Stop	
21	0x0015	Pause	Still
22	0x0016	Fast Forward	Search Forward
23	0x0017	Rewind	Search Reverse
24	0x0018	Instant Replay	Replay
25	0x0019	Record	
26	0x001A	AC3	Dolby Digital
27	0x001B	PVR Menu	Tivo
28	0x001C	Guide	EPG
29	0x001D	Menu	Settings
30	0x001E	Menu Up	Adjust Up
31	0x001F	Menu Down	Adjust Down
32	0x0020	Menu Left	Cursor Left
33	0x0021	Menu Right	Cursor Right
34	0x0022	Page Up	
35	0x0023	Page Down	
36	0x0024	Select	OK
37	0x0025	Exit	
38	0x0026	Input	Input Select
39	0x0027	Power	Standby
40	0x0028	Enter Channel	Channel Enter
41	0x0029	10	
42	0x002A	11	
43	0x002B	12	
44	0x002C	13	

CONFIDENTIAL

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
45	0x002D	14	
46	0x002E	15	
47	0x002F	16	
48	0x0030	+10	10+
49	0x0031	+20	20+
50	0x0032	+100	
51	0x0033	-/--	
52	0x0034	3-CH	
53	0x0035	3D	Simulated Stereo
54	0x0036	6-CH Input	6 Channel
55	0x0037	A	
56	0x0038	Add	Write
57	0x0039	Alarm	
58	0x003A	AM	
59	0x003B	Analog	
60	0x003C	Angle	
61	0x003D	Antenna	External
62	0x003E	Antenna East	
63	0x003F	Antenna West	
64	0x0040	Aspect	Size
65	0x0041	Audio 1	Audio
66	0x0042	Audio 2	
67	0x0043	Audio 3	
68	0x0044	Audio Dubbing	
69	0x0045	Audio Level Down	
70	0x0046	Audio Level Up	
71	0x0047	Auto/Manual	
72	0x0048	Aux 1	Aux
73	0x0049	Aux 2	
74	0x004A	B	
75	0x004B	Back	Previous Screen
76	0x004C	Background	Backlight
77	0x004D	Balance	
78	0x004E	Balance Left	
79	0x004F	Balance Right	
80	0x0050	Band	FM/AM
81	0x0051	Bandwidth	Wide/Narrow
82	0x0052	Bass	
83	0x0053	Bass Down	
84	0x0054	Bass Up	
85	0x0055	Blank	
86	0x0056	Breeze Mode	
87	0x0057	Bright	Brighten
88	0x0058	Brightness	

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
89	0x0059	Brightness Down	
90	0x005A	Brightness Up	
91	0x005B	Buy	
92	0x005C	C	
93	0x005D	Camera	
94	0x005E	Category Down	
95	0x005F	Category Up	
96	0x0060	Center	
97	0x0061	Center Down	Center Volume Down
98	0x0062	Center Mode	
99	0x0063	Center Up	Center Volume Up
100	0x0064	Channel/Program	C/P
101	0x0065	Clear	Cancel
102	0x0066	Close	
103	0x0067	Closed Caption	CC
104	0x0068	Cold	A/C
105	0x0069	Color	
106	0x006A	Color Down	
107	0x006B	Color Up	
108	0x006C	Component 1	RGB 1
109	0x006D	Component 2	RGB 2
110	0x006E	Component 3	
111	0x006F	Concert	
112	0x0070	Confirm	Check
113	0x0071	Continue	Continuous
114	0x0072	Contrast	
115	0x0073	Contrast Down	
116	0x0074	Contrast Up	
117	0x0075	Counter	
118	0x0076	Counter Reset	
119	0x0077	D	
120	0x0078	Day Down	
121	0x0079	Day Up	
122	0x007A	Delay	
123	0x007B	Delay Down	
124	0x007C	Delay Up	
125	0x007D	Delete	Erase
126	0x007E	Delimiter	Sub-Channel
127	0x007F	Digest	
128	0x0080	Digital	
129	0x0081	Dim	Dimmer
130	0x0082	Direct	
131	0x0083	Disarm	
132	0x0084	Disc	

CONFIDENTIAL

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
133	0x0085	Disc 1	
134	0x0086	Disc 2	
135	0x0087	Disc 3	
136	0x0088	Disc 4	
137	0x0089	Disc 5	
138	0x008A	Disc 6	
139	0x008B	Disc Down	
140	0x008C	Disc Up	
141	0x008D	Disco	
142	0x008E	Edit	
143	0x008F	Effect Down	
144	0x0090	Effect Up	
145	0x0091	Eject	Open/Close
146	0x0092	End	
147	0x0093	EQ	Equalizer
148	0x0094	Fader	
149	0x0095	Fan	
150	0x0096	Fan High	
151	0x0097	Fan Low	
152	0x0098	Fan Medium	
153	0x0099	Fan Speed	
154	0x009A	Fastext Blue	
155	0x009B	Fastext Green	
156	0x009C	Fastext Purple	
157	0x009D	Fastext Red	
158	0x009E	Fastext White	
159	0x009F	Fastext Yellow	
160	0x00A0	Favorite Channel Down	
161	0x00A1	Favorite Channel Up	
162	0x00A2	Finalize	
163	0x00A3	Fine Tune	
164	0x00A4	Flat	
165	0x00A5	FM	
166	0x00A6	Focus Down	
167	0x00A7	Focus Up	
168	0x00A8	Freeze	
169	0x00A9	Front	
170	0x00AA	Game	
171	0x00AB	GoTo	Index Search
172	0x00AC	Hall	
173	0x00AD	Heat	
174	0x00AE	Help	

CONFIDENTIAL

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
175	0x00AF	Home	
176	0x00B0	Index	VISS
177	0x00B1	Index Forward	
178	0x00B2	Index Reverse	
179	0x00B3	Interactive	Planner
180	0x00B4	Intro Scan	
181	0x00B5	Jazz	
182	0x00B6	Karaoke	
183	0x00B7	Keystone	
184	0x00B8	Keystone Down	
185	0x00B9	Keystone Up	
186	0x00BA	Language	SAP
187	0x00BB	Left Click	
188	0x00BC	Level	Volume
189	0x00BD	Light	Lamp
190	0x00BE	List	My Shows
191	0x00BF	Live TV	Return to Live
192	0x00C0	Local/Dx	
193	0x00C1	Loudness	
194	0x00C2	Mail	Email
195	0x00C3	Mark	Bookmark
196	0x00C4	Memory Recall	
197	0x00C5	Monitor	Tape Monitor
198	0x00C6	Movie	
199	0x00C7	Multi Room	
200	0x00C8	Music	TV/Radio, My Music (WMC)
201	0x00C9	Music Scan	Memory Scan
202	0x00CA	Natural	
203	0x00CB	Night	
204	0x00CC	Noise Reduction	Dolby NR
205	0x00CD	Normalize	Personal Preference
206	0x00CE	Discrete input Cable	CATV
207	0x00CF	Discrete input CD 1	CD
208	0x00D0	Discrete input CD 2	CDR
209	0x00D1	Discrete input CDR	Compact Disc Recorder
210	0x00D2	Discrete input DAT	Digital Audio Tape
211	0x00D3	Discrete input DVD	Digital Video Disk
212	0x00D4	Discrete input DVI	Digital Video Interface
213	0x00D5	Discrete input HDTV	
214	0x00D6	Discrete input LD	Laser Disc
215	0x00D7	Discrete input MD	Mini Disc
216	0x00D8	Discrete input PC	Personal Computer

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
217	0x00D9	Discrete input PVR	Personal Video Recorder
218	0x00DA	Discrete input TV	
219	0x00DB	Discrete input TV/VCR	TV/DVD
220	0x00DC	Discrete input VCR	
221	0x00DD	One Touch Playback	OTPB
222	0x00DE	One Touch Record	OTR
223	0x00DF	Open	
224	0x00E0	Optical	
225	0x00E1	Options	
226	0x00E2	Orchestra	
227	0x00E3	PAL/NTSC	System Select
228	0x00E4	Parental Lock	Parental Control
229	0x00E5	PBC	Playback Control
230	0x00E6	Phono	
231	0x00E7	Photos	Pictures, My Pictures (WMC)
232	0x00E8	Picture Menu	Picture Adjust
233	0x00E9	Picture Mode	Smart Picture
234	0x00EA	Picture Mute	
235	0x00EB	PIP Channel Down	
236	0x00EC	PIP Channel Up	
237	0x00ED	PIP Freeze	
238	0x00EE	PIP Input	PIP Mode
239	0x00EF	PIP Move	PIP Position
240	0x00F0	PIP Off	
241	0x00F1	PIP On	PIP
242	0x00F2	PIP Size	
243	0x00F3	PIP Split	Multi Screen
244	0x00F4	PIP Swap	PIP Exchange
245	0x00F5	Play Mode	
246	0x00F6	Play Reverse	
247	0x00F7	Power Off	
248	0x00F8	Power On	
249	0x00F9	PPV	Pay Per View
250	0x00FA	Preset	
251	0x00FB	Program	Program Memory
252	0x00FC	Progressive Scan	Progressive
253	0x00FD	ProLogic	Dolby Prologic
254	0x00FE	PTY	Audio Program Type
255	0x00FF	Quick Skip	Commercial Skip
256	0x0100	Random	Shuffle
257	0x0101	RDS	Radio Data System
258	0x0102	Rear	

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
259	0x0103	Rear Volume Down	
260	0x0104	Rear Volume Up	
261	0x0105	Record Mute	
262	0x0106	Record Pause	
263	0x0107	Repeat	
264	0x0108	Repeat A-B	
265	0x0109	Resume	
266	0x010A	RGB	Red Green Blue Component Video
267	0x010B	Right Click	
268	0x010C	Rock	
269	0x010D	Rotate Left	
270	0x010E	Rotate Right	
271	0x010F	SAT	Sky
272	0x0110	Scan	Channel Scan
273	0x0111	Scart	
274	0x0112	Scene	
275	0x0113	Scroll	
276	0x0114	Services	
277	0x0115	Setup Menu	Setup
278	0x0116	Sharp	
279	0x0117	Sharpness	
280	0x0118	Sharpness Down	
281	0x0119	Sharpness Up	
282	0x011A	Side A/B	
283	0x011B	Skip Forward	Next
284	0x011C	Skip Reverse	Previous
285	0x011D	Sleep	Off Timer
286	0x011E	Slow	
287	0x011F	Slow Forward	
288	0x0120	Slow Reverse	
289	0x0121	Sound Menu	Audio Menu
290	0x0122	Sound Mode	Smart Sound
291	0x0123	Speed	Record Speed
292	0x0124	Speed Down	
293	0x0125	Speed Up	
294	0x0126	Sports	Digital Surround Processing
295	0x0127	Stadium	
296	0x0128	Start	
297	0x0129	Start ID Erase	Erase
298	0x012A	Start ID Renumber	Renumber
299	0x012B	Start ID Write	Write
300	0x012C	Step	
301	0x012D	Stereo/Mono	L/R
302	0x012E	Still Forward	Frame Advance

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
303	0x012F	Still Reverse	Frame Reverse
304	0x0130	Subtitle	Subtitle On-Off
305	0x0131	Subwoofer Down	
306	0x0132	Subwoofer Up	
307	0x0133	Super Bass	Bass Boost
308	0x0134	Surround	
309	0x0135	Surround Mode	Sound Field
310	0x0136	S-Video	
311	0x0137	Sweep	Oscillate
312	0x0138	Synchro Record	CD Synchro
313	0x0139	Tape 1	Deck 1
314	0x013A	Tape 1-2	Deck 1-2
315	0x013B	Tape 2	Deck 2
316	0x013C	Temperature Down	
317	0x013D	Temperature Up	
318	0x013E	Test Tone	
319	0x013F	Text	Teletext
320	0x0140	Text Expand	
321	0x0141	Text Hold	
322	0x0142	Text Index	
323	0x0143	Text Mix	
324	0x0144	Text Off	
325	0x0145	Text Reveal	
326	0x0146	Text Subpage	
327	0x0147	Text Timed Page	
328	0x0148	Text Update	Text Cancel
329	0x0149	Theater	Cinema EQ
330	0x014A	Theme	Category Select
331	0x014B	Thumbs Down	
332	0x014C	Thumbs Up	
333	0x014D	Tilt Down	
334	0x014E	Tilt Up	
335	0x014F	Time	Clock
336	0x0150	Timer	
337	0x0151	Timer Down	
338	0x0152	Timer Up	
339	0x0153	Tint	
340	0x0154	Tint Down	
341	0x0155	Tint Up	
342	0x0156	Title	Top Menu
343	0x0157	Track	Chapter
344	0x0158	Tracking	
345	0x0159	Tracking Down	
346	0x015A	Tracking Up	

CONFIDENTIAL

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
347	0x015B	Treble	
348	0x015C	Treble Down	
349	0x015D	Treble Up	
350	0x015E	Tune Down	Audio Tune Down
351	0x015F	Tune Up	Audio Tune Up
352	0x0160	Tuner	
353	0x0161	VCR Plus+	Showview
354	0x0162	Video 1	A/V 1
355	0x0163	Video 2	A/V 2
356	0x0164	Video 3	A/V 3
357	0x0165	Video 4	A/V 4
358	0x0166	Video 5	A/V 5
359	0x0167	View	
360	0x0168	Voice	Vocals
361	0x0169	Zoom	Magnify
362	0x016A	Zoom In	Zoom Up
363	0x016B	Zoom Out	Zoom Down
364	0x016C	eHome	(WMC), version 2
365	0x016D	Details	(WMC), version 2
366	0x016E	DVD Menu	(WMC), version 2
367	0x016F	My TV	(WMC), version 2
368	0x0170	Recorded TV	(WMC), version 2
369	0x0171	My Videos	(WMC), version 2
370	0x0172	DVD Angle	(WMC), version 2
371	0x0173	DVD Audio	(WMC), version 2
372	0x0174	DVD Subtitle	(WMC), version 2
373	0x0175	Radio	(WMC), version 2
374	0x0176	#	(WMC), version 2
375	0x0177	*	(WMC), version 2
376	0x0178	OEM 1	(WMC), version 2
377	0x0179	OEM 2	(WMC), version 2
378	0x017A	Info	Used to request information, version 3
379	0x017B	CAPS NUM	Switch between numeric and alpha (Shift), version 3
380	0x017C	TV MODE	Cycles through video output modes/resolutions, version 3
381	0x017D	SOURCE	Displays the possible sources for the playback. [NFS, ext., USB, UPnP,…], version 3
382	0x017E	FILEMODE	File manipulation. Add/remove to list, create folder, rename file,…, version 3
383	0x017F	Time Seek	This seeks to time position. Used for DVD/CD/others, version 3
384	0x0180	Mouse enable	Mouse pointer enable, version 4
385	0x0181	Mouse disable	Mouse pointer disable, version 4
386	0x0182	VOD	Video on demand, version 4
387	0x0183	Thumbs Up	Thumbs up for positive feedback in GUI, version 4
388	0x0184	Thumbs Down	Thumbs down for negative feedback in GUI, version 4

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
389	0x0185	Apps	Application selection/launch, version 4
390	0x0186	Mouse toggle	Will toggle a mouse pointer between on and off, version 4
391	0x0187	TV Mode	Will direct an AV device to go the TV mode (the mode is configured on the device), version 4
392	0x0188	DVD Mode	Will direct an AV device to go the DVD mode (the mode is configured on the device), version 4
393	0x0189	STB Mode	Will direct an AV device to go the STB mode (the mode is configured on the device), version 4
394	0x018A	AUX Mode	Will direct an AV device to go the AUX mode (the mode is configured on the device), version 4
395	0x018B	BluRay Mode	Will direct an AV device to go the BluRay mode (the mode is configured on the device), version 4
396	0x018C		Reserved for more mode keys, version 4
397	0x018D		Reserved for more mode keys, version 4
398	0x018E		Reserved for more mode keys, version 4
399	0x018F		Reserved for more mode keys, version 4
400	0x0190		Reserved for more mode keys, version 4
401	0x0191		Reserved for more mode keys, version 4
402	0x0192		Reserved for more mode keys, version 4
403	0x0193		Reserved for more mode keys, version 4
404	0x0194	Standby 1	Used for AV devices that support multiple standby mode. Power ON SHOULD be used to turn on the device, version 4
405	0x0195	Standby 2	Used for AV devices that support multiple standby mode. Power ON SHOULD be used to turn on the device, version 4
406	0x0196	Standby 3	Used for AV devices that support multiple standby mode. Power ON SHOULD be used to turn on the device, version 4
407	0x0197	HDMI 1	Discrete command used to set an AV device to HDMI input 1, version 4
408	0x0198	HDMI 2	Discrete command used to set an AV device to HDMI input 2, version 4
409	0x0199	HDMI 3	Discrete command used to set an AV device to HDMI input 3, version 4
410	0x019A	HDMI 4	Discrete command used to set an AV device to HDMI input 4, version 4
411	0x019B	HDMI 5	Discrete command used to set an AV device to HDMI input 5, version 4
412	0x019C	HDMI 6	Discrete command used to set an AV device to HDMI input 6, version 4
413	0x019D	HDMI 7	Discrete command used to set an AV device to HDMI input 7, version 4
414	0x019E	HDMI 8	Discrete command used to set an AV device to HDMI input 8, version 4
415	0x019F	HDMI 9	Discrete command used to set an AV device to HDMI input 9, version 4
416	0x01A0	USB 1	Discrete command used to set an AV device to USB input 1, version 4
417	0x01A1	USB 2	Discrete command used to set an AV device to USB input 2, version 4

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
418	0x01A2	USB 3	Discrete command used to set an AV device to USB input 3, version 4
419	0x01A3	USB 4	Discrete command used to set an AV device to USB input 4, version 4
420	0x01A4	USB 5	Discrete command used to set an AV device to USB input 5, version 4
421	0x01A5	ZOOM 4:3 Normal	Discrete commands that is used to set a TV a direct Zoom mode, version 4
422	0x01A6	ZOOM 4:3 Zoom	Discrete commands that is used to set a TV a direct Zoom mode, version 4
423	0x01A7	ZOOM 16:9 Normal	Discrete commands that is used to set a TV a direct Zoom mode, version 4
424	0x01A8	ZOOM 16:9 Zoom	Discrete commands that is used to set a TV a direct Zoom mode, version 4
425	0x01A9	ZOOM 16:9 Wide 1	Discrete commands that is used to set a TV a direct Zoom mode, version 4
426	0x01AA	ZOOM 16:9 Wide 2	Discrete commands that is used to set a TV a direct Zoom mode, version 4
427	0x01AB	ZOOM 16:9 Wide 3	Discrete commands that is used to set a TV a direct Zoom mode, version 4
428	0x01AC	ZOOM 16:9 Cinema	Discrete commands that is used to set a TV a direct Zoom mode, version 4
429	0x01AD	ZOOM Default	Discrete commands that is used to set a TV the default Zoom mode, version 4
430	0x01AE		Reserved for more Zoom modes, version 4
431	0x01BF		Reserved for more Zoom modes, version 4
432	0x01B0	Auto Zoom	Will set Zoom mode automatically, version 4
433	0x01B1	ZOOM Set as Default Zoom	Will set the current active Zoom level to default, version 4
434	0x01B2	Mute ON	Discrete Mute ON command, version 4
435	0x01B3	Mute OFF	Discrete Mute OFF command, version 4
436	0x01B4	AUDIO Mode AUDYSSEY AUDIO OFF	Discrete Audio mode for Audussey audio processing (OFF) , version 4
437	0x01B5	AUDIO Mode AUDYSSEY AUDIO LO	Discrete Audio mode for Audussey audio processing (Low) , version 4
438	0x01B6	AUDIO Mode AUDYSSEY AUDIO MED	Discrete Audio mode for Audussey audio processing (Medium) , version 4
439	0x01B7	AUDIO Mode AUDYSSEY AUDIO HI	Discrete Audio mode for Audussey audio processing (High) , version 4
440	0x01B8		
441	0x01B9		
442	0x01BA	AUDIO Mode SRS SURROUND ON	Discrete Audio mode for SRS audio processing, version 4
443	0x01BB	AUDIO Mode SRS SURROUND OFF	Discrete Audio mode for SRS audio processing, version 4

Command # [Decimal]	Command # [Hexadecimal]	Universal Label	Description
444	0x01BC		
445	0x01BD		
446	0x01BE		
447	0x01BF	Picture Mode Home	Discrete picture for TVs, version 4
448	0x01C0	Picture Mode Retail	Discrete picture for TVs, version 4
449	0x01C1	Picture Mode Vivid	Discrete picture for TVs, version 4
450	0x01C2	Picture Mode Standard	Discrete picture for TVs, version 4
451	0x01C3	Picture Mode Theater	Discrete picture for TVs, version 4
452	0x01C4	Picture Mode Sports	Discrete picture for TVs, version 4
453	0x01C5	Picture Mode Energy savings	Discrete picture for TVs, version 4
454	0x01C6	Picture Mode Custom	Discrete picture for TVs, version 4
455	0x01C7	Cool	Discrete picture temperature adjustments, version 4
456	0x01C8	Medium	Discrete picture temperature adjustments, version 4
457	0x01C9	Warm_D65	Discrete picture temperature adjustments, version 4
458	0x01CA	CC ON	Discrete Closed caption commands, version 4
459	0x01CB	CC OFF	Discrete Closed caption commands, version 4
460	0x01CC	Video Mute ON	Discrete Video mute command, version 4
461	0x01CD	Video Mute OFF	Discrete Video mute command, version 4
462	0x01CE	Next Event	Go to next state or event , version 4
463	0x01CF	Previous Event	Go to previous state or event, version 4
464	0x01D0	CEC device list	Brings up the CES device list, version 4
465	0x01D1	MTS SAP	Secondary Audio programming, version 4

Table 25, AV Control codes and associated label

4.95.2 Simple AV Control Get Command

The Simple AV Control Get Command is used to request the number of reports necessary to report the supported AC Commands from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SIMPLE_AV_CONTROL							
Command = SIMPLE_AV_CONTROL_GET							

4.95.3 Simple AV Control Report Command

The Simple AV Control Report Command used to report the necessary number of reports to report the supported AC Commands from the device. Use Simple AV Control Get Command to request the Simple AV Control Report Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SIMPLE_AV_CONTROL							
Command = SIMPLE_AV_CONTROL_REPORT							
Number of reports							

Number of reports (8 bits)

The number of reports necessary to report the entire list of supported AC Commands.

4.95.4 Simple AV Control Supported Get Command

The Simple AV Control Supported Get Command is used to request the AV Commands supported by the AV device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SIMPLE_AV_CONTROL							
Command = SIMPLE_AV_CONTROL_SUPPORTED_GET							
Report No							

Report No (8 bits)

Report no. field is used to request wanted report number. The report no. values MUST be a sequence starting from 1.

CONFIDENTIAL

4.95.5 Simple AV Control Supported Report Command

The Simple AV Control Supported Report Command used to report the supported AC Commands from the device. The Simple AV Control Report Command can be send requested by the Simple AV Control Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_SIMPLE_AV_CONTROL							
Command = SIMPLE_AV_CONTROL_SUPPORTED_REPORT							
Report No							
Bit Mask 1							
…							
Bit Mask N							

Report No (8 bits)

Report no. field specify the request report number.

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported AV Control Commands by the device. The bit 0 in Bit Mask 1 field used to indicate whether Command #1 is supported or not. The Command #1 is supported if the bit is 1 and the opposite if 0. The bit 1 in Bit Mask 1 field used by Command #2 and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported Command #. Mask fields bigger than 45 bytes not allowed. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.96 Tariff Table Configuration Command Class, version 1

The Tariff Table Configuration Command Class defines the cost for a range of rates.

The Tariff Table configuration commands are separated for the Tariff Table monitor commands in the Tariff Table Monitor Command Class, allowing the classes to be OPTIONALly supported at different Z-Wave security levels. (E.g. Tariff table monitor commands could be supported in any device, while enabling a strict and certificate based security solution for the Tariff Table Configuration Command class). Please refer to the Hybrid Security Command class for more details regarding Z-Wave security levels.

4.96.1 Tariff Table Supplier Set Command

The Tariff Table Supplier Set Command is used to set the name of the utility supplier in the metering device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_CONFIG							
Command = TARIFF_TBL_SUPPLIER_SET							
Utility Timestamp Year 1							
Utility Timestamp Year 2							
Utility Timestamp Month							
Utility Timestamp Day							
Utility Timestamp Hour Local Time							
Utility Timestamp Minute Local Time							
Utility Timestamp Second Local Time							
Currency 1							
Currency 2							
Currency 3							
Standing Charge Precision			Standing Charge Period				
Standing Charge Value 1							
Standing Charge Value 2							
Standing Charge Value 3							
Standing Charge Value 4							
Reserved			Number of Supplier Characters				
Supplier Character 1							
…							
Supplier Character N							

CONFIDENTIAL

Utility Timestamp Year 1..2 (16 bit)

Tariff applies from the specified year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Utility Timestamp Month (8 bit)

Tariff applies from the specified month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that a accumulated value is not determined yet.

Utility Timestamp Day (8 bit)

Tariff applies from the specified day of the month between 01 and 31.

Utility Timestamp Hour Local Time (8 bit)

Tariff applies from the specified number of complete hours that have passed since midnight (00-23) in local time.

Utility Timestamp Minute Local Time (8 bit)

Tariff applies from the specified number of complete minutes that have passed since the start of the hour (00-59) in local time.

Utility Timestamp Second Local Time (8 bit)

Tariff applies from the specified number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

Currency 1 ... 3

ISO 4217 defines the currency code. In the table below are some examples of the codes listed:

Currency Code	Currency 1	Currency 2	Currency 3
Pound sterling	G	B	P
US Dollar	U	S	D

Standing Charge Period (5 bit)

This field indicates the stated period that standing charge applies e.g. 50p/week.

Period	Value
Weekly	0x01
Monthly	0x02
Quarterly	0x03
Yearly	0x04
Reserved	0x05-0x1F

CONFIDENTIAL

Standing Charge Precision (3 bit)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Standing Charge Value

The Standing Charge value is a 32 bit signed field. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

Reserved (3 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Number of Supplier Characters (5 bit)

Number of characters defining the name of the utility supplier ID (1 … 32).

Supplier Character 1 … N (Number of Supplier Characters * 8 bit)

The supplier character fields hold the string identifying the utility supplier. The character presentation uses standard ASCII codes (values 128-255 are ignored).

CONFIDENTIAL

4.96.2 Tariff Table Set Command

The Tariff Table Set Command add a tariff to a given rate parameter set identifier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL							
Command = TARIFF_TBL_REPORT							
Rate Parameter Set ID							
Tariff Precision			Reserved				
Tariff Value 1							
Tariff Value 2							
Tariff Value 3							
Tariff Value 4							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID indicates the requested parameter set.

Reserved (5 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Tariff Precision (3 bit)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

CONFIDENTIAL

Tariff Value

The Tariff value is a 32 bit signed field. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 4 bytes	
Decimal	**Hexadecimal**
2147483647	0x7FFFFFFF
..	..
1073741823	0x3FFFFFFF
..	..
1	0x00000001
0	0x00000000
-1	0xFFFFFFFF
..	..
-1073741823	0xC0000001
..	..
-2147483648	0x80000000

4.96.3 Tariff Table Remove Command

The Tariff Table Remove Command is used to remove rate parameter set(s).

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_CONFIG							
Command = TARIFF_TBL_REMOVE							
Reserved		Rate Parameter Set IDs					
Rate Parameter Set ID 1							
..							
Rate Parameter Set ID N							

Reserved (2 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Rate Parameter Set IDs (6 bit)

The rate parameter set id's indicates the number of rate parameter set id's in the command.

Rate Parameter Set ID 1 .. Rate Parameter Set ID N

These fields contain a list of Tariffs to be removed from the Tariff Table. All Tariffs are cleared in case no Rate Parameter Set ID's are supplied.

CONFIDENTIAL

4.97 Tariff Table Monitor Command Class, version 1

The Tariff Table Monitor Command Class defines the cost for a range of rates.

4.97.1 Tariff Table Supplier Get Command

The Tariff Table Supplier Get Command is used to request the name of the utility supplier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_SUPPLIER_GET							

4.97.2 Tariff Table Supplier Report Command

The Tariff Table Supplier Report Command reports the name of the utility supplier and MUST be requested by the Tariff Table Supplier Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_SUPPLIER_REPORT							
Utility Timestamp Year 1							
Utility Timestamp Year 2							
Utility Timestamp Month							
Utility Timestamp Day							
Utility Timestamp Hour Local Time							
Utility Timestamp Minute Local Time							
Utility Timestamp Second Local Time							
Currency 1							
Currency 2							
Currency 3							
Standing Charge Precision			Standing Charge Period				
Standing Charge Value 1							
Standing Charge Value 2							
Standing Charge Value 3							
Standing Charge Value 4							
Reserved			Number of Supplier Characters				
Supplier Character 1							
…							
Supplier Character N							

CONFIDENTIAL

Refer to description of fields under the Tariff Table Supplier Set Command (section 4.96.1)

4.97.3   Tariff Table Get Command

The Tariff Table Get Command is used to request the tariff for the corresponding rate parameter set.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_GET							
Rate Parameter Set ID							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID addresses the price for the accompanying rate parameter set. The Rate Table Supported Report Command determines the number of supported rate parameter sets.

4.97.4   Tariff Table Report Command

The Tariff Table Report Command reports rate parameter set for a given rate parameter set identifier and MUST be requested by the Tariff Table Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_REPORT							
Rate Parameter Set ID							
Tariff Precision			Reserved				
Tariff Value 1							
Tariff Value 2							
Tariff Value 3							
Tariff Value 4							

Refer to description of fields under the Tariff Table Set Command (section 4.96.2)

CONFIDENTIAL

4.97.5 Tariff Table Cost Get Command

The Tariff Table Cost Get Command is used to request the cost according to rate parameter set ID, rate type, dataset mask and time interval.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_COST_GET							
Rate Parameter Set ID							
Start Year 1							
Start Year 2							
Start Month							
Start Day							
Start Hour Local Time							
Start Minute Local Time							
Stop Year 1							
Stop Year 2							
Stop Month							
Stop Day							
Stop Hour Local Time							
Stop Minute Local Time							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID indicates the requested parameter set. Rate Parameter Set ID equal to 0xFF returns overall accumulated cost.

Start Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Start Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December).

Start Day (8 bit)

Specify the day of the month between 01 and 31.

Start Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start Minute Local Time (8 bit)

CONFIDENTIAL

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Stop Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte. Setting the parameter to 0xFFFF indicates now.

Stop Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that an accumulated value is not determined yet. Setting the parameter to 0xFF indicates now.

Stop Day (8 bit)

Specify the day of the month between 01 and 31. Setting the parameter to 0xFF indicates now.

Stop Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time. Setting the parameter to 0xFF indicates now.

Stop Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time. Setting the parameter to 0xFF indicates now.

Stop Second Local Time (8 bit)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported. Setting the parameter to 0xFF indicates now.

CONFIDENTIAL

4.97.6 Tariff Table Cost Report Command

The Tariff Table Cost Report Command used to report a number of time stamped values (historical) in physical units in the device. The Tariff Table Cost Report Command can be sent unsolicited or requested by the Tariff Table Cost Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TARIFF_TBL_MONITOR							
Command = TARIFF_TBL_COST_REPORT							
Rate Parameter Set ID							
Reserved						Rate Type	
Start Year 1							
Start Year 2							
Start Month							
Start Day							
Start Hour Local Time							
Start Minute Local Time							
Stop Year 1							
Stop Year 2							
Stop Month							
Stop Day							
Stop Hour Local Time							
Stop Minute Local Time							
Currency 1							
Currency 2							
Currency 3							
Cost Precision			Reserved				
Cost Value 1							
Cost Value 2							
Cost Value 3							
Cost Value 4							

Rate Parameter Set ID (8 bit)

The Rate Parameter Set ID indicates the requested parameter set. Rate Parameter Set ID equal to 0xFF returns accumulated cost.

Reserved (6 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

Rate Type (2 bit)

Rate Type specifies the type of parameters in the report. Rate Type defined as the *Meter Rate Type* variable; refer to section 3.3.3 for a definition of the variable.

Start/Stop Year 1, 2 (16 bit)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Start/Stop Month (8 bit)

Specify the month of the year between 01 (January) and 12 (December). A year equal to 0x0000 indicates that a accumulated value is not determined yet.

Start/Stop Day (8 bit)

Specify the day of the month between 01 and 31.

Start/Stop Hour Local Time (8 bit)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Start/Stop Minute Local Time (8 bit)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Reserved (5 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Currency 1 ... 3

ISO 4217 defines the currency code. In the table below are some examples of the codes listed:

Currency Code	Currency 1	Currency 2	Currency 3
Pound sterling	G	B	P
US Dollar	U	S	D

Cost Precision (3 bit)

The precision field describes what the precision of the value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Cost Value

The Cost value is a 32 bit un-signed field. The first byte is the most significant byte.

The value 0xFFFFFFFF is reserved, and SHOULD be used to report that the cost calculation has not yet been performed.

CONFIDENTIAL

4.98 Thermostat Fan Mode Command Class, version 1

The Thermostat Fan Mode Command Class, version 1 used for the HVAC's systems manual fan.

4.98.1 Thermostat Fan Mode Set Command

The Thermostat Fan Mode Set Command used to set the fan mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE								
Command = THERMOSTAT_FAN_MODE_SET								
Reserved				Fan Mode				

Fan Mode (8 bits)

Fan Mode	Description
0	Auto / Auto Low – Will turn the manual fan operation off unless turned on by the furnace or AC. Lower speed is selected in case it is a two-speed fan.
1	On / On Low – Will turn the manual fan operation on. Lower speed is selected in case it is a two-speed fan.
2	Auto High – Will turn the manual fan operation off unless turned on by the furnace or AC. High speed is selected in case it is a two-speed fan.
3	On High – Will turn the manual fan operation on. High speed is selected in case it is a two-speed fan.

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.98.2 Thermostat Fan Mode Get Command

The Thermostat Fan Mode Get Command is used to request the fan mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE								
Command = THERMOSTAT_FAN_MODE_GET								

4.98.3 Thermostat Fan Mode Report Command

The Thermostat Fan Mode Report Command used to report the fan mode in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_REPORT							
Reserved				Fan Mode			

Fan Mode (8 bits)

Refer to description under the Thermostat Fan Mode Set Command.

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.98.4 Thermostat Fan Mode Supported Get Command

The Thermostat Fan Mode Supported Get Command is used to request the supported fan modes from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_SUPPORTED_GET							

CONFIDENTIAL

4.98.5 Thermostat Fan Mode Supported Report Command

The Thermostat Fan Mode Supported Report Command used to report the supported fan modes from the device. Application can send unsolicited Thermostat Fan Mode Supported Report commands or requested by the Fan Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE								
Command = THERMOSTAT_FAN_MODE_SUPPORTED_REPORT								
Bit Mask 1								
…								
Bit Mask N								

Bit Mask 1 .. Bit Mask N (N * Byte)

The Bit Mask fields describe the supported fan modes by the thermostat. The bit 0 in Bit Mask 1 field used to indicate whether Fan Mode = 0 (Auto / Auto Low) is supported or not. The fan mode is supported if the bit is 1 and the opposite if 0. The bit 1 in Bit Mask 1 field used by Fan Mode = 1 (On / On Low) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported fan mode. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.99 Thermostat Fan Mode Command Class, Version 2

The Thermostat Fan Mode Command Class, version 2 is used for the HVAC's systems manual fan.

The commands not mentioned here will remain the same as specified for Thermostat Fan Mode Command Class (Version 1).

4.99.1 Thermostat Fan Mode Set Command

The Thermostat Fan Mode Set Command is used to set the fan mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_SET							
Off	Reserved			Fan Mode			

Off (1 bit)

The "Off bit" set to "1" will switch the fan fully OFF regardless of what fan mode has been set. In order to activate a fan mode the "Off bit" MUST be set to "0".

Fan Mode (4 bits)

Fan Mode	Description
0	Auto/Auto Low – Will turn the manual fan operation off unless turned on by the furnace or AC. Lower speed is selected in case it is a two-speed fan.
1	Low – Will turn the manual fan operation on. Low speed is selected.
2	Auto High – Will turn the manual fan operation off unless turned on by the furnace or AC. High speed is selected in case it is a two-speed fan.
3	High – Will turn the manual fan operation on. High speed is selected.
4 (Version 2)	Auto Medium – Will turn the manual fan operation off unless turned on by the furnace or AC. Medium speed is selected in case it is a three-speed fan.
5 (Version 2)	Medium – Will turn the manual fan operation on. Medium speed is selected.
6-15	Reserved

Reserved (3 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.99.2   Thermostat Fan Mode Report Command

The Thermostat Fan Mode Report Command is used to report the fan mode in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_REPORT							
Reserved				Fan Mode			

Fan Mode (4 bits)

Refer to description under the Thermostat Fan Mode Set Command.

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

4.100 Thermostat Fan Mode Command Class, Version 3

The Thermostat Fan Mode Command Class, version 3 is used for the HVAC's systems manual fan.

4.100.1 Thermostat Fan Mode Set Command

The Thermostat Fan Mode Set Command is used to set the fan mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_SET							
Off	Reserved			Fan Mode			

Reserved (3 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Off (1 bit)

The "Off bit" set to "1" will switch the fan fully OFF. In order to activate a fan mode the "Off bit" MUST be set to "0". However, for some applications it is critical that the fan is ON in certain modes. In this case, the application can decide to ignore the Off bit.

CONFIDENTIAL

Fan Mode (4 bit)

Fan Mode	Description
0	Auto/Auto Low – Will turn the manual fan operation off unless turned on by the furnace or AC. Lower speed is selected in case it is a two-speed fan.
1	Low – Will turn the manual fan operation on. Low speed is selected.
2	Auto High – Will turn the manual fan operation off unless turned on by the furnace or AC. High speed is selected in case it is a two-speed fan.
3	High – Will turn the manual fan operation on. High speed is selected.
4 (Version 2)	Auto Medium – Will turn the manual fan operation off unless turned on by the furnace or AC. Medium speed is selected in case it is a three-speed fan.
5 (Version 2)	Medium – Will turn the manual fan operation on. Medium speed is selected.
6 (Version 3)	Circulation - Will turn the manual fan operation off unless turned on by the circulation algorithms – This function is in the process of being implemented in the Thermostat Fan Mode Command Class
7 (Version 3)	Humidity Circulation – Will turn the manual fan operation off unless enabled by the Humidity Circulation algorithms - This function is in the process of being implemented in the Thermostat Fan Mode Command Class
8-15	Reserved

4.100.2 Thermostat Fan Mode Get Command

The Thermostat Fan Mode Get Command is used to request the fan mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_GET							

CONFIDENTIAL

4.100.3 Thermostat Fan Mode Report Command

The Thermostat Fan Mode Report Command is used to report the fan mode in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_MODE							
Command = THERMOSTAT_FAN_MODE_REPORT							
Off	Reserved			Fan Mode			

Fan Mode (4 bit)

Refer to description under the Thermostat Fan Mode Set Command.

Reserved (3 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Off (1 bit)

The "Off bit" set to "1" indicates that the fan is fully OFF. The "Off bit" set to "0" indicates that it is possible to change between Fan Modes.

For some applications, it is critical that the fan is ON in certain modes. In this case, the application can decide to ignore the Off bit. This means that the Off bit in the Report MUST always be set to "0"

CONFIDENTIAL

4.101 Thermostat Fan State Command Class, version 1

The Thermostat Fan State Command Class used to obtain the fan operating state of the thermostat.

4.101.1 Thermostat Fan State Get Command

The Thermostat Fan State Get Command is used to request the fan operating state from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_STATE							
Command = THERMOSTAT_FAN_STATE_GET							

4.101.2 Thermostat Fan State Report Command

The Thermostat Fan State Report Command used to report the fan operating state of the device. It can be send either unsolicited or requested by the Thermostat Fan State Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_FAN_STATE							
Command = THERMOSTAT_FAN_STATE_REPORT							
Reserved				Fan Operating State			

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Fan Operating State (4 bits)

The fan operating state identifier can be set to the following values:

Fan Operating State	Description
0	Idle
1	Running / Running Low - Lower speed is selected in case it is a two-speed fan.
2	Running High - High speed is selected in case it is a two-speed fan.

This list MAY evolve in the future. If a fan operating state is not supported then it SHOULD be ignored.

4.102 Thermostat Mode Command Class, version 1-2

The Thermostat Mode Command Class used to control a thermostat. These Commands allow applications to set and get the thermostat parameters. Version 2 extends the available number of modes.

NOTE: A device supporting the Thermostat Mode Command Class cannot support Auto and Auto Changeover mode simultaneously. Devices controlling a device supporting the Thermostat Mode Command Class MUST be able to control both modes to ensure interoperability.

4.102.1 Thermostat Mode Set Command

The Thermostat Mode Set Command used to set the wanted mode in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE							
Command = THERMOSTAT_MODE_SET							
Reserved			Mode				

CONFIDENTIAL

Mode (5 bits)

The thermostat mode identifier can be set to the following values:

Mode	Description	Version
0	Off – System is off. No heating and cooling will come on. Version 1.	1
1	Heat – Only heating will occur.	1
2	Cool – Only cooling will occur.	1
3	Auto – Heating or cooling will come on according to the heating and cooling setpoints. The system will automatically switch between heating and cooling when the temperature exceeds the setpoints.	1
4	Auxiliary/Emergency Heat – A heat pump (especially air exchange types) are not efficient when the outside temperature is below 35 degrees Fahrenheit (approaching 0 degrees centigrade). Thus, the thermostat MAY be put into Aux heat mode simply to use a more efficient secondary heat source when there are no failures of the compressor or heat pump unit itself.	1
5	Resume – The system will resume from last active mode.	1
6	Fan Only – Only cycle fan to circulate air.	1
7	Furnace – Only cycle fan to circulate air.	1
8	Dry Air – The system will cycle cooling in relation to the room and set point temperatures in order to remove moisture from ambient (Dehumidification).	1
9	Moist Air – (Humidification).	1
10	Auto Changeover – Heating or cooling will come on according to the auto changeover setpoint.	1
11	Energy Save Heat – Energy Save Mode Heating will occur (usually lower than normal setpoint).	2
12	Energy Save Cool – Energy Save Mode Cooling will occur (usually higher than normal setpoint).	2
13	AWAY – special Heating Mode, i.e. preventing water from freezing in forced water systems.	2

This list MAY evolve in the future. If a mode is not supported then it SHOULD be ignored.

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

4.102.2 Thermostat Mode Get Command

The Thermostat Mode Get Command is used to request the current mode from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE							
Command = THERMOSTAT_MODE_GET							

4.102.3 Thermostat Mode Report Command

The Thermostat Mode Report Command used to report the mode from the device. It can be send either unsolicited or requested by the Mode Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE							
Command = THERMOSTAT_MODE_REPORT							
Reserved			Mode				

Mode (5 bits)

Refer to description under the Thermostat Mode Set Command.

Reserved (3 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

4.102.4 Thermostat Mode Supported Get Command

The Thermostat Mode Supported Get Command is used to request the supported modes from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE							
Command = THERMOSTAT_MODE_SUPPORTED_GET							

CONFIDENTIAL

4.102.5 Thermostat Mode Supported Report Command

The Thermostat Mode Supported Report Command used to report the supported modes from the device. It can be send either unsolicited or requested by the Mode Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_MODE							
Command = THERMOSTAT_MODE_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported modes by the device. The bit 0 in Bit Mask 1 field used to indicate whether Mode = 0 (Off) is supported or not. The mode is supported if the bit is 1 and the opposite if 0. The bit 1 in Bit Mask 1 field used by Mode = 1 (Heat) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported mode. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

For example if thermostat supports Heat, Cool, Energy Save Heat and Energy Save Cool bit mask would be 0x06 and 0x18 respectively:

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

Bit Mask Byte 1 = 0x06

Heat

Cool

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

Bit Mask Byte 2 = 0x18

Energy Save Heat

Energy Save Cool

CONFIDENTIAL
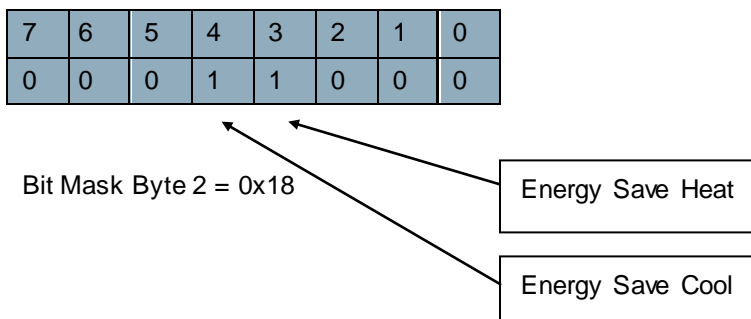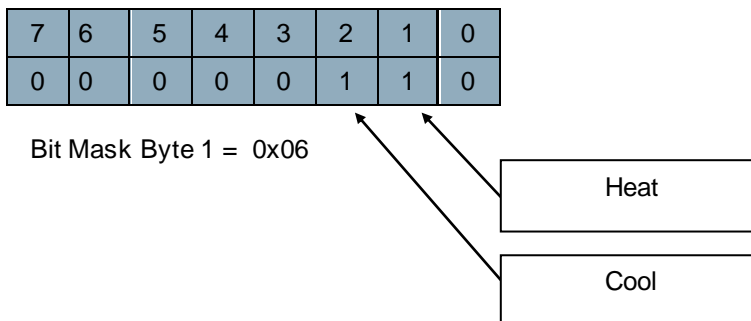
4.103 Thermostat Operating State Command Class, version 1

The Thermostat Operating State Command Class used to obtain the operating state of the thermostat.

4.103.1 Thermostat Operating State Get Command

The Thermostat Operating State Get Command is used to request the operating state from the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_GET							

CONFIDENTIAL

4.103.2 Thermostat Operating State Report Command

The Thermostat Operating State Report used to report the operating state of the device. It can be send either unsolicited or requested by the Operating State Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_REPORT							
Reserved				Operating State			

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Operating State (8 bits)

The thermostat operating state identifier can be set to the following values:

Operating State	Description
0	Idle
1	Heating
2	Cooling
3	Fan Only
4	Pending Heat. Short cycle prevention feature used in heat pump applications to protect the compressor.
5	Pending Cool. Short cycle prevention feature used in heat pump applications to protect the compressor.
6	Vent/Economizer.

This list MAY evolve in the future.

4.104 Thermostat Operating State Command Class, version 2

The Thermostat Operating State Command Class used to obtain the operating state of the thermostat as well as logged operating runtime times of thermostat.

4.104.1 Thermostat Operating State Get (same as version 1)

The Thermostat Operating State Get Command gets the operating state of the thermostat.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_GET							

4.104.2 Thermostat Operating State Report

The Thermostat Operating State Report used to report the operating state of the device. It can be send either unsolicited or requested by the Operating State Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_REPORT							
Reserved				Operating State			

Reserved (4 bits)

The reserved field is for future use. The implementation shall zero these fields and shall make no assumptions on the values of these fields nor perform processing based on their content.

CONFIDENTIAL

Operating State (8 bits)

The thermostat operating state identifier can be set to the following values:

Operating State	Description
0	Idle
1	Heating
2	Cooling
3	Fan Only
4	Pending Heat. Short cycle prevention feature used in heat pump applications to protect the compressor.
5	Pending Cool. Short cycle prevention feature used in heat pump applications to protect the compressor.
6	Vent/Economizer.
7	Aux Heating (version 2)
8	2nd Stage Heating (version 2)
9	2nd Stage Cooling (version 2)
10	2nd Stage Aux Heat (version 2)
11	3rd Stage Aux Heat (version 2)

This list MAY evolve in the future.

4.104.3 Thermostat Operating State Logging Supported Get

The Thermostat Operating State Logging Supported Get Command is used to request the operating state logging supported by the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_LOGGING_SUPPORTED_GET							

CONFIDENTIAL

4.104.4   Thermostat Operating State Logging Supported Report

The Thermostat Operating State Logging Supported Report Command is used to report the operating state logging supported by the device. It can be sent unsolicited or requested by the Operating State Logging Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_LOGGING_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask 1 … Bit Mask N (N * Bytes)

The Bit Mask fields describe the operating state logging supported by the device. The bit 1 in Bit Mask 1 field is used to indicate whether Operating State = 1 (Heating) log is supported or not. The operating state log is supported if the bit is 1 and the opposite of 0. The bit 2 in Bit Mask 1 field used by Operating State = 2 (Cooling) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported operating state log. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

4.104.5   Thermostat Operating State Logging Get

The Thermostat Operating State Logging Get Command is used to request the operating state logging supported by the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_LOGGING_GET							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask 1 … Bit Mask N (N * Bytes)

The Bit Mask fields describe the operating state log types to be requested. The bit 1 in Bit Mask 1 field is used to indicate whether Operating State = 1 (Heating) log is requested. The operating state log is requested if the bit is 1 and the opposite of 0. The bit 2 in Bit Mask 1 field used by Operating State = 2 (Cooling) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last requested operating state log type. The number of Bit Mask fields transmitted can be determined from the length field in the frame.

CONFIDENTIAL

4.104.6 Thermostat Operating State Logging Report

The Thermostat Operating State Logging Report Command is used to report the operating state logged for requested operating states.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_OPERATING_STATE							
Command = THERMOSTAT_OPERATING_STATE_LOGGING_REPORT							
Reports to Follow							
Reserved				Operating State Log Type 1			
Usage Today (Hours)							
Usage Today (Minutes)							
Usage Yesterday (Hours)							
Usage Yesterday (Minutes)							
…							
Reserved				Operating State Log Type N			
Usage Today (Hours)							
Usage Today (Minutes)							
Usage Yesterday (Hours)							
Usage Yesterday (Minutes)							

Reports to Follow (8 bit)

This value indicates how many report frames left before transferring all of the requested thermostat operating state logs.

Operating State Log Type 1 … Operating State Log Type N (4 bit)

The Operating State Log Type indicates the operating state type to be requested.

Usage Today Hours (8 bit)

The number of hours (00:24) the thermostat has been in the indicated operating state since 12:00 am of the current day.

Usage Today Minutes (8 bit)

The number of minutes (00-59) the thermostat has been in the indicated operating state since 12:00 am of the current day.

Usage Yesterday Hours (8 bit)

The number of hours (00:24) the thermostat had been in the indicated operating state between 12:00 am and 11:59pm of the previous day.

CONFIDENTIAL

Usage Yesterday Hours (8 bit)

The number of minutes (00-59) the thermostat had been in the indicated operating state between 12:00 am and 11:59pm of the previous day.

4.105 Thermostat Setback Command Class, version 1

The Thermostat Setback Command Class used to change the current state of a non-schedule setback thermostat.

4.105.1 Thermostat Setback Set Command

The Thermostat Setback Set Command used to set the state of the thermostat.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETBACK							
Command = THERMOSTAT_SETBACK_SET							
Reserved						Setback Type	
Setback State							

Setback Type (2bits)

The setback type field can assume the following values:

Binary	Decimal	Description
0b00	0	No override
0b01	1	Temporary override
0b10	2	Permanent override
0b11	3	Reserved

Note: The temporary override provides an opportunity to implement a timer or equivalent in the device. A temporary override will, if a timer is implemented, be terminated by the timer, if no timer is implemented the temporary override MUST act as permanent override. If the temporary override is implemented it MUST be documented in the user's manual.

CONFIDENTIAL

Setback State (8 bits)

The Setback State can assume the following values:

Setback State		Description
Hexadecimal	**Decimal**	
0x80 … 0xFF 0x00 0x01 … 0x78	-128 … -1 0 1 … 120	The setback in 1/10 degrees (Kelvin) Example: 0 = 0 degrees setback 1 = 0.1 degrees is added to the setpoint 2 = 0.2 degrees is added to the setpoint -1 = 0.1 degrees is subtracted from the setpoint -2 = 0.2 degrees is subtracted from the setpoint
0x79	121	Frost Protection
0x7A	122	Energy Saving Mode
0x7B – 0x7E	123 – 126	Reserved
0x7F	127	Unused State

When converting between Celsius and Fahrenheit proper rounding MUST be applied with at least two decimals in the internal calculations of a device to avoid rounding errors.

Note: The implementation of Energy Saving Mode is manufacturer specific, and MUST be documented in the User's Manual.
If the device is set to an unreachable state, the device MUST set itself to the state which is closest possible to the requested.

4.105.2 Thermostat Setback Get Command

The Thermostat Setback Get Command is used to request the current state of the thermostat.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETBACK							
Command = THERMOSTAT_SETBACK_GET							

CONFIDENTIAL

4.105.3 Thermostat Setback Report Command

The Thermostat Setback Report Command used to report the current state of the thermostat.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETBACK							
Command = THERMOSTAT_SETBACK_REPORT							
Reserved						Setback Type	
Setback State							

Note: If the device is set to an unreachable state, the device MUST report the state which is closest possible to the requested.

Setback Type (2bits)

Refer to description under the Thermostat Setback Set Command

Setback State (8 bits)

Refer to description under the Thermostat Setback Set Command.

CONFIDENTIAL

4.106 Thermostat Setpoint Command Class, version 1-2

The Thermostat Setpoint Command Class used for setpoint handling. Version 2 extends the available number of setpoint types.

4.106.1 Thermostat Setpoint Set Command

The Thermostat Setpoint Set Command used to set the setpoint in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT							
Command = THERMOSTAT_SETPOINT_SET							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
Value 2							
..							
Value n							

CONFIDENTIAL

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Setpoint Type (4 bits)

The setpoint type number specifies the setpoint to be set in the thermostat. Currently the following setpoint types are defined:

Setpoint Type	Description	Version
1	Heating #1	1
2	Cooling #1	1
7	Furnace	1
8	Dry Air	1
9	Moist Air	1
10	Auto changeover	1
11	Energy Save Heating	2
12	Energy Save Cooling	2
13	Away Heating	2

This list MAY evolve in the future. In case a setpoint type is not supported then it SHOULD be ignored.

Precision (3 bits)

The precision field describes what the precision of the setpoint value is. The number indicates the number of decimals. The decimal value 1025 with precision 2 is therefore equal to 10.25.

Scale (2 bits)

The scale field indicates the temperature scale used, 0 indicate the use of the Celsius temperature scale and 1 indicates use of the Fahrenheit scale.

Size (3 bits)

The size field indicates the number of bytes used for the setpoint value. This field can take values from 1 (001b), 2 (010b) or 4 (100b).

Value (variable)

The value is a signed field. The field can be 1, 2 or 4 bytes in size. The first byte is the most significant byte. The table below shows signed decimal values together with their hexadecimal equivalents.

Signed 1 byte decimal value	Hexadecimal	Signed 2 bytes decimal value	Hexadecimal
127	0x7F	32767	0x7FFF
25	0x19	1025	0x0401
2	0x02	2	0x0002
1	0x01	1	0x0001
0	0x00	0	0x0000
-1	0xFF	-1	0xFFFF
-2	0xFE	-2	0xFFFE
-25	0xE7	-1025	0xFBFF
-128	0x80	-32768	0x8000

4.106.2 Thermostat Setpoint Get Command

The Thermostat Setpoint Get Command is used to request a given setpoint type in a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT							
Command = THERMOSTAT_SETPOINT_GET							
Reserved				Setpoint Type			

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Setpoint Type (4 bits)

Refer to description under the Thermostat Setpoint Set Command.

CONFIDENTIAL

4.106.3 Thermostat Setpoint Report Command

The Thermostat Setpoint Report Command used to report the value of the setpoint type in a device.

| 7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT							
Command = THERMOSTAT_SETPOINT_REPORT							
Reserved				Setpoint Type			
Precision			Scale		Size		
Value 1							
Value 2							
..							
Value n							

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Setpoint Type (4 bits)

Refer to description under the Thermostat Setpoint Set Command.

Precision (3 bits)

Refer to description under the Thermostat Setpoint Set Command.

Scale (2 bits)

Refer to description under the Thermostat Setpoint Set Command.

Size (3 bits)

Refer to description under the Thermostat Setpoint Set Command.

Value (variable)

Refer to description under the Thermostat Setpoint Set Command.

4.106.4 Thermostat Setpoint Supported Get Command

The Thermostat Setpoint Supported Get Command is used to request the setpoint types supported by the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT							
Command = THERMOSTAT_SETPOINT_SUPPORTED_GET							

4.106.5 Thermostat Setpoint Supported Report Command

The Thermostat Setpoint Supported Report Command used to report the setpoint types supported by the device. It can be send either unsolicited or requested by the Setpoint Supported Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_THERMOSTAT_SETPOINT							
Command = THERMOSTAT_SETPOINT_SUPPORTED_REPORT							
Bit Mask 1							
…							
Bit Mask N							

Bit Mask 1 .. Bit Mask N (N * Bytes)

The Bit Mask fields describe the supported setpoint types by the device. The bit 1 in Bit Mask 1 field used to indicate whether Setpoint Type = 1 (Heating #1) is supported or not. The setpoint type is supported if the bit is 1 and the opposite if 0. The bit 2 in Bit Mask 1 field used by Setpoint Type = 2 (Cooling #1). The bit 3 in Bit Mask 1 field used by Setpoint Type = 7 (Furnace) and so forth. It is only necessary to send the Bit Mask fields from 1 and up to the one indicating the last supported setpoint type. The number of Bit Mask fields transmitted can be determined from the length field in the frame. Notice that bit 0 in Bit Mask 1 field is not allocated to any Setpoint Type and MUST therefore be zeroed.

CONFIDENTIAL

4.107 Time Command Class, version 1

This Time Command Class version1 used to read date and time from a device in a Z-Wave network.

Notice that the former Time Command Class version 1 (Revision 4 of this document) is discontinued and replaced by a new one.

4.107.1 Time Get Command

The Time Get Command is used to request current time. Be aware that the communication overhead can be significant in case routing is necessary.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME							
Command = TIME_GET							

4.107.2 Time Report Command

The Time Report Command returns current time. The Time Report command can be send unsolicited or requested by a Time Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME							
Command = TIME_REPORT							
RTC failure	Reserved		Hour Local Time				
Minute Local Time							
Second Local Time							

RTC failure (1 bit)

Many RTC chips have a stop bit indicating if the oscillator has been stopped. The RTC failure bit used to indicate to the receiving unit that the RTC in the device has been stopped and that the time might be inaccurate.
If the sending/receiving device does not support this feature it MUST ignore this bit. As a result of this, the bit can only be used to indicate that the time might be inaccurate and not to inform a device that it MUST ignore the time.

Reserved (2 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Hour Local Time (8 bits)

Specify the number of complete hours that have passed since midnight (00-23) in local time.

Minute Local Time (8 bits)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in local time.

Second Local Time (8 bits)

Specify the number of complete seconds since the start of the minute (00-59) in local time. The value 60 used to keep UTC from wandering away is not supported.

4.107.3   Date Get Command

The Date Get Command is used to request current date adjusted according to the local time zone and daylight savings time. Be aware that the communication overhead can be significant in case routing is necessary.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME							
Command = DATE_GET							

CONFIDENTIAL

4.107.4 Date Report Command

The Date Report Command returns current date adjusted according to the local time zone and daylight savings time. The Date Report command can be send unsolicited or requested by a Date Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME							
Command = DATE_REPORT							
Year 1							
Year 2							
Month							
Day							

Year 1..2 (16 bits)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Example 2007: Year1= 00000111, Year2=11010111

Month (8 bits)

Specify the month of the year between 01 (January) and 12 (December).

Day (8 bits)

Specify the day of the month between 01 and 31.

CONFIDENTIAL

4.108 Time Command Class, version 2

The Time Command Class version 2 enables setting time zone offset and daylight savings parameters. The data formats are based on the International Standard ISO 8601.

The Commands not mentioned here will remain the same as in version 1.

4.108.1 Time Offset Get Command

The Time Offset Get Command is used to request time zone offset and daylight savings parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME							
Command = TIME_OFFSET_GET							

CONFIDENTIAL

4.108.2 Time Offset Set Command

The Time Offset Set Command used to set time zone offset and daylight savings parameters to achieve local time depending on the clock source.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME							
Command = TIME_OFFSET_SET							
Sign TZO	Hour TZO						
Minute TZO							
Sign Offset DST	Minute Offset DST						
Month Start DST							
Day Start DST							
Hour Start DST							
Month End DST							
Day End DST							
Hour End DST							

Sign TZO (1 bit)

Plus (0) or minus (1) sign to indicate a positive or negative offset from UTC.

Hour TZO (7 bits)

Specify the number of hours that the originating time zone deviates from UTC. Refer to the DST field regarding daylight savings handling.

Minute TZO (7 bits)

Specify the number of minutes that the originating time zone deviates UTC. Refer to the DST field regarding daylight savings handling.

Sign Offset DST (1 bit)

Plus (0) or minus (1) sign to indicate a positive or negative offset from UTC.

Minute Offset DST (7 bits)

Specify the number of complete minutes the time MUST be adjusted when daylight savings mode is enabled.

Month Start DST (8 bits)

Specify the month of the year between 01 (January) and 12 (December) when daylight savings mode is enabled.

CONFIDENTIAL

Day Start DST (8 bits)

Specify the day of the month between 01 and 31 when daylight savings mode is enabled.

Hour Start DST (8 bits)

Specify the number of complete hours that have passed since midnight (00-23) in local time when daylight savings mode is enabled.

Month End DST (8 bits)

Specify the month of the year between 01 (January) and 12 (December) when daylight savings mode is disabled.

Day End DST (8 bits)

Specify the day of the month between 01 and 31 when daylight savings mode is disabled.

Hour End DST (8 bits)

Specify the number of complete hours that have passed since midnight (00-23) in local time when daylight savings mode is disabled.

4.108.3 Time Offset Report Command

The Time Offset Report Command returns time zone offset and daylight savings parameters and can be requested by the Time Offset Get command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME								
Command = TIME_OFFSET_REPORT								
Sign TZO	Hour TZO						
Minute TZO								
Sign Offset DST	Minute Offset DST						
Month Start DST								
Day Start DST								
Hour Start DST								
Month End DST								
Day End DST								
Hour End DST								

Refer to description under the Time Offset Set command.

CONFIDENTIAL

4.109 Time Parameters Command Class, version 1

This Time Parameters Command Class used to set date and time in a device hosting this facility. In case the clock is updated via an external source such as SAT, internet, Rugby/Frankfurt source then omit this command class. Time zone offset and daylight savings can be set in the Time Command Class if necessary. The data formats are based on the International Standard ISO 8601.

4.109.1 Time Parameters Set Command

The Time Parameters Set Command used to set current date and time in Universal Time (UTC). Be aware that the communication overhead can be significant in case routing is necessary. For two nodes within direct range is the communication overhead less than 100 msec.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME_PARAMETERS							
Command = TIME_PARAMETERS_SET							
Year 1							
Year 2							
Month							
Day							
Hour UTC							
Minute UTC							
Second UTC							

Year 1..2 (16 bits)

Specify the year in the usual Gregorian calendar. The first byte (Year 1) is the most significant byte.

Month (8 bits)

Specify the month of the year between 01 (January) and 12 (December).

Day (8 bits)

Specify the day of the month between 01 and 31.

Hour UTC (8 bits)

Specify the number of complete hours that have passed since midnight (00-23) in UTC.

Minute UTC (8 bits)

Specify the number of complete minutes that have passed since the start of the hour (00-59) in UTC. Minutes are measured in Universal Time (UTC).

Second UTC (8 bits)

Specify the number of complete seconds since the start of the minute (00-59) in UTC. Seconds are measured in Universal Time (UTC).

4.109.2 Time Parameters Get Command

The Time Parameters Get Command is used to request date and time parameters.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME_PARAMETERS							
Command = TIME_PARAMETERS_GET							

4.109.3 Time Parameters Report Command

The Time Parameters Report Command can be requested by the Time Parameters Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TIME_PARAMETERS							
Command = TIME_PARAMETERS_REPORT							
Year 1							
Year 2							
Month							
Day							
Hour UTC							
Minute UTC							
Second UTC							

Refer to description under the Time Parameters Set Command.

CONFIDENTIAL

4.110 Transport Service Command Class, version 1

The Transport Service Command Class supports the following features to obtain reliable data transfer of datagrams larger than the Z-Wave frame.

Long commands

IPv6 datagrams MAY span up to 1280 bytes. Adding 33 bytes overhead from security and additional encapsulation, a payload of at least 1313 bytes MUST be supported.
Z-Wave applications requiring transport of even bigger amounts of data, such as firmware update, MUST partition data appropriately in order not to violate the upper limit of 1313 bytes.
Note that IP applications such as FTP and TFTP will not have to take this into consideration as FTP uses TCP and TFTP uses fixed 512 byte UDP segments.

Reliable checksum

The Transport Service MUST implement a 16 bit checksum to protect the Transport Service fragments.

Transparent functionality

The Z-Wave Transport Service MUST operate transparently below the Z-Wave protocol layer and application layers.
The Transport Service MUST automatically determine if messages are to be fragmented depending on the available payload length. While early Z-Wave chips supported 48 bytes payload only (in source routed frames), more recent implementations have introduced support for significantly longer payload; spanning up to 150 bytes or more.

The Transport Service SHOULD NOT use fragmentation encapsulation if all data can be sent in one frame payload.
The API interface function provided to the higher layers MUST still be the SendData function.

Option: SendData call-back: "Working"

Since the transport layer has to send up to 32 frames in full-size message, a calling application MAY experience a considerable delay before an "OK" call-back status is returned.

In addition to "OK" and "Error", applications sending large amounts of data MAY implement support for the OPTIONAL call-back status "Working". The "Working" call-back MAY be sent to an application at the successful completion of transmission of every single Z-Wave frame. Z-Wave source routing via four repeaters takes longer time than direct range transmission.

By mapping a progress indication to the "Working" call-back status, an application progress indicator LED MAY flash at a slower pace during longer frame delays; indicating to the user that data transfer is going to take longer time.
The call-back function SHOULD accept a 4-bit range of status values all mapping to the status interpretation "Working". If supporting 16 values, the values MUST be interpreted as a progress indication where "0000" represents 0% and "1111" represents 94%. The other values each are uniformly distributed with a distance of 6.25% (MAY be rounded).

Meta-data transmission timing

In order not to congest the Z-Wave network, large data transfers MUST leave transmit opportunities for other nodes in the network. If sending a Transport Service datagram longer than a single frame, a node MUST implement a delay between every transmitted frame. The minimum REQUIRED time delay and number of frames before a delay MUST be inserted depends on the actual bit rate.

CONFIDENTIAL

- 40 kbit/s: At least 35 ms if sending more than 1 frame back-to-back
- 100 kbit/s: At least 15 ms if sending more than 2 frames back-to-back

Support various PHY frame lengths

More recent chip generations support longer Z-Wave frame lengths than the 48 bytes used by current 9.6 kbit/s and 40 kbit/s systems. Future systems running at other frequencies MAY introduce longer frames. The Z-Wave Transport Service MUST support all frame lengths offered by the MAC/PHY layer.

Non-requirement: Support for frame re-ordering

The Z-Wave Transport Service is a point-to-point Transport Service; one fragment is acknowledged at the Z-Wave level before the next fragment is transmitted.
Thus, a reassembly solution MAY be implemented with no support for re-ordered frames at the Z-Wave level.

Non-requirement: Session awareness

Transmitting nodes MUST NOT allow other Z-Wave commands to be transmitted to a destination node before transmission of one fragmented message has been completed to that same destination node.
Doing so could potentially delay the transmission causing security timers to time out, rendering the actual security nonce invalid.

A transmitting node MAY transmit a single-frame command to another destination node while transmitting a fragmented datagram.

Fragment-based re-transmission

The fragment reassembly process MAY end up missing a number of fragments in the receive buffer.
In order to fill out the holes in the receive buffer, the receiving node MUST be able to request the retransmission of individual fragments to complement the data already received.
The algorithm MUST be able to handle the case where retransmitted fragments are also lost.

In case no frames are received in response to a fragment retransmission request, the node MUST discard all data previously received.

Transparent error handling

In case an internal SendDataSingleFrame function repeatedly fails to deliver a fragment to a destination node and returns an "Error" callback, the SendData API function MUST give up transmission of all fragments and issue an "Error" callback to the calling application (or higher layers of the protocol).

CONFIDENTIAL

4.110.1 First Fragment Command

The First Fragment Command initiates the transfer of the datagram.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_FIRST_FRAGMENT				datagram_size_1 [10..8]			
datagram_size_2 [7..0]							
Reserved				Sequence no.			
Payload 1							
...							
Payload N							
Checksum 1							
Checksum 2							

Command (5 bits) = "11000"

The command field is reduced to a 5-bit field in order to limit the overhead introduced by the fragment header. Z-Wave Zniffer tools and parsers MUST mask out the LS 3 bits before decoding the command code.

Datagram Size (11 bits)

Indicates the accumulated overall size of the message carried in the fragments.
The unit is 1 byte; allowing the message to be up to 2047 bytes long. Security time-outs lower the upper limit to 1313 bytes; enough to carry a full IPv6 frame with Z-Wave Security encryption.

Reserved (4 bits)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Sequence no. (4 bits)

A sequence number identifying the session. MUST be incremented when sending out a new First Fragment. Initial value SHOULD be random.

Payload (Variable size)

The Payload field of a First Fragment always starts with two bytes carrying a Z-Wave Command Class and Command code. All extended Z-Wave commands MAY be carried using the Transport Service. Typical examples are security encapsulation and 6lowPan encapsulated IP datagrams.

Checksum (16 bits)

CRC-16 checksum calculated by using the CRC-16 polynium implemented in the 400 series Z-Wave Silicon. For operation in previous platforms such as the 300 series Z-Wave Silicon, the CRC-16 polynomium MUST be calculated by a software function.
The following polynomium MUST be used:

CONFIDENTIAL

$P(x) = x^{16}+x^{12}+x^{5}+1$, also known as CRC–CCITT; Refer to FNC10751 for details.

The CRC value MUST cover all fields of the Transport Service Command, except the CRC itself: From the command class to the end of the payload.

4.110.2 Subsequent Fragment Command

The Subsequent Fragment Command follows the First Fragment Command and carries the remaining part of the datagram.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_SUBSEQUENT_FRAGMENT					datagram_size_1 [10..8]		
datagram_size_2 [7..0]							
reserved	Sequence no.				datagram_offset_1 [10..8]		
datagram_offset_2 [7..0]							
Payload 1							
...							
Payload N							
Checksum 1							
Checksum 2							

Command (5 bits) = "11100"

Refer to section 4.110.1

Datagram Size (11 bits)

Refer to section 4.110.1

Reserved (1 bit)

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Sequence no. (4 bits)

Refer to section 4.110.1

Datagram Offset (11 bits)

Indicates the offset to apply to the data carried in this fragment when reassembling the payload in the receive buffer.
The unit is 1 byte.

Payload (Variable size)

The Payload field of a Subsequent Fragment MAY carry any data. The actual contents can only be derived from the data carried in the First Fragment and previous subsequent fragments.

Checksum (16 bits)

Refer to section 4.110.1

4.110.3 Fragment Request Command

The Fragment Request Command is used by a receiving node to indicate to the sender that one or more fragments are needed in order to re-assemble the datagram in the receiving end.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_FRAGMENT_REQUEST					*reserved*		
reserved				Sequence no.			
Number of offsets							
datagram_offset_1 1							
datagram_offset_2 1							
...							
...							
datagram_offset_1 N							
datagram_offset_2 N							

Command (5 bits) = "11001"

Refer to section 4.110.1

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Sequence no. (4 bits)

Refer to section 4.110.1

Number of offsets (8 bits)

Indicates the number of offsets carried in the Fragment Request. Every offset is a 2-byte field carrying an 11-bit value representing the starting offset of the missing fragment.

The value MUST be in the range 1..22.
If more fragments need retransmission, another Fragment Request MUST be issued after receiving all the requested fragments.

Datagram Offset

Refer to section 4.110.2.

4.110.4 Fragment Complete Command

The Fragment Complete Command is returned from the receiving node when all fragments have been correctly received.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_FRAGMENT_COMPLETE					*reserved*		
reserved				Sequence no.			

Command (5 bits) = "11101"

Refer to section 4.110.1

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Sequence no. (4 bits)

Refer to section 4.110.1

4.110.5 Fragment Wait Command

The Fragment Wait Command MAY be by a receiving node which is already receiving fragments from another node.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_TRANSPORT_SERVICE							
Command = COMMAND_FRAGMENT_WAIT					*reserved*		
pending_fragments							

Command (5 bits) = "11110"

Refer to section 4.110.1

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Pending Fragments (8 bits)

The value indicates the number of fragments still not received by the receiving node. Since delivery time of a fragment depends heavily on path length (direct vs. 4 hop routing), the value does not represent an absolute time. Furthermore, the node currently sending MAY initiate another transfer immediately after the completion of the actual fragmented message.

4.110.6 Managing fragment transfer and handling exceptions

The Transport Service uses the Transport Service Command Class to perform the transfer of message fragments. The following sections present the mechanisms for handling fragments and exceptions such as missing fragments.
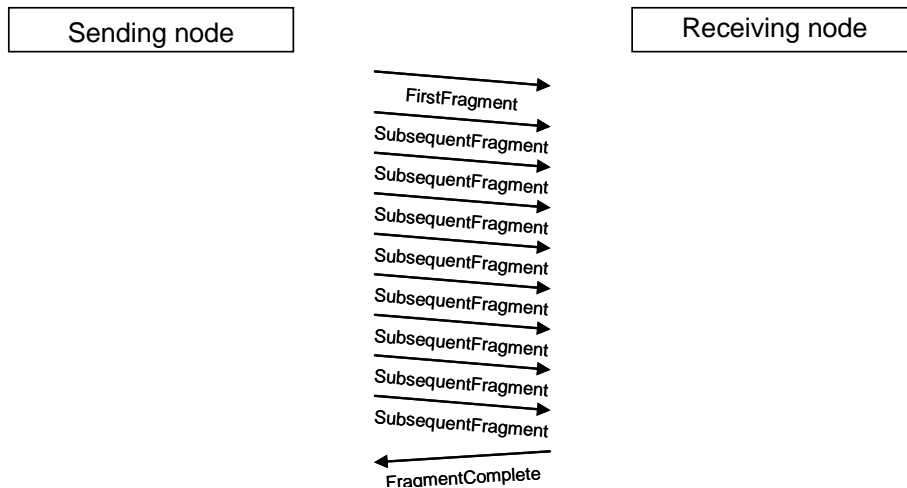
4.110.6.1 Creating the Fragment Tracking List (receiver node)

A receiving node MUST maintain a state variable making the node able to determine that a new fragment transfer has started. The state variable MUST associate to the actual srcNodeId in order to filter out fragments from other sources.
In the normal case a new fragment transfer starts with the reception of a First Fragment message. The receiving node MUST either send a Fragment Wait or open a new session when a Subsequent Fragment is received as the first fragment.

Note: Both fragment types carry the total datagram size. This allows a receiving node to correctly determine the size of the REQUIRED receive buffer as well as the number of entries in the associated Fragment Tracking List.

4.110.6.2 Avoiding concurrent message transfers

It MUST be ensured that a message transfer is not delayed significantly. This could cause a security nonce time-out; rendering the entire received message useless.
Concurrent message transfers SHOULD also be avoided as they would require multiple receive buffers in the receiving node and network bandwidth could be exhausted.

A receiving node MUST return a Fragment Wait to any other node sending a singlecast fragment to the receiving node if the node is already in the process of receiving a fragmented message, except where sec. 4.110.6.8 requires a silent drop. A receiving node MUST silently drop a received broadcast fragment if the node is already in the process of receiving a fragmented message.

A sending node MUST stop sending fragments for a period of minimum 10 ms if receiving a Fragment Wait. When resuming transmission, the sending node MUST restart fragment transmission with the first fragment.

A sending node SHOULD use the pending_fragments value carried in a Fragment Wait command to estimate the proper time for a retry.

CONFIDENTIAL

4.110.6.3   Fragment Rx Timer (receiver node)

A receiving node MUST maintain a timer for monitoring the rate of incoming fragments. The timer MAY be implemented as a software timer with limited precision.

The receiving node MUST (re-)start the Fragment Rx Timer for every fragment received.
The receiving node MUST disable the Fragment Rx Timer when all fragments have been received correctly.

The Fragment Rx Timer interval MUST be:

- 40 kbit/s: at least 800 ms
- 100 kbit/s: at least 400 ms

If receiving a Fragment Rx Timer event, the receiving node MUST evaluate the Fragment Tracking List; Refer to section 4.110.6.4.

4.110.6.4   Requesting missing fragments (receiver node)

If receiving a Fragment Rx Timer event, the receiving node MUST evaluate the Fragment Tracking List. If the message is broadcasted, the node MUST drop the partially received message and MUST NOT send a Fragment Request. If the message is singlecasted, the receiving node MUST send a Fragment Request command (4.110.3) reporting all missing fragments to the sending node. Fragments MUST be identified by the offset needed for reconstruction of the receive buffer.
A receiving node MUST NOT specify more than 22 fragment offsets in one Fragment Request command. A limit of 22 fragment offsets ensures that the command can be carried in a 48-byte Z-Wave frame.

When requesting missing fragments, the last fragment received MAY be another fragment than the last fragment of the complete Fragment Tracking List. The receiving node MUST implement a method for determining when to re-evaluate the fragment tracking list.

The following approach is proposed:
A receiving node requests the (last 22) missing fragments of the fragment tracking list. When receiving a fragment matching the last missing fragment of the Fragment Tracking List, it is time to re-evaluate the Fragment Tracking List. If more fragments are missing, the process is repeated.

If the (relative) last fragment is lost during the process of requesting missing fragments, the Fragment Rx Timer mechanism will make sure the Fragment Tracking List is re-evaluated and the process is re-started.

4.110.6.5   Completing the transfer (receiver node)

Receiving the last fragment of a message (offset value points to last position in fragment tracking list) MUST cause the receiving node to evaluate its fragment tracking list. As a resulting action, the receiving node MUST send out either a Fragment Request or a Fragment Complete message.

The receiving node MUST send a Fragment Complete to the sending node when all fragments have been received correctly when the received message was broadcasted.

The receiver MUST NOT send Fragment Complete when receiving a broadcast message.

4.110.6.6   Fragment Complete Timer (sender node)

A sending node MUST maintain a timer for monitoring the completion of the message transfer. The timer MAY be implemented as a software timer with limited precision.

CONFIDENTIAL

The sending node MUST start the Fragment Complete Timer when the last fragment has been transmitted.
The sending node MUST disable the Fragment Complete Timer if receiving a Fragment Request from the receiving node.
The sending node MUST re-start the Fragment Complete Timer when the last fragment of a Fragment Request has been transmitted.
The sending node MUST disable the Fragment Complete Timer if receiving a Fragment Complete from the receiving node.

The Fragment Complete Timer interval MUST be:

- 40 kbit/s: at least 1000 ms
- 100 kbit/s: at least 500 ms

Refer to 4.110.6.7 for handling of Fragment Complete Timer events.

4.110.6.7 Completing the transfer (sender node)

If receiving a Fragment Complete Timer event, the sending node MUST re-send the last fragment which was just transmitted. At the same time the sending node MUST re-start the Fragment Complete Timer.

If transmission of this last fragment fails or another Fragment Complete Timer event is received, an error code MUST be returned to the calling application and the Fragment Complete Timer MUST be stopped.

4.110.6.8 Sequence numbers

A transmitter MUST increment its sequence number and open a new session when it sends out a First Fragment. When a receiver accepts a First Fragment, it MUST open a session with the sequence number contained therein.

Nodes MUST silently drop Fragment Request and Command Complete messages if the sequence number does not match the current session.

4.110.6.9 Tie-breaking

A receiving node SHOULD perform a tie-break check before dropping an incoming First Fragment. If the check is positive, the receiving node SHOULD instead accept the incoming fragment and abort the ongoing transmission. The tie-break check is positive if the following three conditions are all met:

1. The receiving node is currently transmitting a message.

2. The recipient of the message being transmitted is also the originator of the received frame

3. The receiving node has a lower node ID than the originator

4.110.7 Example Frame flows

The first section presents use cases that assisted in identifying requirements. The use cases make use of commands and other features which are presented in following sections but at a high level which allows the reader to get the big picture.

CONFIDENTIAL

4.110.7.1   As things SHOULD always work – the default case

- Call API function SendData with 120 byte frame
    - SendData sends FirstFragment (offset 0)
    - FirstFragment received with valid 8bit chksum + valid 16bit CRC
        - Receiver creates tracking list for fragments; fragment 0 is marked
        - Receiver starts fragment rx timer
    - SendData sends SubsequentFragment (offset 41)
    - SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
        - Receiver updates tracking list for fragments; fragment 1 is marked
        - Receiver (re-)starts fragment rx timer
    - SendData sends SubsequentFragment (offset 82)
        - Sender starts fragment_complete tx timer
    - SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
        - Offset indicates that this was the last fragment
        - Receiver checks tracking list for missing fragments; none found
    - Receiver sends FragmentComplete(OK)
    - FragmentComplete(OK) received with valid 8bit chksum + valid 16bit CRC
- Transmitter returns "OK" callback to calling application

4.110.7.2   Losing first fragment of a long message

- Call API function SendData with 120 byte frame
    - SendData sends FirstFragment (offset 0)
    - FirstFragment received with valid 8bit chksum + BUT INVALID 16bit CRC
        - Payload is ignored
    - SendData sends SubsequentFragment (offset 41)
        - Receiver Sends Fragment Wait because no session is open.
    - SendData performs wait and restarts transmission from FirstFragment.
    - FirstFragment received with valid 8bit chksum + valid 16bit CRC
        - Receiver creates tracking list for fragments; fragment 0 is marked
        - Receiver starts fragment rx timer
    - SendData sends SubsequentFragment (offset 41)
    - SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
        - Receiver updates tracking list for fragments; fragment 1 is marked
        - Receiver (re-)starts fragment rx timer
    - SendData sends SubsequentFragment (offset 82)
        - Sender starts fragment_complete tx timer
    - SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
        - Offset indicates that this was the last fragment
        - Receiver checks tracking list for missing fragments; fragment 0 is missing
    - Receiver sends FragmentComplete(OK)
    - FragmentComplete(OK) received with valid 8bit chksum + valid 16bit CRC
- Transmitter returns "OK" callback to calling application

4.110.7.3   Losing subsequent fragment

- Call API function SendData with 120 byte frame
    - SendData sends FirstFragment (offset 0)
    - FirstFragment received with valid 8bit chksum + valid 16bit CRC
        - Receiver creates tracking list for fragments; fragment 0 is marked
        - Receiver starts fragment rx timer
    - SendData sends SubsequentFragment (offset 41)
    - SubsequentFragment received with valid 8bit chksum + BUT INVALID 16bit CRC
        - Payload is ignored

CONFIDENTIAL

- SendData sends SubsequentFragment (offset 82)
  - □ Sender starts fragment_complete tx timer
- SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
  - □ Offset indicates that this was the last fragment
  - □ Receiver checks tracking list for missing fragments; fragment 1 is missing
- Receiver sends FragmentRequest(offset 41)
- FragmentRequest(offset 41) received with valid 8bit chksum + valid 16bit CRC
- SendData sends SubsequentFragment (offset 41)
- SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
  - □ Receiver updates tracking list for fragments; fragment 1 is marked
  - □ Receiver checks tracking list for missing fragments; none found
  - □ Receiver clears fragment rx timer
- Receiver sends FragmentComplete(OK)
- FragmentComplete(OK) received with valid 8bit chksum + valid 16bit CRC
- Transmitter returns "OK" callback to calling application

4.110.7.4 Losing last fragment

- Call API function SendData with 120 byte frame
  - SendData sends FirstFragment (offset 0)
  - FirstFragment received with valid 8bit chksum + valid 16bit CRC
    - □ Receiver creates tracking list for fragments; fragment 0 is marked
    - □ Receiver starts fragment rx timer
  - SendData sends SubsequentFragment (offset 41)
  - SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
    - □ Receiver updates tracking list for fragments; fragment 1 is marked
    - □ Receiver (re-)starts fragment rx timer
  - SendData sends (last) SubsequentFragment (offset 82)
    - □ Sender starts fragment_complete tx timer
  - SubsequentFragment received with valid 8bit chksum + BUT INVALID 16bit CRC
    - □ Payload is ignored
  - Fragment rx timer event
    - □ Receiver checks tracking list for missing fragments; fragment 2 (and MAY be more) are missing
    - □ Receiver sendsFragmentRequest(offset 82)
      - Start a timer to wait for the SubsequentFragment frame
      - If timer event occurs, bail out: Discard all received fragments and return to idle (sender MAY be down already?)
  - FragmentRequest(offset 82) received with valid 8bit chksum + valid 16bit CRC
  - SendData sends SubsequentFragment (offset 82)
  - SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
    - □ Receiver updates tracking list for fragments; fragment 2 is marked
    - □ Receiver checks tracking list for missing fragments; none found
  - Receiver sends FragmentComplete(OK)
  - FragmentComplete(OK) received with valid 8bit chksum + valid 16bit CRC
    - □ Sender stops fragment_complete tx timer
- Transmitter returns "OK" callback to calling application

4.110.7.5 Losing FragmentComplete

- Call API function SendData with 120 byte frame
  - SendData sends FirstFragment (offset 0)
  - FirstFragment received with valid 8bit chksum + valid 16bit CRC
    - □ Receiver creates tracking list for fragments; fragment 0 is marked
    - □ Receiver starts fragment rx timer

- SendData sends SubsequentFragment (offset 41)
- SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
  - Receiver updates tracking list for fragments; fragment 1 is marked
  - Receiver (re-)starts fragment rx timer
- SendData sends SubsequentFragment (offset 82)
  - Sender starts fragment_complete timer
- SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
  - Sender starts fragment_complete tx timer
  - Offset indicates that this was the last fragment
  - Receiver checks tracking list for missing fragments; none found
- Receiver sends FragmentComplete(OK)
- FragmentComplete(OK) received w. valid 8bit chksum + BUT INVALID 16bit CRC
  - Payload is ignored
- Fragment_complete tx timer event
  - SendData sends SubsequentFragment (offset 82) one more time
    - Sender starts fragment_complete timer again
    - If timer event occurs, bail out: Return "Error" callback to calling application
  - Sender starts fragment_complete timer
- SubsequentFragment received with valid 8bit chksum + valid 16bit CRC
  - Sender starts fragment_complete tx timer
  - Offset indicates that this was the last fragment
  - Receiver checks tracking list for missing fragments; none found
- Receiver sends FragmentComplete(OK)
- FragmentComplete(OK) received with valid 8bit chksum + valid 16bit CRC
  - Sender stops fragment_complete tx timer
- Transmitter returns "OK" callback to calling application

CONFIDENTIAL

4.111 User Code Command Class, version 1

The purpose of the User Code Command Class is to supply a Z-Wave™ enabled Door Lock Device with a command class to manage user codes.

4.111.1 User Code Set Command

The User Code Set Command used to set a User Code in the device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE							
Command = USER_CODE_SET							
User Identifier							
User ID Status							
USER_CODE1							
..							
USER_CODEn							

User Identifier (8 bits)

The User Identifier used to recognise the user identity. The User Identifier values MUST be a sequence starting from 1. This field can be ignored in case the node only supports one User Code.
Setting the User Identifier to 0 will address all User Identifiers available in the device.

User ID Status

The User ID Status field indicates the state of the User Identifier. All other values not mentioned in below list are reserved for future implementation.

Hexadecimal	Description
0x00	Available (not set)
0x01	Occupied
0x02	Reserved by administrator
0xFE	Status not available

USER_CODE1…USER_CODEn

These fields contain a list of node User Code. Minimum code length is 4 and maximum 10 ASCII digits. If set to zero's then the User Identifier has not been set.

4.111.2 User Code Get Command

The User Code Get Command is used to request the user code of a specific user identifier.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE							
Command = USER_CODE_GET							
User Identifier							

User Identifier (8 bits)

See description for USER_CODE_SET, 0 is not allowed from USER_CODE_GET.

4.111.3 User Code Report Command

The User Code Report Command can be used by e.g. a door lock device to send a report either unsolicited or requested by the User Code Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE							
Command = USER_CODE_REPORT							
User Identifier							
User ID Status							
USER_CODE1							
..							
USER_CODEn							

See parameter description for USER_CODE_SET.

4.111.4 Users Number Get Command

The User Number Get Command is used to request the number of USER CODES that this node supports.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE							
Command = USERS_NUMBER_GET							

CONFIDENTIAL

4.111.5 Users Number Report Command

The Users Number Report Command used to report the maximum number of USER CODES the given node supports. The Users Number Report Command can be send requested by the Users Number Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_USER_CODE							
Command = USERS_NUMBER_REPORT							
Supported Users							

Supported Users (8 bits)

The number of User Codes this node supports. '0' indicates User Code is not supported by the device.

CONFIDENTIAL

4.112 Version Command Class, version 1

This Version Command Class used to obtain the Z-Wave library type, the Z-Wave protocol version used by the application, the individual command class versions used by the application and the vendor specific application version from a Z-Wave enabled device.

NOTE: **A device supporting a Command Class having a version higher than 1 MUST support the Version Command Class to be able to identify the supported version. In case the device doesn't support the Version Command Class then it can be assumed that all command classes are equal to version 1.**

4.112.1 Version Get Command

The Version Get Command can be used to get the library type, protocol version and application version from a device that supports the Version Command Class. How to obtain the type and protocol version of the library used by the application is described in [1].

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_VERSION							
Command = VERSION_GET							

4.112.2 Version Report Command

The Version Report Command can be used to report the library type, protocol version and application version from a device. The Version Report Command can be send unsolicited or requested by the Version Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_VERSION							
Command = VERSION_REPORT							
Z-Wave Library Type							
Z-Wave Protocol Version							
Z-Wave Protocol Sub Version							
Application Version							
Application Sub Version							

Z-Wave Library Type (8 bits)

Returns the Z-Wave Library Type i.e. it are a controller library, slave library, installer library etc. Refer to ZW_basis_api.h source code file for a list of the possible library types and the assigned values.

CONFIDENTIAL

Z-Wave Protocol Version (8 bits)

Returns the Z-Wave Protocol Version of the used library and can have values in the range 0 to 255. In the table below are the Z-Wave Protocol version listed for a given Developer's Kit version :

Developer's Kit Version	Z-Wave Protocol Version	Z-Wave Protocol Sub Version
6.10.01 (JP)	3	38
6.10.00	3	35
6.01.03	3	37
6.01.02	3	33
6.01.01 (2-ch)	3	26
6.01.00	3	10
6.00.05 Beta 1	3	07
6.00.04 Beta 1	3	06
6.00 Beta 1 Patch 3	3	04
6.00 Beta 1 Patch 2	3	03
6.00 Beta 1 Patch 1	2	99
6.00 Beta 1	2	96
5.03.00	3	28
5.02 Patch 3	2	78
5.02 Patch 2	2	64
5.02 Patch 1	2	51
5.02	2	48
5.01	2	36
5.00 Beta 1 Patch 1	2	22
5.00 Beta 1	2	16
4.54.00	3	40
4.53.01	3	36
4.53.00	3	34
4.52.01	3	22
4.52.00	3	20
4.51	2	97
4.50 Beta 1 Patch 1	2	79
4.50 Beta 1	2	74
4.30 Beta 1	2	30
4.28	2	67
4.27	2	40
4.26	2	32
4.25	2	31
4.24 Patch 1	2	28
4.24	2	24
4.23	2	17
4.22	2	09
4.21	2	06
4.20	1	97
4.11	1	91
4.10	1	78
4.07	2	27
4.06	2	23
4.05	2	07
4.04	1	99
4.03	1	81
4.02	1	69

CONFIDENTIAL

Developer's Kit Version	Z-Wave Protocol Version	Z-Wave Protocol Sub Version
4.01	1	68
4.00	1	59
3.40	1	53
3.31	1	44
3.30	1	37
3.22	1	39
3.21	1	25
3.20	1	21

Table 26, Z-Wave Protocol version for a given Developer's Kit version

Warning: Products can only be Z-Wave certified based on matured versions of the Z-Wave Protocol, i.e. Developer's Kit having versions different from x.y0.

Z-Wave Protocol Sub Version (8 bits)

Returns the Z-Wave Protocol Sub Version of the used library and can have values in the range 0 to 255.

Application Version (8 bits)

Returns the Application Version and can have values in the range 0 to 255. The manufacturer assigns the Application Version.

Application Sub Version (8 bits)

Returns the Application Sub Version and can have values in the range 0 to 255. The manufacturer assigns the Application Sub Version.

4.112.3 Version Command Class Get Command

The Version Command Class Get Command can be used to request the individual command class versions from a device. Only versions from the command classes shown in the NIF can be requested. It's not possible to get a version number for the Generic and Specific Device Classes.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_VERSION							
Command = VERSION_COMMAND_CLASS_GET							
Requested Command Class							

Requested Command Class (8 bits)

The Request Command Class field specifies which command class identifier is being requested.

CONFIDENTIAL

4.112.4 Version Command Class Report Command

The Version Command Class Report Command can be used to report the individual command class versions from a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_VERSION							
Command = VERSION_COMMAND_CLASS_REPORT							
Requested Command Class							
Command Class Version							

Requested Command Class (8 bits)

The Requested Command Class field specifies what command class the returned version belongs to.

Command Class Version (8 bits)

Returns the Command Class Version and can have values in the range 1 to 255. It starts with 1 and is incremented every time a new version of the Command Class is released. In case the requested command class is not present in the NIF then Command Class Version is set to zero. Refer to ZW_classcmd.h source code file for actual version number.

CONFIDENTIAL

4.113 Wake Up Command Class, version 1

The Wake Up Command Class version 1 allows battery-operated devices to wake up occasionally and notify another device (always listening), that the device is ready to receive any queued commands.
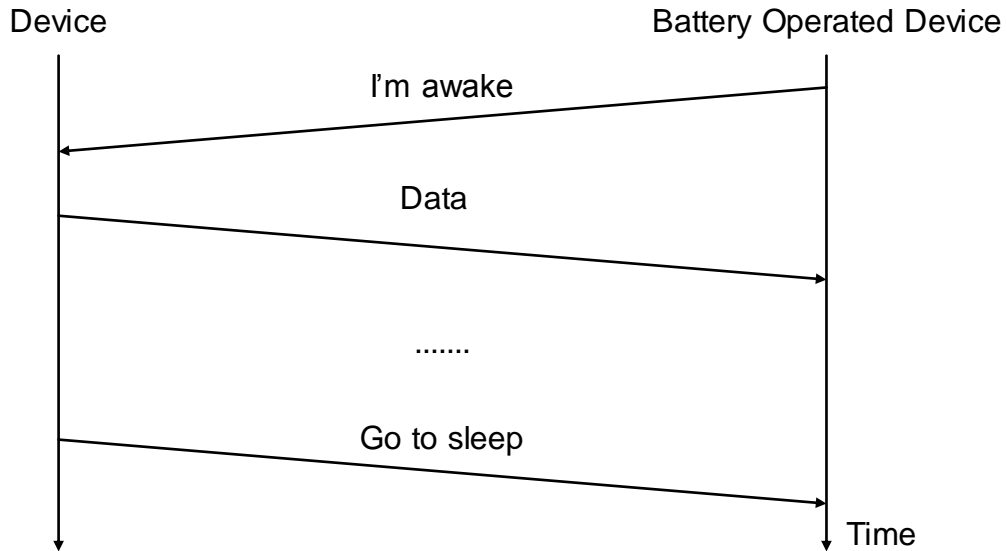
Figure 22, Wake Up sequence

Since this Command Class is normally used by battery operated devices it is REQUIRED that Wake Up Notification commands are handled immediately and response (or data) is sent as soon as possible, in order to minimize battery consumption.

4.113.1 Wake Up Interval Set Command

The Wake Up Interval Set Command used to configure the wake up interval of a device and the node ID of the device receiving the Wake Up Notification Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_INTERVAL_SET							
Seconds 1 (MSB)							
Seconds 2							
Seconds 3 (LSB)							
NodeID							

Seconds 1 … 3 (24 bits)

The Seconds field contains the number of seconds between wake up of a battery-operated device. The first byte is the most significant byte.

CONFIDENTIAL

In case number of seconds is set to 0 then wake up is initiated by an event determined by the application e.g. a pushbutton activation.

NodeID (8 bits)

The NodeID field contains the node ID of the device receiving the Wake Up Notification Command.

4.113.2 Wake Up Interval Get Command

The Wake Up Interval Get Command is used to request the wake up interval of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_INTERVAL_GET							

4.113.3 Wake Up Interval Report Command

The Wake Up Interval Report Command used to report the wake up interval of a device and the node ID of the device receiving the Wake Up Notification Command. The Wake Up Interval Report Command can be send unsolicited or requested by the Wake Up Interval Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_INTERVAL_REPORT							
Seconds 1 (MSB)							
Seconds 2							
Seconds 3 (LSB)							
NodeID							

Seconds 1 … 3 (24 bits)

The Seconds field contains the number of seconds between wake up of a battery-operated device. The first byte is the most significant byte.

NodeID (8 bits)

The NodeID field contains the ID on the node that SHOULD receive the Wake Up Notification Command.

4.113.4 Wake Up Notification Command

Devices will use the Wake Up Notification Command to notify other devices, that it is awake. Devices will normally start a timer that will force the device to power down after some time in case the Wake Up No More Information Command fails. This time SHALL be sufficient for the receiver to receive the command, check if any information is pending and send response.
A device is allowed to send the Wake Up Notification Command as broadcast as long as a node ID is not configured by Wake Up Interval Set Command. If a Wake Up Notification Command broadcast is received then it is not allowed to respond with a Wake Up No More Information Command before the node ID is configured by Wake Up Interval Set Command.

Please note that if the battery operated device wishes to send an unsolicited report, then it SHOULD be done before sending the Wake Up Notification Command. Otherwise, the Wake Up No More Information reply is likely to collide with the unsolicited report.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_NOTIFICATION							

4.113.5 Wake Up No More Information Command

The Wake Up No More Information Command used by devices to notify the sender of the Wake Up Notification Command that it SHOULD NOT expect any more information, and consequently it can go back to sleep to minimize power consumption.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_NO_MORE_INFORMATION							

CONFIDENTIAL

4.114 Wake Up Command Class, version 2

The Wake Up Command Class version 2 enables read back of the Wake up interval capabilities in a node. Version 2 comprises of all the version 1 commands and two new commands to enable this feature.

4.114.1 Wake Up Interval Capabilities Get Command

The Wake Up Interval Capabilities Get Command is used to request the wake up interval capabilities of a device.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_INTERVAL_CAPABILITIES_GET							

4.114.2 Wake Up Interval Capabilities Report Command

The Wake Up Interval Capabilities Report Command used to report the wake up interval capabilities of a device. The Wake Up Interval Capabilities Report Command can be send requested by the Wake Up Interval Capabilities Get Command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_WAKE_UP							
Command = WAKE_UP_INTERVAL_CAPABILITIES_REPORT							
Minimum Wake Up Interval Seconds Byte 1							
Minimum Wake Up Interval Seconds Byte 2							
Minimum Wake Up Interval Seconds Byte 3							
Maximum Wake Up Interval Seconds Byte 1							
Maximum Wake Up Interval Seconds Byte 2							
Maximum Wake Up Interval Seconds Byte 3							
Default Wake Up Interval Seconds Byte 1							
Default Wake Up Interval Seconds Byte 2							
Default Wake Up Interval Seconds Byte 3							
Wake Up Interval Step Seconds Byte 1							
Wake Up Interval Step Seconds Byte 2							
Wake Up Interval Step Seconds Byte 3							

Byte 1 is the most significant byte for all of the following fields.

CONFIDENTIAL

Minimum Wake Up Interval Seconds 1 … 3 (24 bits)

This field specifies the minimum wake up interval a battery-operated device supports.

The following values are possible:

Decimal	Description
0	No minimum / maximum / default wake up interval, the battery-operated device is activated by e.g. user interaction in form of a button press on the battery-operated device. If this field is 0, then all the other fields MUST also be 0. **Note**: This is identical to the specified behavior in Version 1 of the command class.
1 … 16777215	The minimum wake up interval in seconds supported by the battery-operated device.

Maximum Wake Up Interval Seconds 1 … 3 (24 bits)

This field defines the maximum wake up interval a battery-operated device supports.

The following values are possible:

Decimal	Description
0	No minimum / maximum / default wake up interval, the battery-operated device is activated by e.g. user interaction in form of a button press on the battery-operated device. If this field is 0, then all the other fields MUST also be 0. **Note**: This is identical to the specified behavior in Version 1 of the command class.
1 … 16777215	The maximum wake up interval in seconds supported by the battery-operated device. This interval MUST never be lower than the minimum wake up interval, but it can be equal, which means the device only supports one interval.

CONFIDENTIAL

Default Wake Up Interval Seconds 1 … 3 (24 bits)

This field defines the default wake up interval a battery-operated device supports.

The following values are possible:

Decimal	Description
0	No minimum / maximum / default wake up interval, the battery-operated device is activated by e.g. user interaction in form of a button press on the battery-operated device. If this field is 0, then all the other fields MUST also be 0. **Note**: This is identical to the specified behavior in Version 1 of the command class.
1 … 16777215	The default wake up interval in seconds supported by the battery-operated device. This interval MUST never be lower than the minimum wake up interval or higher than the maximum wake up interval.

Wake Up Interval Step Seconds 1 … 3 (24 bits)

This field defines the resolution of possible wake up intervals, which a battery-operated device supports.

The following values are possible:

Decimal	Description
0	No interval steps are possible. The battery-operated device only supports the minimum and maximum wake up interval. The interval step MUST be set to 0 if both the maximum and the minimum interval are equal.
1 … 16777215	The wake up interval step in seconds supported by the battery-operated device. This interval MUST NOT exceed the difference between the minimum and maximum wake up interval. The interval steps SHOULD have a length so the difference between the maximum and minimum Wake Up Interval is a multiple of the interval steps. *Examples:* If a device has minimum wake up interval of 5 minutes (300 seconds) and a maximum wake up interval of 10 minutes (600 seconds), then the wake up interval step MUST NOT exceed 5 minutes (300 seconds) as this would be larger than the difference of the minimum and maximum interval. If a device has minimum wake up interval of 5 minutes (300 seconds) and a maximum wake up interval of 10 minutes (600 seconds) and wake up interval step of 100 seconds you MAY only set the following intervals on the device: 300 seconds 400 seconds 500 seconds 600 seconds

CONFIDENTIAL

4.115 Z/IP Command Class

The Z/IP Command Class is a special command class intended for encapsulation of Z-Wave application commands in IP packets. Commands defined in this command class SHOULD NOT be sent in Z-Wave frames.

Z/IP Packets MAY be exchanged between IP hosts running over physical layers such as Ethernet, WiFi or Z-Wave.

4.115.1 Z/IP Packet Command

IP→UDP:4123→Z/IP Packet header→Z-Wave command

A Z/IP Packet Command is carried in a UDP packet. The Z/IP Packet carries a Z-Wave application command.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZIP							
Command = COMMAND_ZIP_PACKET							
Ack Request	Ack Response	NAck Response	(NAck flags)			Reserved	
			Waiting	Queue Full	Option Error		
Header ext. included	Z-Wave Cmd Included	More Information	Reserved				
Seq No							
Res	Source End Point						
Bit address	Destination End Point						
Header extension 1 (OPTIONAL)							
... (OPTIONAL)							
Header extension N (OPTIONAL)							
Z-Wave command 1 (OPTIONAL)							
... (OPTIONAL)							
Z-Wave command N (OPTIONAL)							

Note: This Z-Wave command is a virtual command.
The Z/IP Packet is for transport of encapsulated Z-Wave commands in an IP environment.
The normal Z-Wave frame size limitation does not apply to this command.

When receiving a Z/IP packet, the receiving IP host MUST inspect the header flags in order to determine the offset to use for accessing the OPTIONAL fields.

CONFIDENTIAL

Ack Request (1 bit)

The `Ack Request` flag signals that the destination IP host MUST return an Ack or Nack message in response to the actual Z/IP Packet.

Ack request	Value
Return Ack or Nack	'1'
No confirmation needed	'0'

This bit is intended only for control of application-level acknowledgement for Z/IP packets.
Z-Wave link-level acknowledgement SHOULD always be used between Z-Wave nodes when Z-Wave is used as link layer.

If Ack is requested, a Z/IP Router MUST ensure that a response is returned to an originating node no later than 250ms after the Z/IP Packet was received.
In case of success for a classic node a Z/IP Ack indication MUST be returned by the Z/IP Router.
In case of success for a Z/IP node the Z/IP node itself MUST return a Z/IP Ack indication.

In case the final delivery status is undetermined after 250ms, a Z/IP Router MUST return a "NAck+Waiting" indication, irrespective whether the node is a classic node or a Z/IP node.
An "Expected Delay" Header Extension MAY be returned by the Z/IP Router. Refer to 4.115.1.1.1.

Ack Response (1 bit)

The `Ack` flag signals that the destination IP host received the Z/IP packet.

Ack	Value
Ack	'1'
(check Nack)	'0'

A Z/IP `Ack` or `Nack` packet MUST have the same `Seq No` value as the Z/IP packet being acknowledged.
In case of link-layer retransmissions, multiple Z/IP Packet Acks MAY be received. Z/IP Packet Ack duplicates MUST be ignored.
If the Ack flag is not set, the Nack flag MUST be inspected

Nack Response (1 bit)

The `Nack` flag signals that the Z/IP packet did not (yet) reach the destination IP host. This message MAY be returned by intermediate nodes such as a Z/IP Router.

Nack	Value
Nack (check Waiting)	'1'
(ignore)	'0'

A Z/IP `Ack` or `Nack` packet MUST have the same `Seq No` value as the Z/IP packet being acknowledged.
In case of link-layer retransmissions, multiple Z/IP Packet acks MAY be received. Z/IP Packet ack duplicates MUST be ignored.

CONFIDENTIAL

If the `NAck` flag is set, all `NAck` flags MUST also be inspected.

If no `NAck` flags are set, the message was lost but no specific reason is provided.

Waiting (NAck flag)

The `Waiting` flag is a companion flag to the NAck flag. It SHOULD only be inspected if the `Nack` flag is true.

The `Waiting` flag signals that the destination IP host MAY have a long response time. The message has not timed out yet. This message MAY be returned by intermediate nodes such as a Z/IP Router. The waiting time depends on the properties of the destination IP host.

Waiting	Value
Waiting	'1'
(not waiting)	'0'

An "Expected Delay" Header Extension MAY be returned by the Z/IP Router. Refer to 4.115.1.1.1. A Z/IP Client SHOULD use this information to provide better user responsiveness. A default value of 90 seconds MUST be used by a Z/IP Client if no "Expected delay" header extension is provided.

A Z/IP "NAck+Waiting" indication is returned for every packet that is queued up. If a Z/IP Client queues up three configuration packets it will receive a NAck Waiting after each packet. It MAY be desirable to queue up three configuration commands if the intention is to perform a few configuration changes and the allow the battery node to return to sleep.

In case one wants to transfer larger amounts of data, e.g. a security certificate or a new firmware image, the RECOMMENDED procedure is to queue up a single "More Information" message to make the destination node stay awake. When a Z/IP Ack is returned to the Z/IP Client, the Z/IP Client can then start transferring packets at a higher rate.

If a new message is delayed for more than 200ms, a Z/IP Router MUST transmit a "NAck+Waiting" indication to let the Z/IP Client know that the message is still in the network.

A Z/IP Client waiting for more than 300ms without receiving an Ack or NAck indication for a new message MAY conclude that the message is lost and retransmit the message.

If a message has been delayed for more than 60 seconds, a Z/IP Router MUST transmit a new "NAck+Waiting" indication every 60 seconds to let the Z/IP Client know that it is still operational.

A Z/IP Client waiting for more than 90 seconds after receiving a "NAck+Waiting" indication MAY conclude that the message is lost and retransmit the message.

A Z/IP "Ack" MUST follow when the message has been delivered. A Z/IP Router MUST return a Z/IP "NAck" if it has information that the message was not delivered.

Queue Full (NAck flag)

The `Queue Full` flag is a companion flag to the NAck flag. It SHOULD only be inspected if the `Nack` flag is true.
Typically, this flag will be returned by a Z/IP Router for packets targeting battery nodes, but in a busy network or during bulk data transfers or route re-discovery, the Queue Full flag MAY also be returned for always listening destinations.
A Z/IP Client MUST wait for at least 10 seconds before re-transmitting the message.

The `Queue Full` flag MUST be returned by a Z/IP Router if there is no more room in the queue system used for delivering messages to the PAN.
The `Waiting` flag MUST be ignored if the `Queue Full` flag is true.

An "Expected delay" header extension MAY indicate the expected waiting time. A Z/IP Client MAY re-send the message after the indicated time. A default value of 90 seconds MUST be used if no "Expected delay" header extension is provided.

If receiving an Ack indication for a previous message, the Z/IP Client MAY skip waiting and re-transmit remaining packets. This MAY happen if one or more message were successfully placed in the queue before the queue was running full.

Queue Full	Value
Queue Full (packet lost)	'1'
Queue OK	'0'

Option Error (NAck flag)

The `Option Error` flag is a companion flag to the NAck flag. It SHOULD only be inspected if the `Nack` flag is true.

The `Option Error` flag MUST be returned if a critical option is not recognized by the receiving node.
The `Option Error` flag MUST NOT be returned if an elective option is not recognized by the receiving node. Elective options MUST be silently ignored by a receiving node.

Option Error	Value
Option Error (packet lost)	'1'
(no error)	'0'

A node setting the Option Error flag SHOULD include the offending Header Extension in the "NAck+OptionError" inditation returned to the originating node.

A node receiving a "NAck+OptionError" indication MUST NOT process the header extension as it is only included for debugging purposes.

Header extension Included (1 bit)

The `Header Extension Included` flag signals that an extended header is included in the Z/IP packet. Refer to the Header Extension description below.

Header extension Included	Value
Extended header included	'1'
Extended header NOT included	'0'

Reserved

The reserved field is for future use. The implementation SHALL zero these fields and SHALL make no assumptions on the values of these fields nor perform processing based on their content.

Z-Wave Command Included (1 bit)

The `Z-Wave Command Included` flag signals that a Z-Wave control command is included in the Z/IP packet.

Z-Wave Command Included	Value
Z-Wave command is included	'1'
Z-Wave command NOT included	'0'

In case a Z/IP Gateway receives a Z/IP Packet for a classic Z-Wave node with no Z-Wave command included AND Ack Request is '1', then the Z/IP Gateway MUST return an Ack Response = '1'.

More Information (1 bit)

Set this flag to prevent a WUN or FLN node from returning to sleep the next minute. A sending Z/IP Client that is aware that it will be sending more data to the destination node MAY set this flag.

More Information	Value
WUN Node SHOULD be kept awake	'1'
WUN Node SHOULD be put to sleep	'0'

Seq No (8 bit)

The `Seq No` value MUST be incremented for every new unique Z/IP Packet. Retransmitted Z/IP packets MUST have same value as the original Z/IP Packet.

A Z/IP `Ack` or `Nack` packet MUST have the same `Seq No` value as the Z/IP packet being acknowledged.

Source End Point (7 bits)

This field indicates the end point from where the command was send. Valid Source End Points are 0 (zero) to 127.
The Source End Point 0 represents the physical device.

Bit address (1 bit)

This bit is set to 0 if the destination end point is addressed individually.

This bit is set to 1 if multiple destination end points are given in a bit mask for parallel addressing. Only destination end points 1..7 are bit addressable.
Bit addressing MUST NOT be used if the encapsulated command is a request (requiring a reply from the destination). A receiving node SHOULD ignore requests if received via bit addressing.

Destination End Point (7 bits)

This field identifies the destination end point.

The value 0 (zero) indicates that the physical device is addressed. All other values identify an end point.

Header Extension (variable length)

The Header extension field MAY be used for additional Z/IP packet options only applicable to certain uses of the packet. A Z/IP node MUST support the Header Extension field.

The size of the complete Z/IP header extension MUST be signaled in byte #0 of the extended header.
The size includes byte #0, i.e. the size field.
The size MUST NOT exceed 255 octets.

The Z/IP header extension MAY contain multiple options.

Z/IP Packet options (variable length)

Z/IP Packet options are carried in the Z/IP Packet Header Extension. Each option is formatted as a type, length, value (TLV) structure following the convention of Next Header length in [8] the value is absent if the length is zero.

Thus,

Byte #0 (T) of each option MUST report the type of the option.
Byte #1 (L) of each option MUST report the length of the option.
Byte #2..#n (V) of each option MAY contain additional option fields.

Option types are categorized in two classes: Elective or Critical.
A receiving node MAY ignore an elective option if it does not support this actual option.
A receiving node MUST NOT ignore a critical option if it does not support this actual option. In that case, the node MUST return a "NAck+OptionError" indication to the originating node.
The most significant bit of the option Type indicates if the option is Elective (0) or Critical (1).

An Elective option MAY be used to make a node deliver a better service, but the service still works even if the node does not support the option. On the other hand, a node MAY potentially do something wrong if it does not support a critical option.

CONFIDENTIAL

Z/IP Header Extension	
Header Extension length	
Option 1	(OPTIONAL)
Option 2	(OPTIONAL)

Z-Wave Command (variable length)

This field carries a Z-Wave command. Since the Z/IP Packet uses IP transport, the classic Z-Wave payload length limitation does not apply.

4.115.1.1   Z/IP Packet options

The Z/IP Packet Header Extension MAY contain one or more options; each identified by a unique type defined in Table 27.

7	6	5	4	3	2	1	0
Elective (0) / Critical (1)	Type						
Length							
Value (OPTIONAL)							

Table 27, Z/IP Packet option types

Option Type	Type	Class
(reserved)	0 (0x00)	Elective
Expected delay	1 (0x01)	Elective
(reserved)	128 (0x80)	Critical

A receiving host MUST accept options in any order.

4.115.1.1.1   Expected Delay

7	6	5	4	3	2	1	0
Type = ZIP_OPTION_EXPECTED_DELAY							
Length = 3							
Seconds 1 (MSB)							
Seconds 2							
Seconds 3 (LSB)							

4.116 Z/IP-ND Command Class

Z/IP-ND Command Class builds on the same principles as IPv6 ND [4], [5] and is inspired by the frame formats. Z/IP-ND does however not implement the full range of functions defined for IPv6 ND.

Z/IP-ND commands allow a Z/IP Gateway to translate between an IPv6 address and a Z-Wave NodeID (Link-Layer address) when requested by an IP host located in a Z-Wave HAN or anywhere else in an IPv6 environment.

The Z/IP-ND Commands are not intended for classic Z-Wave applications. Z/IP-ND messages MUST always be carried in Z/IP UDP datagrams.

4.116.1 Z/IP Node Solicitation Command

The Z/IP Node Solicitation Command is used to resolve an IPv6 address of a Z-Wave node to the NodeID (Link-Layer address) of that node in its actual Z-Wave HAN / IP subnet.
Several IPv6 addresses MAY be resolved to the same NodeID.

The Zip Node Solicitation MUST be transmitted in unicast to the Z/IP Gateway of the actual Z/IP HAN. A Z/IP Gateway MUST NOT respond to Zip Node Solicitation commands received via multicast.

A Zip Node Advertisement MUST be returned in response to the Zip Node Solicitation.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZIP_ND							
Command = COMMAND_ZIP_NODE_SOLICITATION							
Reserved							
Node ID **= 0**							
IPv6 Address 1							
…							
IPv6 Address 16							

Reserved

Reserved bits MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Node ID

The NodeID field is not used in the Zip Node Solicitation. The field MUST be set to zero by a transmitting host and ignored by a receiving host.

IPv6 Address (16 bytes)

The IP address of the target Z-Wave node. It MUST NOT be a multicast address.

CONFIDENTIAL

4.116.2 Z/IP Inverse Node Solicitation Command

The Z/IP Inverse Node Solicitation Command is used to resolve a NodeID (link-layer address) of a Z-Wave node to an IPv6 address of that node in its actual Z-Wave HAN / IP subnet.

The Zip Inverse Node Solicitation MUST be transmitted in unicast to the Z/IP Gateway of the actual Z/IP HAN. A Z/IP Gateway MUST NOT respond to Zip Inverse Node Solicitation commands received via multicast.

A Zip Node Advertisement MUST be returned in response to the Zip Inverse Node Solicitation.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZIP_ND							
Command = COMMAND_ZIP_INV_NODE_SOLICITATION							
Reserved						Local	*Reserved*	
Node ID							

Reserved

Reserved bits MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Local

The flag indicates that the requester would like to receive the site-local address (a.k.a. ULA) even if a global address exists. The flag is typically used by a configuration tool when creating an association between HAN nodes within the same site. Using ULA addresses for intra-HAN association serves to decouple long-term associations in the home from frequently changing global prefixes.

Node ID

The NodeID (Link-Layer Address) that is to be resolved to an IPv6 address.

CONFIDENTIAL

4.116.3 Z/IP Node Advertisement Command

The Z/IP Node Advertisement Command is sent by a Z/IP Gateway in response to a unicast Zip Node Solicitation or a unicast Zip Inverse Node Solicitation. The Zip Node Advertisement MUST valid information in both the IPv6 Address and Node ID fields if such information can be found by the Z/IP Gateway.

A Zip Node Advertisement MUST NOT be transmitted in unsolicited messages.
A Zip Node Advertisement MUST NOT be transmitted in multicast.

7	6	5	4	3	2	1	0
Command Class = COMMAND_CLASS_ZIP_ND							
Command = COMMAND_ZIP_NODE_ADVERTISEMENT							
Reserved					Local	Validity	
Node ID							
IPv6 Address 1							
…							
IPv6 Address 16							
Home ID 1							
…							
Home ID 4							

Reserved

Reserved bits MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Local (1 bit)

The flag indicates that the requester asked for the site-local address (a.k.a. ULA).
A ULA address is returned. A global address MAY exist.

Validity (2 bits)

A two-bit codeword that indicates the validity of the returned information.

Validity	Comment
INFORMATION_OK	The Node Advertisement contains valid information in both the IPv6 Address and Node ID fields.
INFORMATION_OBSOLETE	The information in the IPv6 Address and Node ID fields is obsolete. No node exists in the network with this address information. The information SHOULD only be used to inform a user that the actual node is no more present in the network.
INFORMATION_NOT_FOUND	The responding Z/IP Gateway could not locate valid information. IPv6 Address and Node ID fields MUST be ignored.

Table 28, Encoding of Zip Node Advertisement::Validity parameter

Node ID

The NodeID MUST correspond to the IPv6 Address contained in this Zip Node Advertisement message.

IPv6 Address (16 bytes)

The IPv6 Address MUST corresponds to the NodeID contained in this Zip Node Advertisement message.

An IPv6 host MAY have more than one IPv6 address.
If the Zip Node Advertisement is a response to a Zip Node Solicitation, the IPv6 Address MUST be the same as the one carried in the Zip Node Solicitation.

A Z/IP Gateway returning a Zip Node Advertisement in response to a Zip Inverse Node Solicitation MAY have several IPv6 addresses to choose from. The reported IPv6 Address MUST be selected according to the following priority list:

If "local" flag is set:

      1.   Unique Local Address (ULA) prefix

If "local" flag is not set:

      1.   Global routable address

      2.   Unique Local Address (ULA) prefix

In other words,
if the Z/IP node has a globally routable address then that address MUST be reported.
Else the locally routable address constructed from a ULA prefix and the NodeID MUST be reported.

If a Z/IP Inverse Node Solicitation command is transmitted in an IPv6 packet the returned Z/IP Node Advertisement MUST carry the IPv6 address of the actual node.

If a Z/IP Inverse Node Solicitation command is transmitted in an IPv4 packet the returned Z/IP Node Advertisement MUST carry the IPv4 address of the actual node formatted as an IPv4-mapped IPv6 address [9].

The IP address carried in the Z/IP Node Advertisement MAY be all zeros. The reason may be that the Z/IP Gateway is still waiting for a DHCP response after including a new node. A Z/IP client MAY re-issue

CONFIDENTIAL

another a Z/IP Inverse Node Solicitation command after a delay of 2 seconds. The delay MUST be doubled before each new attempt. The delay SHOULD be capped at 32 seconds.

Home ID (4 bytes)

Unique network address of the link layer network. All nodes in a Z-Wave network share the same Home ID. The Home ID MAY be used for bookkeeping of complete node information in managed installations.

APPENDIX A MANUFACTURER IDS

NOTICE: Sigma Designs upon request MUST assign the Manufacturer ID.

Customer	Manufacturer ID
2B Electronics	0x0028
2gig Technologies Inc.	0x009B
3e Technologies	0x002A
A-1 Components	0x0022
Abilia	0x0117
ACT - Advanced Control Technologies	0x0001
AEON Labs	0x0086
Airline Mechanical Co., Ltd.	0x0111
Alarm.com	0x0094
Asia Heading	0x0029
Atech	0x002B
Balboa Instruments	0x0018
BeNext	0x008A
BeSafer	0x002C
Boca Devices	0x0023
Broadband Energy Networks Inc.	0x002D
BuLogics	0x0026
Cameo Communications Inc.	0x009C
Carrier	0x002E
CasaWorks	0x000B
Chromagic Technologies Corporation	0x0116
Color Kinetics Incorporated	0x002F
Connected Object	0x011B
ControlThink LC	0x0019
ConvergeX Ltd.	0x000F
Cooper Lighting	0x0079
Cooper Wiring Devices	0x001A
Coventive Technologies Inc.	0x009D
Cyberhouse	0x0014
CyberTAN Technology, Inc.	0x0067
Cytech Technology Pre Ltd.	0x0030
Danfoss	0x0002
Destiny Networks	0x0031
Diehl AKO	0x0103
Digital 5, Inc.	0x0032
Ecolink	0x11F
Eka Systems	0x0087
Electronic Solutions	0x0033
El-Gev Electronics LTD	0x0034
ELK Products, Inc.	0x001B
Embedit A/S	0x0035
Everspring	0x0060

CONFIDENTIAL

Customer	Manufacturer ID
Evolve	0x0113
Exceptional Innovations	0x0036
Exhausto	0x0004
Exigent Sensors	0x009F
Fakro	0x0085
Fibargroup	0x010F
Foard Systems	0x0037
FortrezZ LLC	0x0084
Foxconn	0x011D
Frostdale	0x0110
Good Way Technology Co., Ltd	0x0068
GreenWave Reality Inc.	0x0099
HiTech Automation	0x0017
Home Automated Inc.	0x005B
Home Automated Living	0x000D
Home Automation Europe	0x009A
Home Director	0x0038
Homemanageables, Inc.	0x0070
Homepro	0x0050
HomeScenario	0x0162
HomeSeer Technologies	0x000C
Honeywell	0x0039
Horstmann Controls Limited	0x0059
iCOM Technology b.v.	0x0011
Ingersoll Rand (Schlage)	0x006C
Inlon Srl	0x003A
INNOVUS	0x0077
Intel	0x0006
IntelliCon	0x001C
Intermatic	0x0005
Internet Dom	0x0013
IR Sec. & Safety	0x003B
Jasco Products	0x0063
Kamstrup A/S	0x0091
Lagotek Corporation	0x0051
Leviton	0x001D
Lifestyle Networks	0x003C
Logitech	0x007F
Loudwater Technologies, LLC	0x0025
LS Control	0x0071
Marmitek BV	0x003D
Martec Access Products	0x003E
MB Turn Key Design	0x008F
Merten	0x007A
MITSUMI	0x0112
Monster Cable	0x007E

Customer	Manufacturer ID
Motorola	0x003F
MTC Maintronic Germany	0x0083
Napco Security Technologies, Inc.	0x0121
NorthQ	0x0096
Novar Electrical Devices and Systems (EDS)	0x0040
Omnima Limited	0x0119
OpenPeak Inc.	0x0041
Poly-control	0x010E
PowerLynx	0x0016
Pragmatic Consulting Inc.	0x0042
Pulse Technologies (Aspalis)	0x005D
Qees	0x0095
Radio Thermostat Company of America (RTC)	0x0098
Raritan	0x008E
Reitz-Group.de	0x0064
Remotec Technology Ltd	0x5254
Residential Control Systems, Inc. (RCS)	0x0010
RS Scene Automation	0x0065
Ryherd Ventures	0x001E
San Shih Electrical Enterprise Co., Ltd.	0x0093
Scientia Technologies, Inc.	0x001F
Secure Wireless	0x011E
Seluxit	0x0069
Senmatic A/S	0x0043
Sequoia Technology LTD	0x0044
Sigma Designs	0x0000
Sine Wireless	0x0045
Smart Products, Inc.	0x0046
SMK Manufacturing Inc.	0x0102
Somfy	0x0047
Sylvania	0x0009
Team Precision PCL	0x0089
Techniku	0x000A
Tell It Online	0x0012
Telsey	0x0048
There Corporation	0x010C
TKB Home	0x0118
TKH Group / Eminent	0x011C
Trane Corporation	0x008B
TrickleStar	0x0066
Tricklestar Ltd. (former Empower Controls Ltd.)	0x006B
Tridium	0x0055
Twisthink	0x0049
Universal Electronics Inc.	0x0020
VDA	0x010A
Vero Duco	0x0080

Customer	Manufacturer ID
ViewSonic Corporation	0x005E
Vimar CRS	0x0007
Vision Security	0x0109
Visualize	0x004A
Watt Stopper	0x004B
Wayne Dalton	0x0008
Wenzhou MTLC Electric Appliances Co.,Ltd.	0x011A
Wintop	0x0097
Woodward Labs	0x004C
Wrap	0x0003
Xanboo	0x004D
Zdata, LLC.	0x004E
Zonoff	0x0120
Z-Wave Technologia	0x004F
Z-Wave.Me	0x0115
Zykronix	0x0021

Table 29, Manufacturer ID values

CONFIDENTIAL

APPENDIX B ASCII CODES

The standard ASCII table defines 128 character codes (from 0 to 127), of which, the first 32 are control codes (non-printable), and the remaining 96 character codes are printable characters. The table below shows the hexadecimal values of the ASCII character codes, e.g. the ASCII code for the capital letter "A" is equal to 0x41:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	U
2		!	"	#	$	%	&	'	(	)	*	+	,	–	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{			}	~	

Table 30, The standard ASCII Table

CONFIDENTIAL

In addition to the 128 standard ASCII codes (the ones listed above ranging from 0 to 127), most systems have another 128 extra codes which form what is known as extended ASCII (with ranges from 128 to 255). The OEM Extended ASCII character set is included in all PC-compatible computers as the default character set when the system boots before loading any operating system and under MS-DOS. It includes some foreign signs, some marked characters and also pieces to draw simple panels. The table below shows the hexadecimal values of the OEM Extended ASCII character codes, e.g. the ASCII code for the capital letter "Æ" is equal to 0x92:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E	α	β	Γ	Π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	∈	∩
F	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

Table 31, OEM Extended ASCII Table

Below are listed codes for players, radios etc. as an alternative to the OEM Extended ASCII codes. Undefined values MUST be ignored.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	▪	▫	►	❚	■	•	►►	◄◄	◊							
9																
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	‾
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Table 32, Players Table

CONFIDENTIAL

APPENDIX C CRC-CCITT SOURCE CODE

The checksum algorithm implements a CRC-CCITT using initialization value equal to 0x1D0F and 0x1021 (normal representation) as the poly.

```
/****************************** ZW_crc.h ******************************
***********************************************************************/
#ifndef _ZW_CRC_H_
#define _ZW_CRC_H_

#include <ZW_typedefs.h>
/**********************************************************************
/
/*                          EXPORTED FUNCTIONS
*/
/**********************************************************************
/

WORD
ZW_CreateCrc16(
  BYTE *pHeadeAddr,
  BYTE bHeaderLen,
  BYTE *pPayloadAddr,
  BYTE bPayloadLen
);


/*========================= ZW_CheckCrc16 =========================
**    CRC-CCITT (0x1D0F) calculation / check
**
**    In:  byte string excluding 16bit check field
**    Out: CRC-16 value
**  or
**    In:  byte string including 16bit check field
**    Out: zero when OK
**
**---------------------------------------------------------------------
*/
WORD
ZW_CheckCrc16(
  WORD crc,
  BYTE *pDataAddr,
  WORD bDataLen
);

#endif /* _ZW_CRC_H_ */
```

```
/***************************** ZW_crc.c ******************************
********************************************************************/
#include <ZW_typedefs.h>
#include <ZW_crc.h>
#include <ZW_uart_api.h>
#define POLY 0x1021         /* crc-ccitt mask */

/*                 CRC calculation                        */

/*========================== ZW_CheckCrc16 ===========================
**    CRC-CCITT (0x1D0F) calculation / check
**
**    In:  byte string excluding 16bit check field
**    Out: CRC-16 value
**  or
**    In:  byte string including 16bit check field
**    Out: zero when OK
**
**  and
**    In: The crc input SHOULD normally be set to the initialization
**        value = 0x1D0F.
**        It can also be used to carry over crc value between separate
**        calculations of multiple parts of a frame, e.g. header and body.
**
**-------------------------------------------------------------------------
*/
WORD
ZW_CheckCrc16(
  WORD crc,
  BYTE *pDataAddr,
  WORD bDataLen
)
{
  BYTE WorkData;
  BYTE bitMask;
  BYTE NewBit;

  while(bDataLen--)
  {
    WorkData = *pDataAddr++;
    for (bitMask = 0x80; bitMask != 0; bitMask >>= 1) {
      /* Align test bit with next bit of the message byte, starting with msb.
*/
      NewBit = ((WorkData & bitMask) != 0) ^ ((crc & 0x8000) != 0);
      crc <<= 1;
      if (NewBit) {
        crc ^= POLY;
      }
    } /* for (bitMask = 0x80; bitMask != 0; bitMask >>= 1) */
  }
  return crc;
}
```

```
WORD
ZW_CreateCrc16(
  BYTE *pHeaderAddr,
  BYTE bHeaderLen,
  BYTE *pPayloadAddr,
  BYTE bPayloadLen
)
{
  WORD crc;

  crc = 0x1D0F;
  crc = ZW_CheckCrc16(crc, pHeaderAddr, bHeaderLen);
  crc = ZW_CheckCrc16(crc, pPayloadAddr, bPayloadLen);
  return crc;
}
```

REFERENCES

[1]   Sigma Designs, SDS10242, Z-Wave Device Class Specification.
[2]   Sigma Designs, INS10247, Z-Wave ZW0102/ZW0201/ZW0301 Application Programming Guide.
[3]   Sigma Designs, APL10720, Programming the ZW0201 Flash from Internal MCU.
[4]   Sigma Designs, SDS10865, Software Design Spec., Z-Wave Application Security layer specification
[5]   IETF RFC 4861, Neighbor Discovery for IP version 6 (IPv6),
        http://tools.ietf.org/pdf/rfc4861.pdf
[6]   IETF RFC 3122, Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification,
        http://tools.ietf.org/pdf/rfc3122.pdf
[7]   IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels,
        http://tools.ietf.org/pdf/rfc2119.pdf
[8]   IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification,
        http://tools.ietf.org/pdf/rfc2460.pdf
[9]   IETF RFC 4291, IP Version6 Addressing Architecture,
        http://tools.ietf.org/pdf/rfc4291.pdf

CONFIDENTIAL

INDEX

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

R

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL